THE NATA CODE - PROGRAMMER'S MANUAL

JSC-11103

W. L. Bade and J. M. Yos
AVCO SYSTEMS DIVISION
201 Lowell Street
Wilmington, Massachusetts 01887

NAS9-9744
CR-141744
2 September 1975
Final Report
Volume III

Prepared for

LYNDON B. JOHNSON SPACE CENTER
Houston, Texas 77058

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-141744		

4. Title and Subtitle	5. Report Date
THE NATA CODE - PROGRAMMER'S MANUAL, Volume III	2 September 1975
	6. Performing Organization Code: JSC-11103

7. Author(s)	8. Performing Organization Report No.
W. L. Bade and J. M. Yos	AVSD-0138-75-RR
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Avco Systems Division 201 Lowell Street Wilmington, Massachusetts 01887	NAS9-9744
	13. Type of Report and Period Covered: Contractor Report Final Report

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058	

15. Supplementary Notes: The final report on Contract NAS9-9744 consists of three volumes:
- Vol. I The NATA Code - Theory and Analysis (NASA CR-2547)
- Vol. II The NATA Code - User's Manual (NASA CR-141743)
- Vol. III The NATA Code - Programmer's Manual (NASA CR-141744) (present document)

16. Abstract

The NATA code is a computer program for calculating quasi-one-dimensional gas flow in axisymmetric and two dimensional nozzles and rectangular channels. The code also computes stagnation point conditions on axisymmetric or two-dimensional models and the conditions on blunt wedges inserted into the flow. The theory and analysis underlying the calculations have been documented in Volume I of this final report. Code inputs, outputs, and precoded data have been defined in Volume II, which is a user's manual. The present, third volume of the final report is a programmer's manual for the code. It provides a listing of the Fortran IV source program; a complete glossary of Fortran symbols; a discussion of the purpose and method of operation of each subroutine (including mathematical analyses of special algorithms); and a discussion of the operation of the code on IBM/360 and UNIVAC 1108 systems, including required control cards and the overlay structure used to accommodate the code to the limited core size of the 1108. In addition, this volume provides similar information to document the programming of the NOZFIT code, which is employed to set up nozzle profile curvefits for use in NATA.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Nonequilibrium flow Nozzle flow Arc-heated wind tunnels Stagnation-point heat transfer Wedge heat transfer Laminar boundary layer Thermal nonequilibrium	STAR Subject Category: 34 (Fluid Mechanics and Heat Transfer)

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	637	$16.25

*For sale by the National Technical Information Service, Springfield, Virginia 22151

PREFACE

This is the third volume of a final report documenting the development of a computer program (NATA) for calculating the flow in arc-heated wind tunnels and the conditions on models tested in such reentry simulation facilities. Volume I of the report provides a general summary of the code's capabilities and presents the theory and analysis underlying its operation. Volume II is a user's manual which defines NATA's inputs, outputs, and precoded data on standard gas models and nozzle geometries. The present volume is a programmer's manual for NATA. It documents the programming of NATA in sufficient detail to support code modifications, and provides information needed for running the code on the UNIVAC 1108 under the EXEC II system, or on the IBM 360/75 under HASP II.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

THE NATA CODE - PROGRAMMER'S MANUAL

By W. L. Bade and J. M. Yos
Avco Systems Division
Wilmington, Massachusetts

1. INTRODUCTION

The NATA code is a computer program for solving problems of steady, quasi-one-dimensional gas flow in nozzles. The code's capabilities, and the theory and analysis underlying its operation, have been documented in the first volume of this report (ref. 1). The inputs, outputs, and precoded data on standard gas models and nozzle geometries have been defined in the second volume (ref. 2). The present, third volume provides information required for running NATA on UNIVAC 1108 and IBM 360 computers, and documents the programming of NATA so that programmers and computer-oriented users will be able to evaluate the techniques used, maintain the code in the face of operating-system changes, convert it for use on other types of computer, introduce modifications and correct errors.

Section 2 provides a general description of NATA, including a catalog of its subroutines with a statement of function and a list of calling routines for each. This section also gives sample control-card decks for running the code on the UNIVAC 1108 under the EXEC II system, and on the IBM 360/75 under HASP II and OS 21.

Section 3 discusses each subroutine, explaining its purpose in the code and its method of operation. Many special algorithms are analyzed. Where appropriate, flowcharts are used to display the logic and clarify the methods used. The Fortran variables stored in common blocks and used in the individual routines are all defined in a glossary of symbols (Section 4). Section 5 lists the subroutines in which each common variable is set, used, or referenced as a subroutine argument. Finally, a complete listing of the source program is given in Section 6.

An appendix provides similar documentation of the NOZFIT code, a much smaller program provided for setting up nozzle profile curvefits for use in NATA. A user's manual for NOZ-FIT is included in Volume II (ref. 2). The appendix includes documentation of a general-purpose plotting subroutine (GRAPH) for producing plots on S-C 4020, S-C 4060, or other plotting equipment using the NAA software package for the SC-4020 (ref. 3).

2. STRUCTURE AND OPERATION

2.1 Brief Description of the NATA Code

NATA is a Fortran IV program consisting of a main program, two block data routines, and 66 subroutines. The source deck contains approximately 8500 cards. The program exists in two versions, one for use on the IBM 360 system, the other for the UNIVAC 1108. The IBM version is entirely in double precision, whereas the UNIVAC version is a single precision program with some double precision arrays and subroutines. One version can be converted into the other by inserting or removing most of the IMPLICIT REAL*8 (A-H, Ø-Z) cards which type all floating-point variables as double precision in the IBM 360 version.*

The IBM 360 version requires about 410K bytes of core storage, including buffers. The UNIVAC version is run on the 1108 using overlay, and requires about 123000_8 or 42500_{10} words of storage. It fits into the two-bank processors at NASA/JSC with about 2000 words to spare.

2.2 Subroutines and Linkage

A list of the routines of NATA is presented below. Names of entry points are included, and are enclosed in parentheses to distinguish them from subroutine and function names. The function performed by each routine or entry point is described briefly, and in each case a complete list of calling routines is included. The order of the list is as follows: main program; block data routines; subroutines, functions, and entry points in alphanumeric order.

---

*Also, the IBM version contains two extra subroutines, EXP and RESET, for reasons explained below.

MAIN	Main program; calls other routines to perform the required calculations.
BLKD1	BLØCK DATA routine containing precoded data on elements, species, reactions, gas models, and nozzle geometri s.
BLKD2	BLØCK DATA routine containing precoded data on transport-property cross sections.
AESØLN	Calculates the effective area ratio A_e and its derivative dA_e/dx from data on nozzle geometry and the boundary layer displacement thickness.* Called by GEØM.
AGSØLN	Calculates the geometric area ratio A_g and the axial coordinate x from data on the effective area ratio A_e and the boundary layer displacement thickness. Called by FRØZEN, EQUIL, and AXFIT.
AXFIT	Calculates the axial coordinate x corresponding to a given value of the effective area ratio A_e in the nonequilibrium solution by the perturbation method. Called by NØNEQ.
AXSECT	For a given list of species pairs, adds those pairs whose cross sections have not previously been specified to a particular step of the transport cross section calculation. Called by XSECT.
BLAYER	Calculates boundary layer momentum and displacement thicknesses, heat flux, and shear stress. Called by FRØZEN, EQUIL, and BLCALL.
BLCALL	Sets up and executes calls to BLAYER in the nonequilibrium solution. Called by DERIVS and PRTA.

*Throughout this report, the mathematical notation is as defined in Volume I (ref. 1).

BXSECT — Searches a parameter array for references to the indices of species pairs used in the cross section calculation; also determines the correspondence between entries in the parameter array and the steps in the cross section calculation. Called by XSECT.

CØMM — Computes the species production terms $P_i \chi_i$ in the nonequilibrium solution; in runs with an electronic nonequilibrium gas model, also computes the radiative loss and the energy transfer to the electron gas. Called by DERIVS.

CXSECT — Determines the correspondence between species pairs in the master list of species and those in the species list for the current case. Called by XSECT.

DERIVS — Organizes the calculations of rates of change of the species concentrations and other flow variables in the nonequilibrium solution. Called by NØNEQ and RNKT.

DSMSØL — Performs simultaneous solution of a system of linear, inhomogeneous equations by calling subroutine SIMQ. Called by EQCALC, EXACT, NEWRAP, NØNEQ, and PERT.

DUMP — Prints name of routine in which an error has been detected and sets an error indicator to terminate the case. Called by AGSØLN, DSMSØL, EQCALC, FINDX, GEØM, GEØMAR, MATINV, NEWRAP, NØNEQ, PRØP, RESTMP, and WEDGE.

DUMPER — Prints a large diagnostic dump of common data when a case is terminated by a call to DUMP because of an error. Called by MAIN.

ELCØND — Computes the electrical conductivity of the gas mixture. Called by TRANSP.

ELTIME — Prints elapsed time since the beginning of the run and time since the last previous time message. Called by MAIN and MØDEL.

EPART — Computes parameters for radiative energy loss and energy transfer to the electrons for a specified reaction. Called by CØMM.

EQCALC — Computes mole fractions and thermodynamic conditions for thermochemical equilibrium at specified temperature and pressure. Called by INTA and MØDEL.

(EQUIL) — Entry point in subroutine FRØZEQ; organizes the calculation of the equilibrium flow solution. Called by MAIN.

EXACT — Computes the derivatives of the species concentrations and other flow variables in the non-equilibrium solution. Called by DERIVS.

EXP — In the IBM 360 version, the double precision exponential function e^x with underflow suppression; included to avoid underflow messages. Not used in UNIVAC version. Called by BLAYER, CØMM, EQCALC, NEWRAP, ØUT1, PRØP, RNKT, THERM, and WEDGE.

FINDX — Solves for the axial coordinate x corresponding to a given geometric area ratio A_g in the upstream or downstream region. Called by AGSØLN, AXFIT, FRØZEQ, MAIN, and NEXTMP.

(FINDXC) — Entry point in subroutine FINDX; solves for the axial coordinate x corresponding to a given half width of one of the profiles in a channel. Called by NEXTMP.

(FRØZEN) — Entry point in subroutine FRØZEQ; organizes the calculation of the frozen flow solution. Called by MAIN.

FRØZEQ	Organizes the frozen and equilibrium flow solutions. The subroutine name is not referenced in NATA; called only through the entry points EQUIL and FRØZEN.
GEØM	Organizes the geometry calculations for the nonequilibrium solution. In the direct integration, it computes A_e and $d\ell n A_e/dx$; in the inverse method, it computes ρ and $d\ell n\rho/dx$. Called by CØMM.
GEØMAR	Computes the geometric area ratio A_g and its derivative $d A_g/dx$ at a given axial coordinate x. Called by AGSØLN, BLAYER, FINDX, GEØM, ØUT1, and THRØAT.
(GMAR)	Entry point of subroutine GEØMAR; computes the profile ordinate y at a specified axial coordinate x. Called by CØMM, ØUT1, and RADIUS.
(GMAR2)	Entry point of subroutine GEØMAR; computes the profile ordinates y, z at a specified axial coordinate x in a channel. Called by AGSØLN, CØMM, ØUT1, and RADIUS.
(GMAR3)	Entry point of subroutine GEØMAR; computes the profile ordinates, y, z and their derivatives dy/dx, dz/dx at a specified axial coordinate x in a channel. Called by AESØLN, BLAYER, and FINDX.
INGAS	Sets up arrays used in the chemical description of the gas mixture. Called by MAIN.
INIT	Initializes same control parameters and nondimensionalizes the species thermochemical data using the reservoir temperature. Called by MAIN and RESTMP.

(INTA)	Entry point of subroutine INGAS; organizes the calculation of equilibrium conditions in the upstream reservoir for given reservoir temperature and pressure. Called by MAIN and RESTMP.
KANDMU	Calculates quantities proportional to the viscosity and translational thermal conductivity of the gas mixture. Called by TRANSP.
KINT	Computes quantities used in the transport property calculations. Called by TRANSP.
LIST	Prints the "problem summary" portion of the output for each case. Called by MAIN.
MATINV	Computes the inverse A_{ki} of the matrix $\bar{\alpha}_{ij}$ specifying the elemental composition of the independent species. Called by INGAS.
MØDEL	Computes stagnation conditions on models inserted into the flow. Called by FRØZEQ, NØNEQ, and PRTA.
NEWRAP	Computes the equilibrium flow conditions at specified temperature and entropy. Called by FRØZEQ, MAIN, NØNEQ, and NRMAX.
NEXTMP	Determines the axial coordinate of the next location at which model condition calculations are to be done. Called by FRØZEQ and NØNEQ.
NØNEQ	Organizes the nonequilibrium flow solution. Called by MAIN.
NRMAX	Calculates the temperature and other conditions at the sonic point in an equilibrium flow starting at specified upstream reservoir conditions. Called by MAIN and RESTMP.
ØUT	Prints a list of definitions of output variables. Called by MAIN.

ØUT1	Prints conditions in the upstream reservoir and at the sonic points of the frozen and equilibrium solutions. Called by MAIN.
(ØUT2)	Entry point of subroutine ØUT1; prints the conditions at each point in the flow solutions. Called by FRØZEQ and PRTA.
PERT	Computes the perturbations in the flow variables during the nonequilibrium solution by the perturbation method. Called by NØNEQ.
PIØMEG	Computes the Maxwell-averaged electron-neutral momentum transfer cross section for given electron temperature T_e. Called by CØMM.
PRØP	Computes the conditions at a point in the frozen flow solution. Called by FRØZEQ.
PRTA	Organizes the output of conditions at each point in the nonequilibrium flow solution. Called by NØNEQ.
PUTQIN	Organizes the calculations of the cross sections used in computing the transport properties. Called by TRANSP.
QCØUL	Computes the factor 0.8 Q_c in the Coulomb cross section for use in the transport property calculations. Called by PUTQIN.
QEX	Computes the cross sections for exchange interaction. Called by PUTQIN.
QEXP	Computes the cross sections for the exponential potential $\phi = Ae^{-r/\rho}$. Called by PUTQIN.
QINTRP	Interpolates tabular data in cross section calculations. Called by QEXP, QLJ, and QTAB.
QLJ	Computes the cross sections for the Lennard-Jones (6-12) potential. Called by PUTQIN.

QMIX — Computes transport cross sections from the mixing rule, equation I(102).* Called by PUTQIN.

QREPP — Computes the cross sections for an attractive or repulsive power law potential $\phi = Ar^{-\eta}$. Called by PUTQIN.

QSAME — Sets the cross sections for a species pair equal to a constant multiple of those computed previously for another pair. Called by PUTQIN.

QTAB — Computes the transport cross sections by linear interpolation in tables of $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$, and B* versus temperature. Called by PUTQIN.

Q11 — Computes the transport cross sections using the fairing option, equations I(107) and I(108). Called by PUTQIN.

Q12 — Computes the cross sections by the generalized mixing rule, equation I(109). Called by PUTQIN.

Q13 — Sets one of the cross sections for a given species pair equal to a constant times another cross section for the same pair. Called by PUTQIN.

Q14 — Recomputes the cross sections for a species pair using equations I(112). Called by PUTQIN and Q11.

RADIUS — Computes the profile ordinate, the geometric area ratio, and another quantity required in the calculation of the boundary layer at the first point in the flow solution. Called by BLAYER.

---

*Equations in Volume I (ref. 1) are cited in the form I( ).

READ — Reads the input data for each case and sets up arrays defining the flow geometry and gas model. Called by MAIN.

RESET — In the IBM 360 version, simulates a UNIVAC 1108 library routine for storing the time at the start of a run. Called by ELTIME.

RESTMP — Computes the equilibrium conditions in the upstream reservoir based on pressure and mass flow or mass flow and stagnation enthalpy data. Called by MAIN.

RNKT — Computes the changes in the species concentrations and other flow variables over a step in the nonequilibrium integration. Called by NØNEQ.

SHØCK — Computes the pressure ratio for an oblique shock in a perfect gas. Called by WEDGE.

SIMQ — Solves systems of linear inhomogeneous equations. Called by DSMSØL and RESTMP.

STUNTS — Prints an edit of species thermal properties. Called by MAIN.

(STUNT2) — Entry point in subroutine STUNTS; prints edits of the steps in the transport cross section calculation. Called by MAIN.

THERM — Computes the species thermal properties at a specified temperature. Called by DERIVS, EQCALC, FRØZEQ, MØDEL, NEWRAP, and STUNTS.

(THERM1) — Entry point in subroutine THERM; computes only the species specific heats at a specified temperature. Called by MAIN, ØUT1, and TRANSP.

THRØAT — Implements the switch from the upstream solution by the inverse method to the downstream direct integration. Called by NØNEQ.

(TIME) Entry point in subroutine RESET (IBM 360 version only); simulates a library timing routine available on the UNIVAC 1108. Called by ELTIME.

TRANSP Organizes the transport property calculations. Called by BLAYER, FRØZEQ, MAIN, MØDEL, ØUT1, WEDGE, and STUNT2.

(TRANSX) Entry point in subroutine TRANSP; sets up arrays used in the transport property calculations. Called by MAIN.

WEDGE Computes and prints the conditions on the surface of a blunt wedge model inserted into the flow. Called by MØDEL.

WESØLN Calculates the quantities z, zz', and (zz')', based on the approximate solution I(484) of the wedge equation, for given values of ζ and Γ. Called by WEDGE.

XSECT Sets arrays required for applying the precoded cross section data to the species present in the current gas model and sets up the edited cross section calculation. Called by TRANSX.

2.3 UNIVAC 1108 Version

The UNIVAC version of NATA has been set up to run on the 1108 processors at NASA/JSC under the EXEC II operating system. This version is a single precision program in which some arrays and three subroutines (SHØCK, SIMQ, and WESØLN) are typed as double precision to maintain adequate accuracy. These three routines all perform operations which, under some conditions, can lead to loss of many significant digits because of cancellation of terms. The typing of these routines as double precision is accomplished by retaining the IMPLICIT REAL*8 (A-H, Ø-Z) statements from the IBM 360 version. The UNIVAC Fortran V compiler interprets these statements correctly, a feature included by Sperry Rand in the interest of compatibility with IBM 360 Fortran IV.

NATA is too large to be run on the 1108 without use of overlay. Figure 1 shows the overlay structure employed. The overlay is implemented by means of the EXEC II Memory Allocation Processor (ref. 4). In figure 1, the names enclosed in boxes represent the routines of NATA listed in Section 2.2. Each box represents a segment or subsegment with the associated name (S.---) given in parentheses. The level 0 segment S.MAIN is permanently in core. The various level 1 segments overlay one another. In a typical NATA case, the loading sequence for the level 1 segments is as follows:

S.READ	Read input data, set up gas model, etc.
S.LIST	Print problem summary.
S.INIT	Compute reservoir conditions, etc.
S.XSEC	Set up transport cross section calculation for current gas model.
S.ØUT	Print definitions of output labels.
S.FRØZ	Calculate equilibrium solution to throat, to obtain displacement thickness at throat.
S.INIT	Recalculate reservoir conditions corrected for displacement thickness at throat.
S.FRØZ	Calculate frozen solution; calculate equilibrium solution.
S.NØNE	Calculate nonequilibrium solution.

In the nonequilibrium solution, the subsegment S.N2 is loaded first for use in the initial solution by the perturbation method. When the nonequilibrium integration is started, the integration routine RNKT is loaded, overlaying S.N2.

Control cards for using NATA on the UNIVAC 1108 processors at NASA/JSC are shown in figures 2 and 3. Figure 2

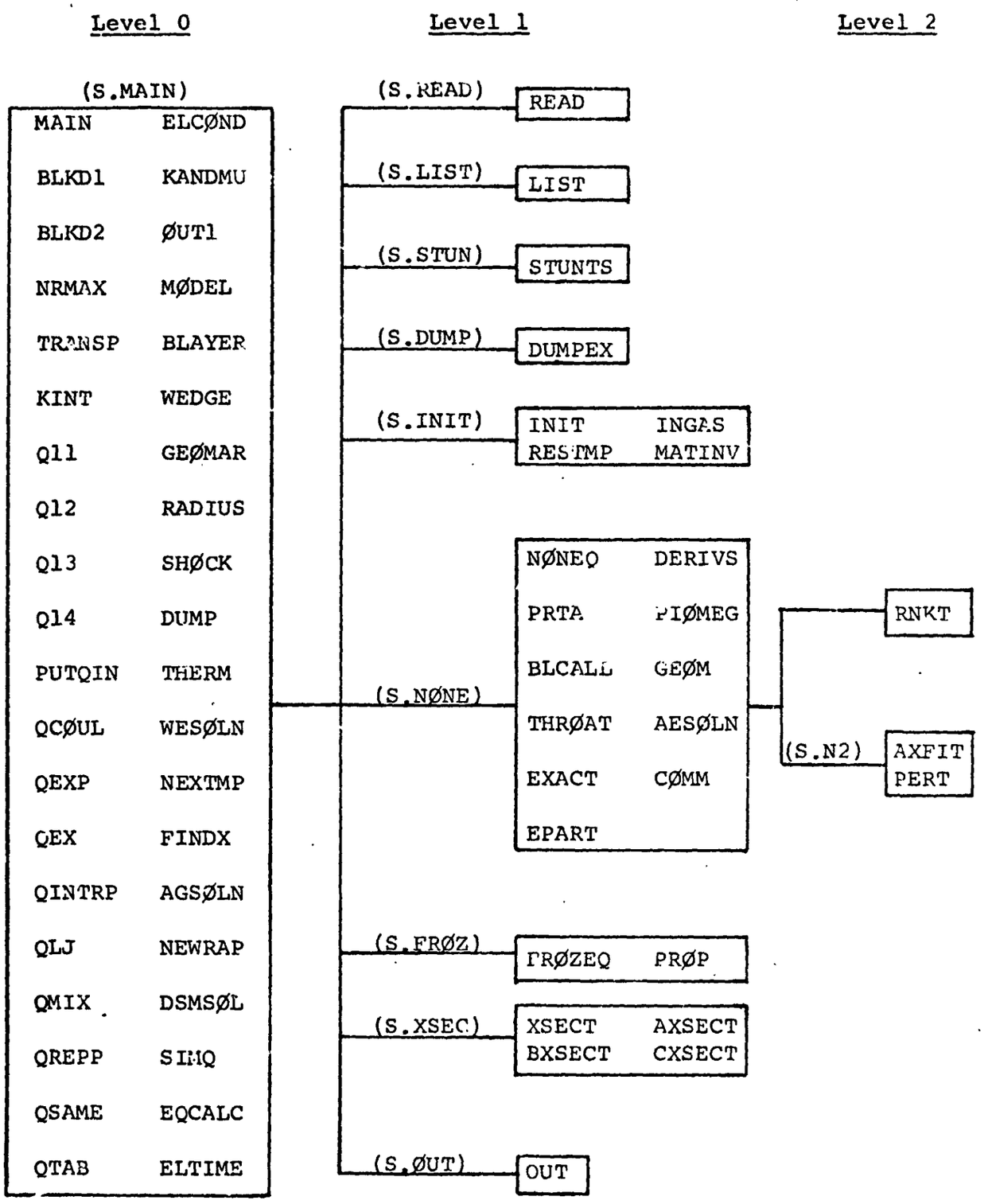


Figure 1. Overlay Structure for UNIVAC 1108 Version of NATA

```
Z RUN N0320.....                                              SCOTT/
N MSG      FILE REQ.   TAPE 01 FH432 0    FSTRN 00
   ; C=OUT
 FOR MAIN.MAIN

  FOR SUB.SUB

  MAP NATA/NATA
S.0     SEG S.MAIN-*(S.READ.S.INIT.S.XSEC.S.FROZ.S.NONE.S.OUT.S.LIST.
            S.STUN.S.DUMP)
S.MAIN SEG MAIN-BLKD1-BLKD2-NRMAX-TRANSP-KINT-Q11-Q12-Q13-Q14-PUTQIN-
           QCOU_-QEXP-QEX-QINTRP-QLJ-QMIX-QREPP-QSAME-QTAB-FLCOND-KANDMU-
           OUT1-MODEL-BLAYER-WEDGE-GEOMAR-RADIUS-SHOCK-DUMP-THERM-WESOLN-
           NEXTMP-FINDX-A' .OLN-NEWRAP-DSMSOL-SIMQ-EQCALC-ELTIME
S.READ SEG READ
S.LIST SEG LIST
S.STUN SEG STUNTS
S.DUMP SEG DUMPEX
S.INIT SEG INIT-RESTMP-INGAS-MATINV
S.XSEC SEG XSECT-AXSECT-BXSECT-CXSECT
S.FROZ SEG FROZEQ-PROP
S.NONE SEG NONEQ-PRTA-BLCALL-THROAT-EXACT-EPART-DERIVS-PIOMEG-GEOM-AESOLN-
           COMM-*(RNKT.S.N2)
S.N2   SEG AXFIT-PERT
S.OUT   SEG OUT
  XQT CUR
  TRW C
  UT C
  EF C
  REL C
```

Figure 2. Control Cards for Generating a NATA PCF Tape on the UNIVAC 1108

```
Z RUN N0320.....                                              SCOTT/
N MSG      FILE REQ.   TAPE 01 FH432 0    FSTRN 00
 ASG A=V01234
 XQT CUR
 TRW A
 IN A
 REL A
 XQT NATA
```

Figure 3. Control Cards for Running NATA on the UNIVAC 1108

illustrates a control card deck for generating a NATA PCF tape. The first card is a schematic run card. The deck is set up to compile MAIN and a subroutine "SUB"; the source decks are assumed to be inserted following the "FØR" cards. The "RUN", "MSG", "ASG", "FØR", "MAP", "XQT", "TRW","ØUT", "TEF", and "REL" cards all have a 7-8 punch in column 1 which was not reproduced by the equipment used to print this listing.

Figure 3 shows the control cards for a NATA run from an existing PCF. The "V01234" following the equals sign on the ASG card is a representative reel number for the PCF tape. The "XQT NATA" card should be followed by the input data for the case, which are described in Section 2 of Volume II (ref. 2).

The control cards required for use of NATA at other UNIVAC 1108 installations may differ in some respects from those illustrated in figures 2 and 3.

2.4 IBM 360 Version

Since NATA is run as a single precision program on the UNIVAC 1108, single precision function names (SQRT, ABS, AMAX1, etc.) are used throughout the program, except in the three double precision routines, SHØCK, SIMQ, and WESØLN. To avoid the tedium of changing function names when the code is converted for use as a double precision program on the IBM 360, the single precision names are retained in the source program for the IBM version and are interpreted as referring to the double precision functions. In the IBM 360/75 installation at Avco Systems Division, this reinterpretation of the single precision Fortran function names is accomplished by using a special subroutine library in the link edit step. At other installations where such a special library may not be available, the same result can be achieved by including, in the NATA source deck, Fortran subroutines defining all of the single precision Fortran function names in terms of the corresponding double precision functions. An example of this procedure is provided by the function EXP in the IBM version of NATA. As shown by the listing of EXP in Section 5, this subroutine redefines EXP in terms of DEXP. It also

sets EXP(X) to 0 for X less than -180, to avoid the production of paper-wasting underflow messages.

Since the Fortran G compiler gives a level 8 diagnostic when a single precision function name is used with a double precision argument, the G compiler cannot be used on the NATA version presented here. Instead, the H compiler, which gives only a level 4 diagnostic, must be used. Several of the NATA routines, including READ and the two block data routines, are too large for the standard H compiler. At Avco Systems Division, an available large compiler (FØRTHMAX) is used to process the NATA source deck.

In both versions of NATA, the elapsed time since the beginning of the run is printed out at several stages in each case. In the UNIVAC version, these times are obtained by calling routines RESET and TIME, which are available in the 1108 system library. In the IBM 360 version in operation at Avco Systems Division, RESET and TIME are simulated using the system library routine ACUCPU, which gives the unexpended CPU time remaining before automatic job termination at TIME.GØ.

In summary, conversion of the UNIVAC version of NATA into the IBM 360 version requires the following operations:

(1) An IMPLICIT REAL*8 (A-H, Ø-Z) card must be inserted into each NATA routine except ELTIME, ØUT, RESET, SHØCK, SIMQ, and WESØLN. In MAIN, this must be the first card apart from comment cards. In the other routines, the IMPLICIT statement must directly follow the SUBRØUTINE, FUNCTIØN, or BLØCK DATA statement.

(2) The subroutines EXP and RESET must be inserted.

(3) If a special system library for interpreting the single precision Fortran function names as double precision functions is not available at the computer installation, Fortran subroutines to define the single precision names as double precision functions must be added to the source deck.

Figure 4 illustrates the control cards used to run NATA from a tape containing the binary deck, on the Avco Systems Division IBM 360/75 system. (The required JCL cards will differ somewhat at other IBM 360 installations). The fourth, fifth, and sixth cards in figure 4 invoke the special system library which interprets the single precision function names as referring to the corresponding double precision functions. The binary tape containing NATA is identified on the DD card with VØLUME = SER = NATA3. The data set GØ.FT08F001 is the binary tape upon which NATA can write data for subsequent plotting. GØ.FT07F001 is the data set for punched output.

```
//          JOB                                                          *X
//              MSGLEVEL=1
//   XEC FORTHLG.REGION.GO=420K.TIME.GO=4
//LKED.SYSLIB DD DSNAME=VER1.TEMPLIB.DISP=SHR
// DD DSNAME=SYS1.FORTLIB.DISP=SHR
// DD DSNAME=SYS1.DOUBLEP.DISP=SHR
//LKED.SYSUT1 DD SPACE=(1024,(600,50))
//LKED.SYSLMOD DD SPACE=(3072,(200,10,1),RLSE)
//LKED.SYSLIN DD UNIT=(TAPE9,,DEFER),LABEL=(,BLP),DISP=(OLD,KEEP),      X
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),VOLUME=SER=NATA3
//  DD *
 ENTRY MAIN
/*
//GO.FT08F001 DD UNIT=(TAPE9,,DEFER),LABEL=(,BLP),DISP=(NEW,KEEP),      C
//             DCB=(RECFM=VBS,LRECL=92,BLKSIZE=4604),VOLUME=SER=DATASV
//GO.FT07F001 DD SYSOUT=B
//GO.FT05F001 DD *
```


Figure 4. Control Cards for Running NATA on the IBM 360

3. ANALYSIS OF ROUTINES

The present section discusses and explains the main program and each subroutine of the NATA code. The purpose of this exposition is to provide an entry into the coding for programmers who wish to analyze errors, make corrections, or introduce modifications into NATA.

All of the physics and gas dynamics and most of the mathematical analysis underlying the programming have already been documented in Volumes I and II of this report (refs. 1, 2). Where appropriate, individual sections and equations of Volumes I and II are referenced.* In cases where the underlying mathematical analysis has not been documented previously, it is explained in the present section.

The routines of NATA are discussed in the following order: first the main program, then all subroutines and functions in alphanumeric order of their names. This is the same order as that in which the routine names are listed in Section 2.2 and at the beginning of Section 5, and in which the source listings are presented in Section 6.

3.1 Main Program

The main program performs certain initialization functions, calls the major subroutines in proper sequence, implements some of the input options, and controls job termination. A flowchart of MAIN is shown in figure 5. The operations shown in this flowchart will be discussed approximately in the order in which they are performed during execution of the code.

3.1.1 Job Initialization

The 12 input control parameters ISW1A to ISW6B are initialized by arithmetic statements in MAIN before any input data

---

*Equations from Volumes I and II will be referenced in the form I( ) and II( ).

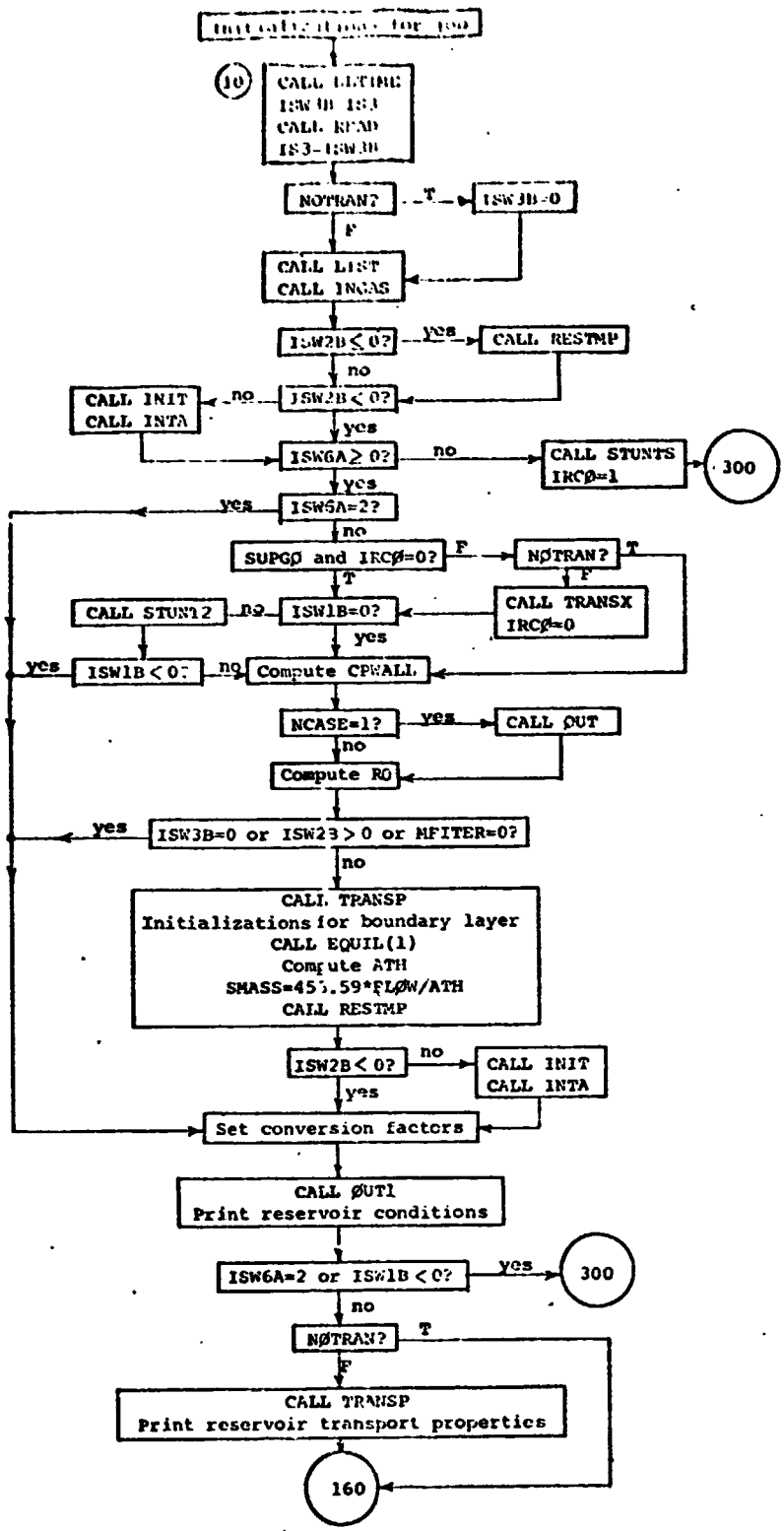
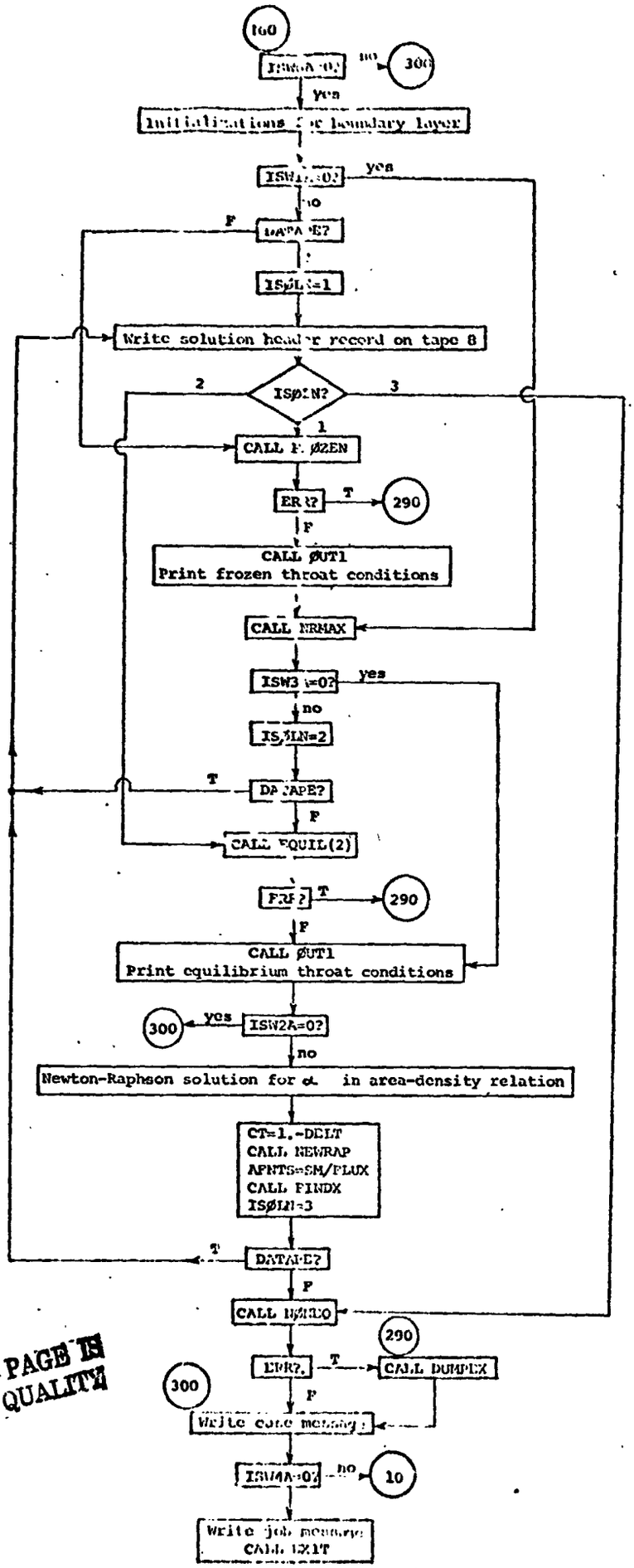


Figure 5a. Flowchart of NATA Main Program (Part a)



have been read. This procedure is used because these parameters are in unlabelled common, and hence cannot be preset in a BLØCK DATA routine.

3.1.2 Case Initialization

The calculations for each case in a job always begin with statement 10. First, subroutine ELTIME is called to determine the elapsed time ET0 since the start of the first case. Then subroutine READ is called. This routine reads the input data for the case, sets up arrays describing the gas model and the nozzle geometry, and prints some of the data defining the case being run. Next, subroutine LIST is called to print the remaining data defining the current case. Subroutine ELTIME is then called again to print the time. ELTIME is called at various locations in MAIN, to provide data on the execution time consumed by major portions of the calculation. These subsequent calls to ELTIME are omitted from the flowchart in the interest of simplification.

Before the call to LIST, the value of the logical input variable NØTRAN is checked. NØTRAN is a switch to allow the suppression of all transport property calculations throughout NATA. With NØTRAN = .TRUE., flow solutions can be run for a gas model containing user-specified species for which transport cross section data have not been provided. Of course, when the transport property calculations are suppressed, the boundary layer cannot be included in the flow solutions. Hence, for NØTRAN = .TRUE., ISW3B is set to zero. However, the imput value of ISW3B is saved in the location IS3, and is reset before the call to READ in subsequent cases.

MAIN next increments the case counter NCASE and initializes some variables including the indicator ERR in common block /ERRØR/. This indicator is set to .FALSE. near the beginning of each case. If the programmed checks in any subroutine subsequently detect an error, ERR is set to .TRUE., and this indication causes control to be returned to MAIN as quickly as possible, with no further calculations. Statements in MAIN then cause subroutine DUMPEX to be called. DUMPEX prints out a diagnostic dump consisting of the current values of most of the variables in labelled and unlabelled

common. The details of this scheme are explained more fully in the sections (below) on subroutines DUMP and DUMPEX. Most of the tests on ERR in MAIN have been omitted from the flowchart for brevity, but the ones following CALL FRØZEN, CALL EQUIL(2), and CALL NØNEQ have been included to illustrate the method.

MAIN next calls subroutine INGAS to set up certain arrays used to describe the chemical nature of the gas mixture. Then, if ISW2B ≤ 0 (i.e., if the determination of reservoir conditions is to be based partly on the total mass flow), subroutine RESTMP is called to compute the reservoir temperature and pressure. Unless ISW2B is negative, subroutine INIT and INTA are then called. INIT initializes a number of control variables and nondimensionalizes some species properties using the reservoir temperature. INTA computes the species mole fractions and the thermal properties of the gas in the reservoir. For negative ISW2B, INIT and INTA are called in RESTMP with the final reservoir temperature and pressure, and thus do not have to be called in MAIN.

3.1.3 Special Options

If ISW6A is negative, NATA computes and prints tables of thermal properties for all of the species in the current gas model, instead of calculating a flow solution. This operation is performed by subroutine STUNTS.

If ISW6A is positive, NATA computes and prints out the reservoir conditions (as usual), but does not proceed with the flow solutions. For the specific value ISW6A = 2, the calculation of the transport properties in the reservoir is also omitted. This feature allows the NATA user to employ the code for thermochemical calculations for a gas model containing nonstandard species without setting up transport cross section data for the species.

If ISW6A is 0 (the normal, preset value), NATA generates one or more flow solutions.

If ISW6A is zero or positive (but not equal to 2), the transport properties in the reservoir are computed. Before transport property calculations can be done, the entry TRANSX of subroutine TRANSP must be called. However, the call to TRANSX can be omitted if TRANSX has been called in a previous case with the same gas model. The gas model is the same as in the preceding case if SUPGØ = .TRUE., and TRANSX was called in a previous case if IRCØ = 0. If NØTRAN is .TRUE., the call to TRANSX is skipped.

Another special option is an edit of the transport cross section calculations, performed by calling the entry STUNT2 of subroutine STUNTS. This call is executed if the input control parameter ISW1B is nonzero. For ISW1B negative, STUNT2 also prints a table of averaged transport cross sections; in this case, no flow solution is run.

3.1.4 Specific Heat of the Gas at the Wall Temperature

MAIN computes the specific heat of the gas at the nozzle wall temperature, TWALL. This calculation is based on the assumption that the gas at this low temperature is a mixture of the "cold species" used to define the elemental composition. The specific heat is calculated as

$$C_{pw} = \frac{R_0}{W_c} \sum_{k=1}^{n_c} X_k \frac{C_{pk}}{R_0} \qquad (1)$$

in which W_c is the mean molecular weight of the cold gas, n_c the number of cold species, X_k the mole fraction of the k^{th} cold species, and C_{pk}/R_0 the nondimensional molar specific heat at constant pressure for the k^{th} cold species. This species specific heat is given by CCPJ(I), where I = IJCS(K) is the index, in the list of species for the current gas model, of the k^{th} cold species.

3.1.5 Definitions of Output Variable Labels

In the first case of each job (NCASE = 1), MAIN calls subroutine ØUT to print a list of definitions, with units,

of the output variable labels used in printing out the flow conditions and test conditions on models. These labels differ, in most cases, from the Fortran variable names used in the code.

3.1.6 Correction of the Reservoir Conditions for the Displacement Thickness at the Throat

If the boundary layer is to be included in the solution (ISW3B $\neq$ 0), and if the reservoir condition calculations are based partly on the total mass flow (ISW2B $\leq$ 0), then NATA normally recalculates the reservoir conditions, taking account of the effects of the boundary layer displacement thickness on the equilibrium sonic mass flux. For ISW2B $\leq$ 0, subroutine RESTMP determines the reservoir temperature and pressure so as to match the sonic mass flux SMASS and one other condition (on pressure or enthalpy). Initially, when RESTMP is called the first time, SMASS is calculated as the ratio of the input total mass flow to the geometric cross sectional area of the nozzle at the throat. To correct these results for the displacement thickness effect, MAIN calls entry EQUIL(IPASS) of subroutine FRØZEQ with the argument IPASS = 1. EQUIL then computes the equilibrium solution up to the throat, including the boundary layer. MAIN then calculates the effective cross sectional area ATH at the throat including the effects of the displacement thickness, recomputes the sonic flux SMASS, and calls RESTMP for a second time to recompute the reservoir conditions. This correction can be suppressed, if it is not desired, by setting MFITER = 0 in the input.

3.1.7 Conversion Factors

Next, MAIN sets the array of conversion factors CF(I) in common block /CØNVRT/. These factors convert the nondimensional flow variables used in the internal computations to the units used in the printed output.

3.1.8 Output of Reservoir Conditions

The nondimensional flow variables CT = T/T_0, PRES = p/p_0, etc., are now set to their reservoir values and subroutine ØUT1 is called to print out the reservoir conditions.

3.1.9 Solution Header Records on Tape 8

If the frozen flow solution is to be generated (ISW1A $\neq$ 0) and if data are to be written on a binary output tape for subsequent plotting (DATAPE = .TRUE.), the main program next writes a "header" record on tape 8 for the frozen solution. This record contains the nozzle index NØZZLE = NPRØFL(1), the type of solution (ISØLN = 1 for frozen flow), and the reservoir conditions.

3.1.10 Frozen Solution

To generate the frozen flow solution, MAIN calls the entry FRØZEN of subroutine FRØZEQ. It then calls ØUT1 to print the sonic-point conditions in the frozen solution.

3.1.11 Equilibrium Solution

MAIN calls subroutine NRMAX to compute the conditions at the sonic point in the equilibrium solution. These data are needed for both the equilibrium and nonequilibrium solutions. Then, unless ISW3A = 0, MAIN checks DATAPE. If this control variable is .TRUE., a header record is written on tape 8 for the equilibrium solution, and the entry EQUIL(IPASS) of subroutine FRØZEQ is called with IPASS = 2 to produce the solution. Finally, ØUT1 is called to print out the equilibrium throat conditions.

3.1.12 Nonequilibrium Solution

If the nonequilibrium solution is to be computed (ISW2A $\neq$ 0), MAIN determines the constants C and α in the area-density relation I(383) assumed in the solution by the inverse method upstream of the throat. The parameter α is determined by a Newton-Raphson solution of equation I(385) or I(386). The iteration is started using the initial estimate

$$\alpha \simeq 4 + 8 \ln \left(\frac{\rho_*}{\rho_0}\right) \qquad (2)$$

which can be derived by taking the logarithm of I(385) and

expanding $\ell n\ (\alpha + 2)$ around $\alpha = 0$:

$$\ell n\ (\alpha + 2) \simeq 2 + \frac{\alpha}{2} - \frac{\alpha^2}{8} \qquad (3)$$

Then to obtain starting conditions for the perturbation solution, MAIN sets the nondimensional temperature CT to 1-DELT, calls subroutine NEWRAP to compute the equilibrium flow conditions at this temperature, computes the corresponding effective area ratio AFNTS, and calls subroutine FINDX to determine the axial coordinate CX at which these conditions occur. If DATAPE = .TRUE., a header record is written on tape 8 for the nonequilibrium solution. Then subroutine NØNEQ is called to generate the nonequilibrium flow solution.

3.1.13 Diagnostic Dumps

If a .TRUE. value of the error indicator ERR is detected anywhere in MAIN, control is immediately transferred to the statement CALL DUMPEX, which prints out the current values of a large number of common variables as an aid to identifying the cause of failure. If data have been written on tape 8, the tape is backspaced to eliminate the records written during the failed case.

3.1.14 Job Termination

At the end of the calculations for each case, a line of output is written containing the case number, the number of records written on tape 8 (if any), and the total number of records written thus far in the job. Then the value of the job termination control parameter ISW4A is checked. If ISW4A is nonzero, the input data for the next case are read. If ISW4A is zero, a summary of the completed and failed cases in the job is written and, if DATAPE is .TRUE. an end file mark is written on tape 8 and the tape is rewound.

3.2 Subroutine AESØLN

This subroutine calculates the effective area ratio at a given position, X, in the nozzle. The value X is a subroutine

argument. Other input data to the routine are contained in the common block /AEGEØM/, namely, the geometric area ratio S1, the derivative of the geometric area ratio S2, and the square root of the geometric area ratio SQRTA. To set S1 and S2, a call to AESØLN(X) must be preceded by a call to GEØMAR(X,S1,S2). In addition, if NPRFLS = 1 (the flow is in a nozzle, not a channel) and JDIM = 1 (the nozzle is axisymmetric), then SQRTA must be computed in the calling routine before AESØLN is called.

The calculations of the effective area ratio are based upon equations I(126), I(130) and I(134), respectively, for the cases of a two-dimensional nozzle (NPRFLS = 1, JDIM = 0), an axisymmetric nozzle (NPRFLS = 1, JDIM = 1), and a rectangular channel (NPRFLS = 2). The effective area ratio obtained is stored in the location S1 of common block /AEGEØM/, replacing the geometric area ratio.

In addition, the subroutine computes the derivative, dA_e/dx, of the effective area ratio, using the following formulas:

2D Nozzle

$$\frac{dA_e}{dx} = \frac{\frac{dA_g}{dx} - \frac{1}{y_0}\frac{d\delta^*}{dx}}{1 - \delta^*_* / y_0} \qquad (4)$$

Axisymmetric Nozzle

$$\frac{dA_e}{dx} = \sqrt{A_e}\left[\frac{\frac{1}{\sqrt{A_g}}\frac{dA_g}{dx} - \frac{2}{y_0}\frac{d\delta^*}{dx}}{1 - \delta^*_* / y_0}\right] \qquad (5)$$

Channel

$$\frac{dA_e}{dx} = \frac{(z - \delta^*_2)\left(\frac{dy}{dx} - \frac{d\delta^*_1}{dx}\right) + (y - \delta^*_1)\left(\frac{dz}{dx} - \frac{d\delta^*_2}{dx}\right)}{(y_0 - \delta^*_{1*})(z_0 - \delta^*_{2*})} \qquad (6)$$

These relations can easily be derived from equations I(126), I(130), and I(134). The computed value of dA_e/dx is stored in the location S2 of common /AEGEØM/, where it replaces the derivative of the geometric area ratio.

3.3 Subroutine AGSØLN

The purpose of this subroutine is to solve for the geometric area ratio A_g when the effective area ratio A_e and boundary layer displacement thickness δ^* are given. These input data are provided in the argument list: A_e = AE, δ_i^* = DEL(i), where the index i runs from 1 to 2 in the case of a channel. The argument list also includes the indicator UPDØWN, which specifies whether an upstream (-1.) or downstream (+1.) solution is desired. The computed value of A_g, and the corresponding value of x, are returned to the calling routine through the arguments AG and X, respectively.

The solution for A_g is straightforward and explicit in the case of a nozzle. It is given by equation I(127) if the flow geometry is two-dimensional and by I(131) if it is axisymmetric. In both cases, the value of X corresponding to A_g is determined by calling FINDX.

In the case of a channel, the relation between A_e and A_g is given by equation I(135) and the profile functions y(x), z(x). This relation does not permit an explicit solution for A_g. In AGSØLN, a numerical Newton-Raphson method is used to determine A_g. Equation I(135) is written as

$$A_g = A_c + \frac{y\,\delta_2^* + z\,\delta_1^*}{y_0 z_0} \qquad (7)$$

where

$$A_c = \left(1 - \frac{\delta_{1*}^*}{y_0}\right)\left(1 - \frac{\delta_{2*}^*}{z_0}\right) A_e - \frac{\delta_1^*\,\delta_2^*}{y_0\,z_0} \qquad (8)$$

The quantity A_c and the factors δ_1^*/y_0z_0 and δ_2^*/y_0z_0 are constants during the solution for a particular call to the subroutine. From (7), the object of the calculation is to find a value of x such that F(x) = 0, where

$$F(x) \equiv A_c + \frac{y(x)\ \delta_2^* + z(x)\ \delta_1^*}{y_0 z_0} - A_g(x) \qquad (9)$$

Let x, F denote the values of these quantities from the current iteration and x_o, F_o the values for the previous iteration. Then, based on a linear approximation to (9), the value of x required to make F(x) = 0 is estimated to be

$$x_n = x - F \cdot \frac{x - x_o}{F - F_o} \qquad (10)$$

The iteration based on this relation begins at statement 60 in the subroutine. To start the iteration, A_g is set equal to A_e, and the corresponding x is computed using FINDX. The second point is obtained by computing A_g from (7) and again using FINDX. Equation (10) is then used in the third and all subsequent steps. The convergence criterion is that $|F|/A_g$ be less than or equal to 10^{-5}, or that $F = F_o$.

If $A_g(x)$ in (9) is expressed by means of equation I(120), equation (9) can be rewritten in the form

$$F(x) = \left(1 - \frac{\delta_{1*}^*}{y_0}\right)\left(1 - \frac{\delta_{2*}^*}{z_0}\right) A_e - \left(\frac{y}{y_0} - \frac{\delta_1^*}{y_0}\right)\left(\frac{z}{z_0} - \frac{\delta_2^*}{z_0}\right) \qquad (11)$$

If y and z both have minima at the throat x = 0, as is normally the case in channels, then the maximum value of F(x) occurs at x = 0 and is equal to

$$F_{max} = F(0) = \left(1 - \frac{\delta_{1*}^*}{y_0}\right)\left(1 - \frac{\delta_{2*}^*}{z_0}\right) A_e - \left(1 - \frac{\delta_1^*}{y_0}\right)\left(1 - \frac{\delta_2^*}{z_0}\right) \qquad (12)$$

In order for a solution of the equation $F(x) = 0$ to exist, it is necessary that F_{max} be greater than or equal to zero. This is always true if $\delta_1^* = \delta_{1*}^*$ and $\delta_2^* = \delta_{2*}^*$. Also, in practice, it is always true when $A_e \gg 1$. However, when A_e is nearly equal to 1 and $\delta_1^* < \delta_{1*}^*$, $\delta_2^* < \delta_{2*}^*$, F_{max} can be negative. These circumstances can arise near the sonic point (where $A_e = 1$) when the sonic point is displaced from the geometric throat by the presence of large gradients in the displacement thicknesses. When this occurs, the iteration formula (10) extrapolates past the throat and gives an x_n of the wrong sign, i.e., the opposite sign to UPDØWN. When such a condition is detected, AGSØLN attempts to "fix" the problem by resetting the displacement thicknesses δ_1^*, δ_2^* to their values at the sonic point, δ_{1*}^* and δ_{2*}^* respectively, and restarting the iterative solution at the beginning. Under the circumstances in question, this adjustment can have little effect on the value of A_g computed (which is near unity). Equation (12) shows that after this replacement, F_{max} is the product of a positive factor and $(A_e - 1)$, and hence is ≥ 0. Thus, if the problem was indeed caused by the combination of circumstances described above, the "fix" should guarantee the existence of a solution of the equation $F(x) = 0$. The coding provides a dump and an error exit should the problem recur after the "fix" has been carried out. In this case, the problem has a different cause, such as an error in a user-supplied channel profile curvefit.

3.4 Subroutine AXFIT

This subroutine solves for the value of x corresponding to a given effective area ratio (AFNTS) during the initial (perturbation) portion of the nonequilibrium solution. If the solution is being calculated without the boundary layer (ISW3B = 0), then the geometric area ratio is equal to the effective area ratio, and x (denoted by CX) is determined by calling subroutine FINDX. If the boundary layer is included in the solution, but subroutine THRØAT has not yet been called to reset IUPD from 1 to 0, the effects of the boundary layer displacement thickness on the effective flow geometry are neglected, and x is again obtained from FINDX. However, if the boundary layer is included and IUPD = 0, then the inviscid flow is coupled to the boundary layer and

x is accordingly determined by calling AGSØLN. Both FINDX and AGSØLN require an indicator which specifies whether the upstream (UPDØWN = -1) or downstream (UPDØWN = 1) solution for the x corresponding to the effective area ratio AFNTS is desired. In AXFIT, this indicator is determined by testing whether the nondimensional temperature CT is greater than or less than its value CTMAX at the geometric throat based on the equilibrium flow solution.

3.5 Subroutine AXSECT

Subroutine AXSECT is called by subroutine XSECT for the purpose of adding data for additional species pairs to the edited cross section data in common block/TRANS7/. The first argument LL of the subroutine gives the step in the cross section computations at which the additional pairs are to be added, while the second argument NN gives the total number of species to be considered in adding the data. The indices of the actual species to be considered in the added data are then given, in numerical order, by the first NN entries of the variable I in common block/TRANS2/. Subroutine AXSECT searches through all pairs (II, JJ) of these species with II $\leq$ JJ, to find those for which cross section data have not been previously specified, as indicated by the fact that Q(1,II, JJ) = 0. All such pairs are then added to the IQ, JQ arrays, starting at the location NQ(LL) + 1, with the original data in these locations being moved back to make room for the new values. Further, the subroutine sets Q(1,II, JJ) = 1. for all pairs added to the IQ, JQ arrays and revises the NQ array to be consistent with the new IQ, JQ values, assuming that all of the added pairs are to be included in the LLth step of the cross section computations.

3.6 Subroutine BLAYER

The boundary layer calculations in NATA are carried out by the method described in Section 5 of Volume I (ref. 1). These calculations are performed by subroutine BLAYER. The boundary layer properties are calculated from the integral I, equation I(172). This integral is used to determine the correlation parameter n, equation I(174). The momentum thickness

θ, displacement thickness δ^*, shear stress τ_w, and heat flux q_w are then calculated from n and other quantities.

The overall structure of BLAYER is diagrammed in figure 6. The calculations are all contained in a DØ loop with index L running from 1 to NPRFLS. In the case of a nozzle (NPRFLS=1), the calculations are executed only once. For a channel (NPRFLS=2), they are executed twice to compute the boundary layers on the two pairs of channel walls.

The subroutine argument FINAL is a logical variable specifying whether the new flow point is an actual point of the solution or merely an intermediate step in the Runge-Kutta integration used to generate the nonequilibrium flow solution. For FINAL = .FALSE., the important variables used in BLAYER are restored to their values at the beginning of the step before the RETURN is executed, and the calculations of shear stress TAUW and heat flux QWDØT are omitted.

IPØINT is a counter for the flow points at which boundary layer calculations are done in each flow solution. IPØINT is initialized to 1 in subroutine FRØZEQ for frozen and equilibrium solutions and in subroutine NØNEQ for nonequilibrium solutions. Each time BLAYER is called with FINAL = .TRUE., IPØINT is incremented by 1 before the RETURN.

Some of the physical variables which are used frequently are given short mnemonic names which are equivalenced to their standard names in unlabelled common

3.6.1 Initializations

When BLAYER is called for the first time in a given flow solution (IPØINT = 1), a number of quantities are set or initialized by the statements down to 90. These quantities include the coefficient B in the linear relation I(164) between the momentum parameter N and the correlation parameter n, and the transport properties PRW = Pr_w and VISCW = μ_w at the wall temperature. The wall is assumed to be catalytic; hence, these transport property calculations are based on the mole fractions QPJ(K) for the cold gas mixture. This section of the routine also sets the indicator

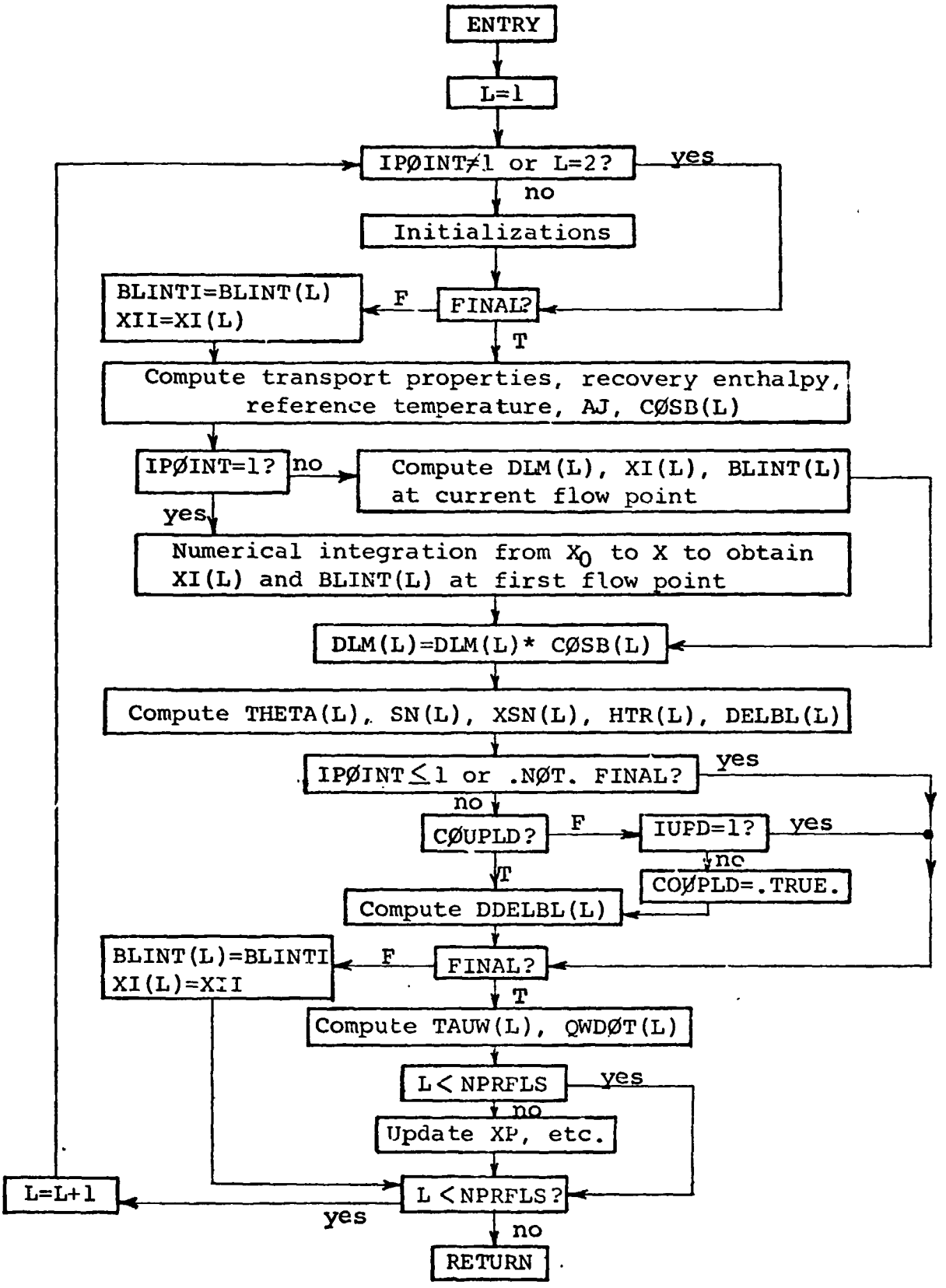


Figure 6. Flow Chart of Subroutine BLAYER

ITYPE, which specifies the type of nozzle geometry:

ITYPE = 1	Two-dimensional nozzle
ITYPE = 2	Axisymmetric nozzle
ITYPE = 3	Rectangular channel

3.6.2 Transport Property and Geometry Calculations

The section of BLAYER from statement 90 down to 270 computes the values of various physical quantities at the current flow point, including the transport properties and geometric parameters entering into the boundary layer calculation.

BLAYER uses the viscosity and Prandtl number at three temperatures, the wall temperature TWALL, the free-stream temperature, TE, and the reference temperature TREF. The calculations for TWALL are done only once per solution because the wall temperature is assumed constant. To save computing time, the transport property calculations at TE and TREF are not repeated at every flow point, but are done only when the temperature or the gas mole fractions have changed significantly. The criterion for performance of the transport property calculations is a change of 0.01 in any of the mole fractions or a change of 0.5 percent in the temperature. The mole fractions at which calculations were last done are represented by XTEST(J). The corresponding temperatures are TEP and TREFP.

The factor r^{2j} in equation I(171) is denoted by AJ in BLAYER. Because of the form of equations I(171) to I(173), r^{2j} can be replaced by a constant times r^{2j} without affecting the computed results for the boundary layer thicknesses, the heat flux, and the shear stress. For two-dimensional nozzles, AJ is set to 1, which is the value of r^{2j} for $j = 0$. For axisymmetric nozzles, AJ is set to the geometric area ratio,

$$A_g = (y/y_0)^2 \qquad (13)$$

which is r^{2j}/y_0. In the case of a rectangular channel, for each profile, AJ is set to the square of the radius of the <u>other</u> profile, since this radius is proportional to the width of the surface whose boundary layer is being calculated.

For nozzles, the cosine of the angle b between the profile tangent and the axis is calculated from dA_g/dx. In general,

$$\cos b = \frac{1}{\sqrt{1 + \tan^2 b}} = \frac{1}{\sqrt{1 + (dy/dx)^2}} \qquad (14)$$

where y is the profile ordinate. In the case of a two-dimensional nozzle, from equation I(118),

$$\frac{dy}{dx} = y_0 \frac{dA_g}{dx} \qquad (15)$$

and hence

$$\cos b = \frac{1}{\sqrt{1 + y_0^2 (dA_g/dx)^2}} \qquad (16)$$

In the axisymmetric case, from I(116),

$$\frac{dy}{dx} = \frac{y_0}{2\sqrt{A_g}} \frac{dA_g}{dx} \qquad (17)$$

and thus

$$\cos b = \frac{1}{\sqrt{1 + \frac{y_0^2}{4 A_g} \left(\frac{dA_g}{dx}\right)^2}} \qquad (18)$$

The derivative dA_g/dx in (16) and (18) is obtained by calling subroutine GEØMAR. In the case of a channel, dy/dx is computed by calling the entry GMAR3 of GEØMAR, and cos b is evaluated directly from (14).

The viscosity-temperature exponent ω, defined by equation I(191), is determined by fitting I(191) to the viscosity at the wall and reference temperatures. The coefficient A in I(164) (denoted by EK1) is calculated from I(201) at the first flow point (IPØINT = 1), and then held constant through the solution.

The integrand Φ of I(172) is represented by ØRDIN(L). Actually, ØRDIN(L) differs from equation I(171) by a constant factor, which cancels out in the calculation of n' I(173) and thus has no effect on the boundary layer properties.

3.6.3 Integration from x_0 to First Flow Point

In general, the flow point at which BLAYER is first called lies upstream of the throat. The Mach number at this point is low. However, the point may be well downstream of the point x_0 at which boundary layer development is assumed to start (see Sec. 5.10 of Volume I). Accordingly, the interval from x_0 to the first point of the flow solution is not treated as a single step for the purpose of the boundary layer calculation. Instead, when BLAYER is called with IPØINT = 1, the integral I(172) (denoted by BLINT(L)) and the streamwise boundary layer coordinate ξ (denoted by XI(L)) are evaluated by a 10-step numerical integration from x_0 to the first flow point.

In the region covered by this integration, the Mach number is low so that the flow can be approximated as incompressible. Also, the boundary layer is thin in this region, in comparison with the nozzle or channel radius, so that the effective area ratio can be approximated by the geometric area ratio. Thus, the continuity equation $\rho u A_e' = \text{constant}$, can be approximated

$$M A_g = \text{constant} \qquad (19)$$

The constant in (19) is represented by CC. At each point in the integration from x_0 to x_1, the geometric area ratio A_g = AG1 and the quantity AGJ corresponding to r^{2j} are calculated by calling subroutine RADIUS, M is obtained from (19), and a quantity ØRD1 proportional to ØRDIN(L) is computed with the

approximation $a_e/a_0 \simeq p_e/p_0 \simeq 1$. The increment in ξ is computed using the Pythogorean theorem:

$$(\Delta \xi)^2 = (\Delta x)^2 + (\Delta y)^2 \qquad (20)$$

where Δy is the change in the profile ordinate due to a change Δx in the axial coordinate. BLINT(L) is calculated using trapezoidal-rule integration. At the first flow point, DLM(L) = $d\ell n\, M/dx$ is evaluated from (19).

3.6.4 Derivatives of the Mach Number

The derivative dM_e/dx of the flow Mach number is an important variable in the boundary layer calculation because it represents the effect of a streamwise pressure gradient. The related quantity $dM_e/d\xi$ appears in the formula I(174) for the correlation parameter, n. All of the boundary layer properties depend upon n.

The derivative $dM_e/d\xi$ differs between the two profiles of a channel because the boundary layer coordinate lies along the channel surface and the two profiles have different inclinations to the nozzle axis. The value of $d\ell n\, M_e/d\xi$ for the L^{th} profile is denoted by DLM(L). It is determined by first computing $d\ell n\, M_e/dx$ and then correcting for the inclination of the nozzle or channel surface to the axis. For each flow point beyond the first, $d\ell n\, M_e/dx$ is calculated from rate data if the flow solution is of nonequilibrium type and the numerical integration technique is being used (INEQ = 1). Otherwise, if the flow solution is of frozen or equilibrium type or if the perturbation technique is being used in a nonequilibrium solution (INEQ = 0), $d\ell n\, M_e/dx$ is calculated by numerical differentiation. If the inviscid solution is being calculated by nonequilibrium integration, DLM(L) is calculated from equations I(420), I(421), and I(291) or I(422). In the case of a chemical nonequilibrium gas model (NT = 1), the stagnation enthalpy h_0 is constant and hence I(421) and I(291) give

$$\frac{du}{dx} = -\frac{1}{u} \sum_{j=1}^{n} \left[\gamma_j \frac{dH_j}{dx} + H_j \frac{d\gamma_j}{dx} \right] \qquad (21)$$

The variables used in the code are

$$\text{SHJ(j)} = \frac{H_j}{R_C T_0} \tag{22a}$$

$$\text{CCPJ(j)} \cdot \text{DT} = \frac{1}{R_0 T_0} C_{pj} \frac{dT}{dx} = \frac{1}{R_0 T_0} \frac{dH_j}{dx} \tag{22b}$$

$$\text{SU} = u\sqrt{\frac{W_0}{R_0 T_0}} \tag{22c}$$

$$\text{GJ(j)} = \gamma_j \tag{22d}$$

Hence (21) is coded as

$$\text{DUDX} \equiv \frac{d(\text{SU})}{dx} = -\frac{W_0}{\text{SU}} \sum_{j=1}^{n} \left[\text{GJ(j)}\cdot\text{CCPJ(j)}\cdot\text{DT} + \text{SHJ(j)}\cdot\text{DGJ(j)}\right]$$

$$= -\frac{\text{CMA}}{\text{SU}} (\text{SCPG}\cdot\text{DT} + \text{SHJDGJ}) \tag{23}$$

The sums denoted by SCPG and SHJDGJ are obtained from common. Similarly, in the case of an electronic nonequilibrium model (NT = 2), equations I(421) and I(422) give

$$\text{DUDX} = \frac{\text{DCHA}}{\text{SU}} - \frac{\text{CMA}}{\text{SU}} \left[\text{GJ(1)}\cdot\text{CCPJ(1)}\cdot\text{DTE} + \text{SCPGH}\cdot\text{DT} + \text{SHJDGJ}\right] \tag{24}$$

where

$$\text{DCHA} = \frac{W_0}{R_C T_0} \frac{dh_0}{dx} \tag{25}$$

$$\text{SCPGH} = \sum_{j=2}^{n} \text{GJ(j)} \cdot \text{CCPJ(j)} \tag{26}$$

In (24), quantities subscripted (1) refer to the electron species.

For equilibrium and frozen solutions, and for nonequilibrium solutions which are being developed by the perturbed equilibrium flow technique (INEQ = 0), the quantities SCPG and SHJDGJ are not available. In these cases, DLM(L) is calculated by numerical differentiation. For IPØINT = 2, a simple first-order difference expression

$$\frac{dM}{dx} = \frac{M-M_p}{\Delta x} \tag{27}$$

is used, where M is the Mach number at the current point and M_p that at the previous point. For IPØINT $\geq$ 3, the following second-order difference calculation is used: Let (M,x) be the data for the current point (M_p,x_p) those for the previous point, and (M_{pp},x_{pp}) those for the point preceding the previous one. A quadratic fit to these data can be written in the form

$$M = c_0 + c_1 x + c_2 x^2 \tag{28}$$

from which

$$\frac{dM}{dx} = c_1 + 2c_2 x \tag{29}$$

Substitution of the three data points into (28) gives, after elimination of c_0,

$$\frac{M-M_p}{x-x_p} = c_1 + c_2 (x + x_p) \tag{30a}$$

$$\frac{M-M_{pp}}{x-x_{pp}} = c_1 + c_2 (x + x_{pp}) \tag{30b}$$

Hence

$$c_2 = \frac{\frac{M-M_p}{x-x_p} - \frac{M-M_{pp}}{x-x_{pp}}}{x_p - x_{pp}} \qquad (31)$$

Now that c_2 is known, c_1 can be obtained from (30b);

$$c_1 = \frac{M-M_p}{x-x_p} - c_2 \ (x + x_p) \qquad (32)$$

dM/dx is then calculated from (29), and DLM(L) is obtained from

$$\frac{d\,\ell n\ M}{dx} = \frac{1}{M}\ \frac{dM}{dx} \qquad (33)$$

3.6.5 Calculation of Boundary Layer Thicknesses

Next, the increase in ξ in the step from the previous flow point to the current one is calculated,

$$\Delta \xi = \frac{\Delta x}{\cos b} \qquad (34)$$

and ξ is incremented by $\Delta \xi$. The boundary layer integral I (equation I(172)), denoted by BLINT(L), is updated using the trapezoidal integration rule. The derivative $d\ell n\ M/dx$ is converted into $d\ell n\ M/d\xi$ by

$$\frac{d\,\ell n\ M}{d\xi} = \cos b\ \frac{d\,\ell n\ M}{dx} \qquad (35)$$

Then n', equation I(173), is evaluated, and the correlation parameter n is calculated from I(174). The characteristic length L in these equations is taken to be R0, defined as the profile radius in the case of a nozzle, and defined by

$$\pi\,(R0)^2 = \text{throat area} \qquad (36)$$

in the case of a channel. The momentum thickness is calculated from I(184). If the flow geometry is that of an axisymmetric nozzle, the transverse curvature correction I(187) is applied. The smoothed correlation parameter $\bar{n}$, denoted by XSN(L), is computed from equation I(217).

The effect of free stream Mach number upon the correlations of boundary layer properties is contained in the dependence upon the hypersonic parameter σ, equation I(190). The correlations used in BLAYER are based upon the Dewey-Gross calculations (ref. 5) for a perfect gas. In frozen and nonequilibrium solutions, the parameter σ as calculated from I(190) never approaches unity, even at high Mach numbers, because of the frozen-in dissociation energy in h_0. To represent the Mach number effect more realistically, the σ defined by I(190) is replaced by an effective value σ_e, defined as

$$\sigma_e = \left[1 + \frac{2}{(\gamma_e - 1)\, M^2}\right]^{-1} \qquad (37)$$

where γ_e is an effective specific heat ratio. For a perfect gas, (37) is equivalent to I(190). In BLAYER, γ_e is calculated as

$$\gamma_e = a^2 W / R_0 T \qquad (38)$$

where W is the mixture molecular weight and a the sound speed.

The incompressible form factor HTR(L) is computed from I(189) and the displacement thickness DELBL(L) from I(188). In the case of a nonequilibrium flow solution, the coupling of the boundary layer to the inviscid flow is switched on when the indicator IUPD is set to zero in subroutine THRSHAT.* The only effect of the coupling in subroutine BLAYER is that the derivatives DDELBL(L) of the displacement thicknesses

*IUPD is always equal to 1 in the frozen and equilibrium solutions.

have to be calculated. To avoid excessive disturbance of the nonequilibrium solution, DDELBL(L) is gradually built up from zero over 29 steps in the flow solution, as explained in Section 7.6 of Volume I.

3.6.6 Heat Flux and Shear Stress

For FINAL = .FALSE., the boundary layer integral BLINT(L) is reset to its value BLINTI at the beginning of the step, and the boundary layer coordinate XI(L) to its initial value XII. Then control is transferred to statement 540, the limit of the DØ loop over L.

For FINAL = .TRUE., the shear parameter XL is calculated from the curvefit I(194), the shear stress TAUW(L) from I(193), the Reynolds analogy factor RAF from I(200), and the heat flux QWDØT(L) from I(195). Then, if L is less than NPRFLS, control is transferred to the end of the DØ loop over L. If L is equal to NPRFLS, several physical quantities are saved for use in the calculations for the next step. When the DØ loop over L has been completed, diagnostic outputs are written if the control variable ISW4B is nonzero, and the RETURN is executed.

3.7 Subroutine BLCALL

During the nonequilibrium solution, all calls to the boundary layer routine BLAYER are made through subroutine BLCALL. Before calling BLAYER, subroutine BLCALL sets up certain data required by BLAYER and by the transport property routines. The species mole fractions are computed from equation I(1) and loaded into the common array SAVEC(J). If the perturbation technique is still being used (INEQ = 0), the perturbations PERTGJ(J) are added to the concentrations γ_j = GJ(J) before equation I(1) is used. In this case, in addition, the temperature perturbation PCT is added to the nondimensional temperature CT and the density perturbation PRHØ is added to RHØ. After BLAYER has been called, the temperature and density are restored to their unperturbed (equilibrium) values. It is not necessary to reset SAVEC(J), because this array is used only in the transport, boundary layer, and model condition calculations, not in the inviscid

flow solution. Indeed, subroutine PRTA (which controls the output of the nonequilibrium flow solution) calls BLCALL to load the SAVEC array even in solutions neglecting the boundary layer (ISW3B = 0), because the output routine ØUT1 obtains the mole fractions from SAVEC.

3.8 Subroutine BXSECT

Subroutine BXSECT is used in searching the array V in common block/TRANS7/for references to the indices of species pairs used in the cross section computations, and also in determining the correspondence between the entries in the V array and the steps in the cross section computations. The argument L of the subroutine is the index of the step in the cross section computations, MV is the index of the last entry in the V array used in step L of the computations, and I and J are the indices of the species pair referenced in V. When the subroutine is called with L = 0, the other arguments MV, I, and J are set internally by the subroutine and need not be furnished by the calling routine; however, all arguments are required when L $\neq$ 0.

In use, subroutine BXSECT is first called with L = 0 to begin the search over the V array. The routine then goes through each step L of the cross section computations, starting with L = 1 and checking for steps with KQ = 9 or 12. The value of MV is accumulated at each step L of the computations from the number of V values NV(KQ(L)) required for that step, where NV(KQ) is a fixed value which has been preset in the code. When a step with KQ = 9 or 12 is found, subroutine BXSECT sets I and J to the indices of the species pair referenced in the V array for that step and returns control to the calling program. The latter program may then reset I and J to any desired values and recall subroutine BXSECT again with the current values of L and MV to enter the revised data in the V array and resume the search of the computations again from the point at which it was left off. It should be noted that for each step with KQ = 12,there are two species pairs referenced in the V array, and returned by subroutine BXSECT, while for KQ = 9 there is only one.

When all steps of the computations have been searched, as indicated by the index L of the step becoming equal to the number of steps NKQ in the edited cross section data in common block /TRANS7, the subroutine sets L to 0, as a signal that the search has been completed, and returns control to the calling program.

3.9 Subroutine CØMM

The function of subroutine CØMM is to compute some physical quantities upon which the rates of change of the species concentrations and the other dependent variables depend. The actual calculations of the rates of change are then carried out in subroutine EXACT.

CØMM is called only from subroutine DERIVS, and the call to CØMM is always preceded by a call to subroutine THERM, which computes the species properties at the current nondimensional temperature, $CT = T/T_0$. In the case of an electronic nonequilibrium model, DERIVS calls THERM twice to obtain the required properties at both the gas temperature CT and the electron temperature $CTE = T_e/T_0$.

To facilitate the discussion, the correspondence of some of the Fortran symbols with the mathematical notation used in Volume I is reviewed in the following table. Symbols not listed here are defined in Section 4.77.

ALGJ(J)	$\ell n \gamma_j$
BETA(I,J)	β_{ij}
CHI(I)	χ_i
CLKF	$\ell n\, k_f$
CLKR	$\ell n\, k_r$
CLNPI(I)	$\ell n\, P_i$
CLN1MC(I)	$\ell n\, (1 - \chi_i)$
CLNTD(L)	$\ell n\, T$ for L = 1; $\ell n\, T_e$ for L = 2

CLTBF	$\ln(\rho \sum \gamma_j)$, sum over third bodies
CLZ2	$\ln(R_0 T)$ or $\ln(R_0 T_e)$, argument in cm^3 atm/mole
CLZ3	$\ln(\rho R_0 T)$ or $\ln(\rho R_0 T_e)$, argument in atm-g/mole
CTD(L)	T/T_0 for L = 1; T_e/T_0 for L = 2
EF	ϵ_{fi}
ER	ϵ_{ri}
GJ(J)	γ_j
PI(I)	P_i
PICHI(I)	$P_i \chi_i$
QDPE	$\dot{q}_e$
QDPR	$\dot{q}_r$
QF	q_{fi}
QR	q_{ri}
RHØP	ρ
RHPL	$\ln \rho$
SCPGH	$\frac{1}{R_0} \sum_{j=2}^{n} \gamma_j c_{pj}$
SU	u/u_s
TEP	T_e
T6	$\sum_{n=1}^{n} \nu_{ij} \ln \gamma_j$ or $\sum_{j=1}^{n} \nu'_{ij} \ln \gamma_j$

T	$\sum \nu_j$, sum over third-body species
T8	$\sum_{j=1}^{n} \nu'_{ij}$
T9	$\sum_{j=1}^{n} \beta_{ij} \left[\ln (R_0 T') + \frac{\mu_j^{\circ}(T')}{R_0 T'} \right]$
	where T' is T or T_e
XMJATD(J,L)	$\mu_j^{\circ}(T')/R_0 T'$, where T' = T for L = 1, T' = T_e for L = 2
XNULJ(I,J)	ν_{ij}
XNULJP(I,J)	ν'_{ij}
Z1	$\ln (1/u)$, for u in cm/sec

Subroutine CØMM may be divided, for purposes of explanation, into three parts: an initialization section, a main loop over the reactions, and a final section. These parts of the subroutine will now be discussed.

3.9.1 Initialization Section

This section extends from the subroutine entry point through the statement RCALC = .FALSE. (card CØM 107). It sets up the ALGJ array, computes a number of sums which are required later in CØMM or in other parts of NATA, calculates the nondimensional flow velocity SU from the energy equation I(245), and calls subroutine GEØM to compute the gas density.

In addition, it contains the controls for intermittent dumping of diagnostic data in the nonequilibrium routines. The indicator for these dumps is the integer variable ISW5B in blank common. ISW5B is preset to zero. For the zero

value, no dumps are produced. If ISW5B is set in the input to a positive value, then dumps are produced each time CØMM, EXACT, RNKT, or PRTA is called. These dumps provide a detailed trace of the operation of the nonequilibrium integration, but give several pages of printed output for each integration step. Thus, it is hardly feasible to run a complete solution with continuous dumps. However, operation of the nonequilibrium routines at late stages in the solution can be studied by calling for intermittent dumps. This is done by setting ISW5B to a negative value in the input. Upon the first entry into CØMM, the variable ICYCLE is then reset from its preset value of 90000000 to |ISW5B| . The counter ICØUNT is incremented by 1 every time CØMM is entered. Whenever ICØUNT is an exact multiple of ICYCLE, ISW5B is reset to 1; otherwise, it is reset to 0. Thus, upon every ICYCLEth entry into CØMM, the dumps are switched on and are produced in all of the nonequilibrium routines until CØMM is entered the next time, when they are switched off again. The coding of these controls is in the first nine executable statements of CØMM.

3.9.2 Main Loop Over the Reactions

The main calculations in CØMM are performed in a large DØ loop headed by the statement, DØ 460 I = 1, ISR (card CØM 109). Figure 7 is a flowchart of this main loop. In essence, this loop performs only two functions:

1) First, it computes the quantities $P_i \chi_i$ appearing in the rate equations I(287).

2) Second, in the case of an electronic nonequilibrium model, it evaluates the energy transfer to the electron gas $\dot{q}_e$ and the radiative loss $\dot{q}_r$ due to the reactions.

The quantities P_i and χ_i are computed from equations I(288) and I(289), respectively, in logarithmic form:

$$\ell n\ P_i = \ell n\left(\frac{1}{u}\right) + \ell n\ k_{fi} + (\nu_i - 1)\ell n \rho + \sum_{j=1}^{n} \nu_{ij}\, \ell n \gamma_j \quad (39)$$

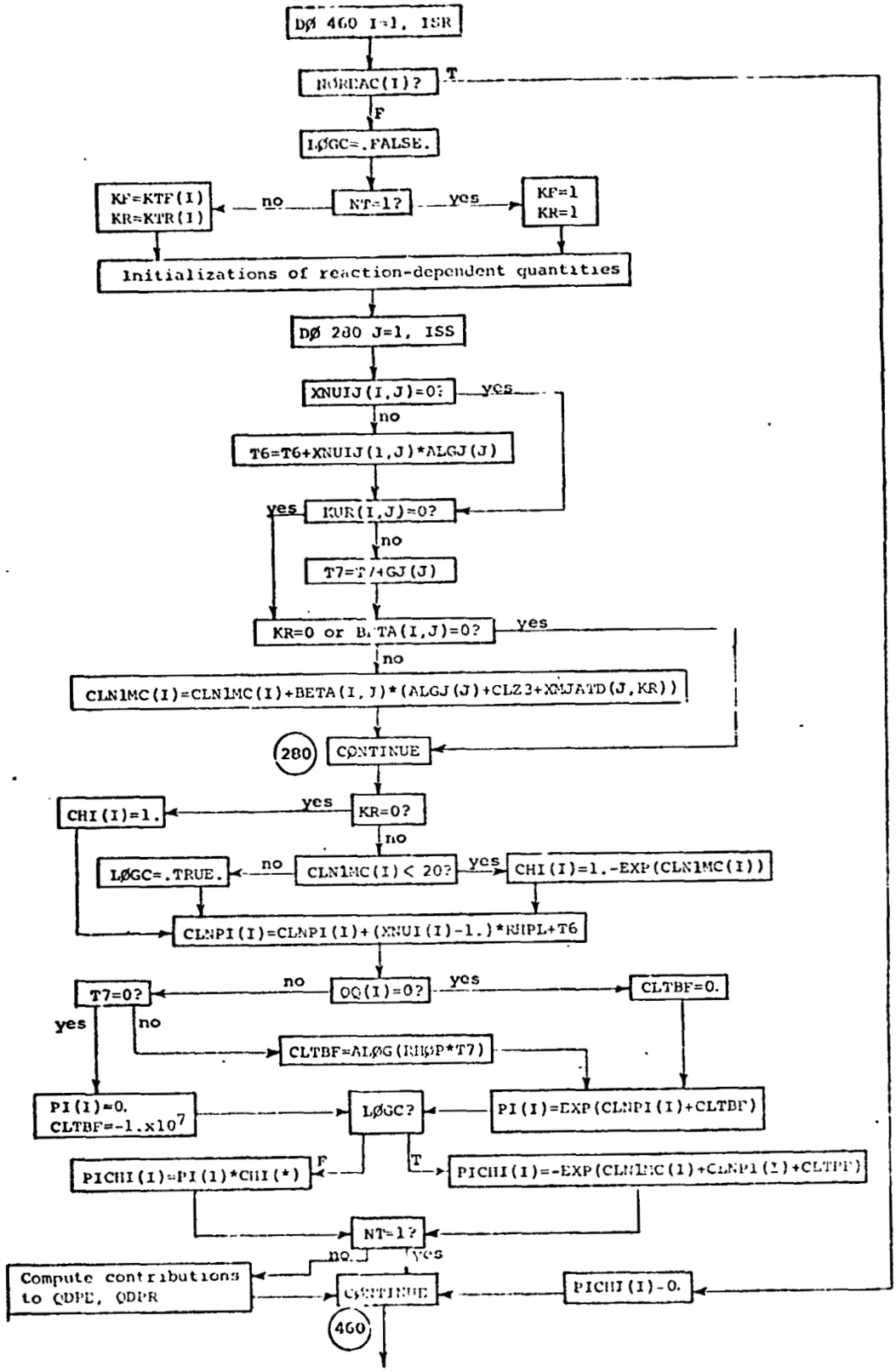


Figure 7. Flow Chart of Main Loop Over Reactions in Subroutine CØMM

$$\ell n\ (1 - \chi_i) = \beta_i \ell n\ (\rho R_0 T) + \sum_{j=1}^{n} \beta_{ij} \left[\ell n\ \gamma_j + \frac{\mu_j^o}{R_0 T} \right] \quad (40)$$

In equation (40), the equilibrium constant K_i in I(289) has been expressed in terms of more basic quantities using equation I(278). In the summations indicated in (39) and (40), terms in which the stoichiometric factor ν_{ij} or β_{ij} is zero are omitted.

If $\ell n\ (1 - \chi_i)$ is less than 20, P_i and χ_i are computed from (39) and (40) using

$$P_i = \exp\ (\ell n\ P_i) \quad (41)$$

$$\chi_i = 1 - \exp\left[\ell n\ (1 - \chi_i) \right] \quad (42)$$

and $P_i \chi_i$ is obtained by multiplication. On the other hand, if $\ell n\ (1 - \chi_i)$ is greater than or equal to 20, so that $(1 - \chi_i)$ is greater than 4.85×10^8, the term 1 in $(1 - \chi_i)$ is neglected and the product $P_i \chi_i$ is computed as

$$P_i \chi_i = - \exp\left[\ell n\ P_i + \ell n\ (1 - \chi_i) \right] \quad (43)$$

This algorithm allows the computation of a finite $P_i \chi_i$ value in situations where P_i is so small that an exponential underflow occurs in (41), but χ_i is negative and so large that $P_i \chi_i$ is within the range representable by floating-point numbers. It also makes possible a simple treatment of the special cases which arise when one or more of the γ_j are zero, as discussed below.

If a list of third-body (catalytic) species has been specified for the reaction, the expression I(288) for P_i is replaced by

$$P_i = \left[\frac{\rho^{\nu_i - 1}}{u}\ k_{fi}\ \prod_{j=1}^{n} \gamma_j^{\nu_{ij}} \right] \cdot \rho \sum_{tb} \gamma_k \quad (44)$$

in which the sum on the right is over the third-body species specified. The corresponding logarithmic expression is (39) with a term $\ell n\ (\rho \sum_{tb} \gamma_k)$ added to the right. The third bodies do not affect χ_i. Equation (44) may be derived from I(283) and I(287) by noting that, for a catalytic species (k), $\nu_{ik} = \nu'_{ik} = 1$. Thus, a factor $\rho\ \gamma_k$ can be removed from both terms on the right in I(283). If the i^{th} reaction involves more than one third-body species, each such species gives a separate three-body reaction. The effects of these reactions can be summed to give a single reaction with an effective third-body concentration which is the sum of the concentrations of the actual catalytic species, as in (44).

In the case of an electronic nonequilibrium gas model, there are two temperatures, the gas temperature T and the electron temperature T_e. The forward reaction rate k_{fi} can depend on either T or T_e, or both, or neither. The form of temperature dependence of k_{fi} is specified by the array KTF(I). If the indicator KF = KTF(I) is equal to 1, k_{fi} is given by I(69). If KF is equal to 2, the temperature T in I(69) is replaced by T_e. This procedure is coded by placing the two nondimensional temperatures into a two-element array CTD(L), and placing the logarithms of the two temperatures (expressed in degrees Kelvin) into an array CLNTD(L). Both cases are then treated by a single formula involving CTD(KF) and CLNTD(KF), i.e., statement 190 of CØMM, which evaluates $\ell n\ k_{fi}$ from

$$\ell n\ k_{fi} = \ell n\ C_i + \eta_i\ \ell n\ T_k - \frac{E_{ai}}{R_0 T_k} \qquad (45)$$

where T_k is equal to T for k = 1 and equal to T_e for k = 2. In (45), $\ell n\ C_i = \ell n A_i - \eta_i\ \ell n\ (10^4)$.

Three special forms for the forward reaction rate, used in the electronic nonequilibrium model for argon (Appendix A in Volume II), are indicated by KTF(I) = 3, 4, and 5. For KTF(I) = 3, k_f is assumed to be given by

$$k_f = C_i\ T_e^{\eta_i}\ (1 - e^{-E_i/R_0 T}) \qquad (46)$$

where E_i is a parameter with dimensions of energy. For KTF(I) = 4,

$$k_f = \frac{C_i \, T_e^{\eta_i}}{\max(1, \tau)} \tag{47}$$

where

$$\tau = b \, \rho \, \gamma_a \, R \tag{48}$$

Here R is the local nozzle radius (or equivalent radius in the case of a channel), and γ_a is the concentration of the product atom in the reaction. For KTF(I) = 5,

$$k_f = \frac{C_i}{\sqrt{R}} \tag{49}$$

In NATA, the reverse rate constant k_{ri} is calculated from the detailed-balancing relation I(277), using the equilibrium constant K_i, equation I(278). In electronic nonequilibrium gas models, the coding of CØMM allows for three types of temperature dependence of K_i. If the indicator KR = KTF(I) is zero, the reverse reaction is assumed to occur only at a negligible rate. If KR is 1, K_i is calculated from I(278) using the gas temperature T. If IR = 2, the temperature T in I(278) is replaced by T_e. The chemical potentials μ_j^o in I(278) are temperature dependent. During calculations based on an electronic nonequilibrium model, both $\mu_j^o(T)$ and $\mu_j^o(T_e)$ are computed, and these data are stored in the doubly dimensioned array XMJAT(J,L) with dimensions (20,2). The coding of equation (40) utilizes the array with the index L = KR so that both cases can be treated using a single formula, card CØM 153.

Cases in which one or more of the concentrations $\gamma_j = 0$ require some special consideration because of the appearance of $\ln \gamma_j$ in the formulas used in CØMM. In NATA, the γ_j are represented by the double precision array GJ(J). A zero

value for an element of this array could come about in one of two ways:

(1) One of the concentrations GJ(J) could underflow as a result of its decrease during the flow solution. However, this is most unlikely to occur in practice because, to underflow, GJ(J) must fall below 10^{-78} (on the IBM 360/75) or below 10^{-308} (on the UNIVAC 1108). Since GJ(J) can decrease by no more than 10 percent in a single integration step because of step size controls, the number of steps required to reach underflow from initial reservoir values of (say) 1.0^{-1} to 10^{-20} is of the order of 1000 or more, even on the IBM 360. Also, unless the control parameter GAMIN were set to zero, the decrease of GJ(J) would be stopped, long before underflow was reached, by the freezing of minor species (Section 7.5.3 of Volume I).

(2) A zero GJ(J) value could arise accidentally, at an intermediate point in the Treanor-Runge-Kutta integration step, as a result of a negative $\Delta\gamma_j$ which happened to be exactly equal in magnitude to the γ_j value at the start of the step. This is quite unlikely, but it could happen.

To allow for these rather remote possibilities, the coding of CØMM is designed to cope with zero concentrations if they should occur. If one of the species concentrations, γ_k, is zero, four special cases can arise, depending upon the values of the stoichiometric coefficients ν_{ik}, ν'_{ik} in the reaction I(219):

Case 1: $$\nu_{ik} = \nu'_{ik} = 0 \qquad (50a)$$

Case 2: $$\nu_{ik} = 0, \quad \nu'_{ik} > 0 \qquad (50b)$$

Case 3: $\nu_{ik} > 0, \quad \nu'_{ik} = 0$ (50c)

Case 4: $\nu_{ik} > 0, \quad \nu'_{ik} > 0$ (50d)

In the Cornell Aeronautical Laboratory program from which NATA was derived, each of these cases was treated separately by special coding. In the present version of the code, all of these cases are treated using the formulas (39) to (44) with the aid of a computational artifice.

Cases involving zero species concentrations γ_j require special consideration because of the appearance of $\ell n \gamma_j$ in the formulas (39) and (40). Any attempt of the program to evaluate ℓn 0 would lead to immediate termination of the run by the computer operating system. In subroutine CØMM, the quantities $\ell n \gamma_j$ are stored in an array, ALGJ(J). For species whose concentrations γ_j are zero, ALGJ(J) is set to a large negative number, L, whose magnitude is chosen in accordance with the following criteria:

(1) L must be sufficiently large in magnitude that, even when combined additively with other terms as indicated in (39) and (40), it is sufficiently negative to produce a guaranteed exponential underflow, i.e.,

$$\exp(L + \text{other terms}) = 0 \quad \text{(exactly)} \qquad (51a)$$

This will happen so long as the argument of the exponential function is less than about -88 on the UNIVAC 1108 or less than about -180 on the IBM 360/75.

(2) L must be sufficiently small in magnitude that the operation

$$L + \text{other terms} - L = \text{other terms} \qquad (51b)$$

gives a result for "other terms" containing several significant digits of accuracy. The "other

terms" in equations (39) and (40) are all natural logarithms of the values of physical quantities whose values lie roughly in the range 10^{-20} to 10^{20}; thus, the logarithms range from about -50 to +50. The second condition can thus be satisfied by taking L to be less than about 10^4 or 10^5. If L were chosen to be much larger, say 10^{10}, then the first addition in (51b) would give L + other term = L (exactly) on the UNIVAC 1108, because the change in L due to addition of the "other terms" would be less than the accuracy with which L can be represented as a floating point number (about 8 significant figures).

In subroutine CØMM, L has been chosen to be -7.00, a value which clearly satisfies both conditions.

Now assume that a particular concentration γ_k is zero. With $\ell n\, \gamma_k = L$, the formulas (39) and (40) treat the four cases (50) as follows:

In Case 1, ν_{ik} and β_{ik} are both zero, so that the term $\ell n\, \gamma_k$ does not actually appear in either (39) or (40). In this case, according to (50a), the species k whose concentration γ_k is zero does not participate in the reaction as either a reactant or a product.

In Case 2, $\ell n\, \gamma_k$ does not appear in (39), so that the evaluation of $\ell n\, P_i$ proceeds normally. However, $\beta_{ik} = \nu'_{ik}$ is positive, so that the right hand side of (40) contains a term $\nu'_{ik}\, L$ which is negative and large in magnitude. The resulting exponential underflow during the evaluation of (42) then gives $\chi_{\gamma_k^{\beta_{ik}}} = 1$. This is the correct result for this case, since $\gamma_k^{\beta_{ik}}$ is a zero factor in the second term on the right in equation I(289).

In Case 3, equations (39) and (40) can be written

$$\ell n\, P_i = \ell n\, (\tfrac{1}{u}) + \ell n k_{fi} + (\nu_i - 1)\ell n \rho + \sum_{j \neq k} \nu_{ij}\, \ell n\, \gamma_j + \nu_{ik}\, L \qquad (52a)$$

$$\ln(1-\chi_i) = \beta_i \ln(\rho R_0 T) + \sum_{j\neq k} \beta_{ij}\left[\ln\gamma_j + \frac{\mu_j^\circ}{R_0 T}\right] - \nu_{ik}\left(L + \frac{\mu_k^\circ}{R_0 T}\right) \tag{52b}$$

Here the term $-\nu_{ik} L$ on the right in (52b) is positive and very large; thus, equation (42) is not used and the product $P_i \chi_i$ is evaluated using (43), which gives

$$P_i \chi_i = -\exp\left\{\ln\left(\frac{1}{u}\right) + \ln k_{fi} + (\nu_i - 1)\ln\rho + \beta_i \ln(\rho R_0 T) + \sum_{j\neq k} \nu'_{ij} \ln\gamma_j + \sum_{j=1}^{n} \beta_{ij} \frac{\mu_j^\circ}{R_0 T}\right\} \tag{53}$$

$$= -\frac{k_{fi}\rho^{\beta_i+\nu_i-1}}{K_i} \prod_{j\neq k} \gamma_j^{\nu'_{ij}}$$

The terms involving L have cancelled out. The validity of equation (53) for Case 3 can be verified directly by multiplying equation I(288) and I(289), letting $\gamma_k \to 0$, and taking (50c) into account.

In Case 4, equations (39) and (40) give

$$\ln P_i = \ln\left(\frac{1}{u}\right) + \ln k_{fi} + (\nu_i - 1)\ln\rho + \sum_{j\neq k} \nu_{ij} \ln\gamma_j + \nu_{ik} L \tag{54a}$$

$$\ln(1-\chi_i) = \beta_i \ln(\rho R_0 T) + \sum_{j\neq k} \beta_{ij}\left[\ln\gamma_j + \frac{\mu_j^\circ}{R_0 T}\right] + (\nu'_{ik} - \nu_{ik})\left[L + \frac{\mu_k^\circ}{R_0 T}\right] \tag{54b}$$

Addition of these two equations gives

$$\ln [P_i(1 - \chi_i)] = \ln \left(\frac{1}{u}\right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \beta_i \ln (\rho R_0 T)$$

$$+ \sum_{j \neq k} \nu'_{ij} \ln \gamma_j + \sum_{j \neq k} \beta_{ij} \frac{\mu_j^o}{R_0 T} + \beta_{ik} \frac{\mu_k^o}{R_0 T}$$

$$+ \nu'_{ik} L \qquad (55)$$

Since $\nu'_{ik} L$ is negative and large in magnitude, (55) implies that

$$P_i(1 - \chi_i) = P_i - P_i \chi_i = 0 \qquad (56)$$

Also, (54a) and (41) give $P_i = 0$; thus, $P_i \chi_i = 0$. This result is correct for Case 4, as may be seen by setting one of the γ_k to zero in I(286) and taking (50d) into account.

The preceding analysis can easily be generalized to include cases in which more than one of the concentrations is zero. Thus, the artifice of setting $\ln \gamma_k = L$ when $\gamma_k = 0$ provides a satisfactory treatment of all cases that can arise.

In calculations for a chemical nonequilibrium model (NT = 1), the main loop provides only calculations of P_i and $P_i \chi_i$. In calculations for an electronic nonequilibrium model the loop also computes the contributions to the energy transfer to the electron gas (QDPE) and the radiative power loss (QDPR). These computations are provided by the coding below statement 390, and simply consist of evaluation of the formulas

$$(\Delta \dot{q}_e)_i = k_{fi} \epsilon_{fi} \prod_{j=1}^{n} (\rho \gamma_j)^{\nu_{ij}} - k_{ri} \epsilon_{ri} \prod_{j=1}^{n} (\rho \gamma_j)^{\nu'_{ij}} \qquad (57a)$$

$$(\Delta \dot{q}_r)_i = k_{fi} q_{fi} \prod_{j=1}^{n} (\rho \gamma_j)^{\nu_{ij}} - k_{ri} q_{ri} \prod_{j=1}^{n} (\rho \gamma_j)^{\nu'_{ij}} \tag{57b}$$

which are obtained by combining equations I(320) and I(321b,c). In equations (57), $(\Delta \dot{q}_e)_i$ and $(\Delta \dot{q}_r)_i$ denote the contributions of the i^{th} reaction to $\dot{q}_e$ and $\dot{q}_r$, respectively. The energy partition parameters ϵ_{fi}, ϵ_{ri}, q_{fi}, q_{ri} are obtained by calling subroutine EPART. Effects of third-body species, if any, are taken into account using the quantity CLTBF = $\ln (\rho \sum_{tb} \gamma_j)$, already evaluated during the computation of $P_i \chi_i$.

3.9.3 Final Section

After the main loop over the reactions has been completed, CØMM checks the value of INEQ. If this indicator is zero (i.e., if the perturbation technique is still being used to generate the flow solution), the RETURN is executed immediately. If INEQ is nonzero (i.e., if the numerical integration of the rate equations has been started), CØMM calculates the nondimensional pressure, PRES = p/p_0. If a chemical nonequilibrium model is being used, the pressure is computed from equation I(273) in the form

$$\frac{p}{p_0} = \frac{\rho}{\rho_0} \frac{T}{T_0} \frac{W_0}{W} \tag{58}$$

If an electronic nonequilibrium model is in use, equation I(323) is used to calculate the pressure in the form

$$\frac{p}{p_0} = \frac{\rho}{\rho_0} \cdot \frac{\gamma_e T_e + \gamma_h T}{T_0} \cdot W_0 \tag{59}$$

In either case, p/p_0 is then divided by RØBARA = $\tilde{\rho}_0 / \rho_0$ to eliminate the effects of gas imperfections, if any, upon the value of the reservoir density, ρ_0. This is done because gas imperfections are not taken into account in the nonequilibrium integration, as explained in Section 2.5 of Volume I. In addition, the gas entropy is calculated from equation I(251).

If an electronic nonequilibrium model is in use, CØMM computes the contribution to the electron energy transfer $\dot{q}_e$ from elastic collisions, using the formula

$$(\Delta \dot{q}_e)_{elas} = \sum_{j=2}^{n} \frac{2W_e}{W_j} \cdot \frac{3}{2} R_0(T - T_e) \frac{n_e \nu_{ej}}{N_0} \qquad (60)$$

where for neutral species j, the collision frequency ν_{ej} is given by

$$\nu_{ej} = \frac{4}{3}\sqrt{\frac{8 R_0 T_e}{\pi W_e}}\ n_j\ \bar{Q}_{ej}^{(1,1)} \qquad (61)$$

and for ionic species

$$\nu_{ei} = \frac{8}{3}\sqrt{\frac{\pi}{m_e}}\ \frac{n_i e^4}{(2k\ T_e)^{3/2}}\ \ell n\ \left(\frac{k^3 T_e^{\,2}}{\pi n_e e^6}\right) \qquad (62)$$

These formulas are derived in Appendix A of Volume II; see equations II(25-31). The notation is

W_e, W_j	molecular weights for electron and j^{th} species
$n_e = N_0 \rho \gamma_e$	electron concentration
n_j	concentration of a neutral atom or molecule
n_i	concentration of an ion
$\bar{Q}_{ej}^{(1,1)}$	Maxwell-averaged momentum transfer cross section (assumed the same for all neutrals)
m_e	electron mass
k	Boltzmann constant
N_0	Avogadro number
e	electronic charge (esu)

In NATA, the cross section $\bar{Q}_{ej}^{(1,1)}$ is obtained by calling the function PIØMEG(TEP). The constant factors in (61) and (62) are represented by VIC1, VIC2, and VNC, and are preset in a data statement in CØMM. Equation (60) is evaluated straightforwardly, and the result QELAS = $(\Delta \dot{q}_e)_{elas}$ is added to the $\dot{q}_e$ due to the reactions to obtain the total rate of energy transfer to the electron gas.

3.10 Subroutine CXSECT

Subroutine CXSECT is used by subroutine XSECT to determine the correspondence between species pairs in the master species list for the code and in the rearranged species list stored in the common variable I(J) (see the discussion of subroutine XSECT below). When the subroutine is called with its arguments set equal to the indices of a pair of species in the master species list, it returns the arguments as the indices of the corresponding species pair in the rearranged list. If the original indices are denoted by L,J, the corresponding revised indices are I(L), I(J) if L-J and I(L)-I(J) have the same sign, and are I(J), I(L) otherwise.

3.11 Subroutine DERIVS

During the nonequilibrium solution, the derivatives of the dependent variables, including the species concentrations, are calculated by calling subroutines THERM, CØMM, and EXACT in sequence. If an electronic nonequilibrium gas model is in use, THERM must be called twice, once with CT = T/T_0, and once with CT = T_e/T_0, to provide species properties at both the gas temperature and the electron temperature. The use of these properties has been explained in the analysis of subroutine CØMM. Also, if the boundary layer is being included in the flow solution, an iteration is necessary to determine the self-consistent solution for the boundary layer displacement thickness δ^* and the derivatives of the flow variables (Section 7.6 of Volume I). These calculations are called for at several locations in subroutines NØNEQ and RNKT. For compactness of coding, these operations are all placed in a separate subroutine, DERIVS.

For an electronic nonequilibrium model (TN = 2), DERIVS saves the current value of CT in the location CTSAVE, resets CT to CTE = T_e/T_0, and calls THERM to compute the species properties at the electron temperature. The values $\mu_i^{\circ}(T_e)/R_0T_e$ are stored directly into XMJATD(I,2). The enthalpy, specific heat, and entropy of the electrons are saved temporarily in the locations HJE, CPJE, and SENTE. Then CT is reset to the value T/T_0 and THERM is called again to compute the species properties at the gas temperature. THERM overwrites the previously calculated data for the electrons (species 1), but these are reset from HJE, CPJE, and SENTE.

For a chemical nonequilibrium model (NT = 1), THERM is called only once, to obtain the properties for all the species at the gas temperature, T.

In either case, DERIVS next calls CØMM and EXACT to compute the derivatives of the species concentrations and other dependent variables. It then applies the condition I(407b); if this condition is violated, the indicator FAILED is set to .TRUE. and a RETURN is executed.

The remaining portion of DERIVS controls the iteration to obtain a self-consistent solution for the boundary layer displacement thickness and the derivative $d\ln M/dx$ of the logarithm of the Mach number. If the boundary layer is not being included in the flow solution (ISW3B = 0), or if the solution is still in the upstream (uncoupled) region (IUPD = 1), this iteration is not performed. The technique used in the iteration is explained in Section 7.6 of Volume I, in the vicinity of equation I(424).

3.12 Subroutine DSMSØL

DSMSØL is the subroutine called for simultaneous solution of systems of linear equations in the calculation of thermochemical equilibrium conditions, in the equilibrium flow solution, and in the nonequilibrium solution. The calling sequence for DSMSØL was embodied in the original program version received from Cornell Aeronautical Laboratory (reference 6), but the subroutine itself was not provided by CAL because it was written in machine language and would not be operable on other types of computer system. To provide the required

capability for solution of systems of equations, subroutine SIMQ from the IBM Scientific Subroutine Package (reference 7) was incorporated into NATA, and a Fortran DSMSØL routine was written to rearrange the data for the equations from the form provided by the CAL routines into the form required by SIMQ.

For a system of linear equations

$$\sum_{k=1}^{i} a_{jk}X_k = b_j \quad (j = 1,\ldots, i) \tag{63}$$

in i unknowns (X_1 ..., X_i), NATA stores the coefficients a_{jk} and constants b_j in the double precision array AA in unlabelled common. The dimensions of AA are (22,24). The matrix of coefficients a_{jk} is stored as

$$a_{jk} = AA(j,k) \quad \begin{cases} j = 1,\ldots, i \\ k = 1,\ldots, i \end{cases} \tag{64}$$

The constants b_j are stored as

$$b_j = AA(j, i + 1) \tag{65}$$

Subroutine SIMQ (discussed below) requires that the coefficients a_{jk} be stored columnwise in a singly dimensioned array A, without gaps, and that the constants b_j be stored in an array B. DSMSØL reorganizes the data into this form, using the storage locations of the array AA to contain A and B. The array A is assigned to the first i^2 storage locations in AA, that is, AA(1,1), AA(2,1), ..., A(22,1), AA(1,2), AA(2,2),... The array B is assigned the first i storage locations in the 23rd column of AA, that is, AA(1,23), ..., AA(i,23). If the number of equations i is equal to 22, the reorganization leaves the matrix AA unaltered. If i is less than 22, the coeffi ients a_{jk} are shifted down so as to fill up the first columns of AA without gaps, and the constants b_j are shifted

up into the previously unused 23rd column. After SIMQ has been called, the solution values x_k provided by SIMQ are shifted from the 23rd column into the (i + 1)th column, for use by the calling routine.

The coding of DSMSØL is straightforward, and can be followed easily with the help of the definitions given in the glossary of Fortran symbols (Section 4.80).

3.13 Subroutine DUMP

DUMP is part of the NATA system for dealing with possible errors in code operation. NATA contains numerous validity checks. If one of these is violated, the routine in which the error was detected calls DUMP. This small subroutine sets the indicator ERR to .TRUE. and writes a message giving the name RNAME of the calling routine. It then returns control to the calling routine. The calling routine then executes a RETURN. At each higher-level routine, statements of the form IF(ERR) RETURN pass control up is still higher levels until the main program is reached. In MAIN, when a .TRUE. value of ERR is detected, subroutine DUMPEX is called to print diagnostic data (see below). After the return from DUMPEX, MAIN proceeds as if a normal case completion had occurred; i.e., the job is not terminated unless the input ISW4A is zero.

Originally, the operations now performed by DUMP and DUMPEX were included in a single subroutine (called DUMP). When NATA become so large that it had to be run using overlay on the UNIVAC 1108, the original DUMP routine was divided into two parts, and the larger of these (DUMPEX) was placed into a separate overlay segment so as to minimize the core storage requirement.

3.14 Subroutine DUMPEX

DUMPEX produces the diagnostic dumps when an error has been detected in the operation of the code. The routine contains unlabelled common and many of the labelled common blocks used in NATA. When DUMPEX is executed, most of the variables and arrays listed in these common statements are printed out

in namelist format. In addition, if a binary tape of data for plotting is being written (DATAPE = .TRUE.), DUMPEX backspaces this tape to eliminate all of the records written during the current case.

3.15 Subroutine ELCØND

Subroutine ELCØND computes the electrical conductivity SIGMA of the gas in mhos/cm from equation I(98) using the values of SSIG = $(e/k)^2/\sqrt{T}$ and Q(1,I,J) = $k^{-1}\sqrt{T}\ \Delta^{(1)}_{ij}$ computed in subroutine TRANSP. In this computation it is assumed that if electrons are present in the gas, they are the first species. If, on the other hand, electrons are not present (IELEC = 0), the electrical conductivity is set to 0.0 and the computation is bypassed.

3.16 Subroutine ELTIME

The function of subroutine ELTIME is to determine and print out the elapsed execution time since the beginning of the run. When NATA is run on the UNIVAC 1108, ELTIME uses the utility routines RESET and TIME, which are provided by the EXEC II operating system. When the code is used on the IBM 360/75 at Avco, the functions of RESET and TIME are simulated by a small Fortran program (called RESET, Sec. 3.58) which utilizes the system subroutine ACUCPU.

ELTIME contains a logical indicator CALLED, which is preset to .FALSE. in a data statement. When ELTIME is called for the first time in a job, a test on CALLED leads to a call to RESET, which initializes the time counter I for subroutine TIME to 0, and ETØ and ET are both set to zero. Also, CALLED is set to .TRUE., so that this part of the routine is bypassed when ELTIME is called on subsequent occasions in the run.

In such subsequent calls, ELTIME sets ETP = ETØ, and then sets ETØ to ET, where ET is the value left in storage by the previous call (unless this argument has been altered by the calling routine). Then ELTIME calls subroutine TIME(I), which returns the elapsed time I since the call to RESET, measured in milliseconds. ELTIME converts this time to seconds

and stores it in ET. Then, unless the argument IP is zero, ELTIME converts ET into minutes and stores the result in ETM, and also computes the elapsed time DET in seconds since the last previous printed output of the time. A message is then printed giving ETM and DET. If the printed message is skipped because IP = 0, ETØ is reset to ETP to maintain its significance as the elapsed time since the printing of the last time message.

3.17 Subroutine EPART

This subroutine computes the parameters governing the partition of reaction energy between radiative loss and energy transfer to the electron gas for each reaction in a gas model which includes electronic nonequilibrium. The arguments in the calling sequence are defined in Section 4.85. The subroutine treats six cases, or types of reactions, which are distinguished by an index IT. In the following discussion of these cases, ϵ_f = EF and $-\epsilon_r$ = -ER denote the energies gained by the electron gas in one mole of reactions in the forward and reverse directions, respectively; and q_f = QF and $-q_r$ = -QR denote the corresponding energies lost by radiation.

The six cases are

IT = 1 $\quad \epsilon_f = -a\, R_0 T_e$

$$q_f = \epsilon_0 - \epsilon_f$$

$$\epsilon_r = q_r = 0$$

IT = 2 $\quad \epsilon_f = -\frac{3}{2} R_0 T_e$

$$q_f = \epsilon_0 - \epsilon_f$$

$$\epsilon_r = q_r = 0$$

IT = 3 $\quad \epsilon_f = \epsilon_r = q_f = q_r = 0$

IT = 4 $\quad \epsilon_f = \epsilon_r = -\frac{3}{2} R_0 T_e$

$q_f = q_r = 0$

IT = 5 $\quad \epsilon_f = \epsilon_r = \epsilon_0$

$q_f = q_r = 0$

IT = 6 $\quad q_f = \epsilon_0$

$\epsilon_f = \epsilon_r = q_r = 0$

In these formulas, ϵ_0 = E0 denotes the energy available for partitioning between the electrons and the radiative losses, R_0 the universal gas constant (1.9872 cal/mole-deg), and T_e the electron temperature ($^{\circ}$K).

3.18 Subroutine EQCALC

EQCALC computes the conditions of thermochemical equilibrium at specified temperature T ($^{\circ}$K) and pressure P (atm), using the method explained in Section 6.1 of Volume I. As an aid to following the coding, the correspondence of Fortran and mathematical symbols is reviewed in the following list:

BET(I)	$\nu^*_{i-c} - 1$	(I = i-c)
CDIJ(I,J)	$\bar{\nu}_{i-c,j}$	(I = i-c)
CGI(I)	W_i	
QM(I)	q_i	

XMJAT(I)	μ_i°/R_0T
ZCAP(I)	X_i
ZGMU(I)	ℓnX_i
ZPZ	ℓnp

EQCALC first saves the current value of the nondimensional temperature CT in the location CTSAVE. This is done because EQCALC may be called with temperatures T differing from the temperature at the current flow point (for example, during the equilibrium normal shock solution). The nondimensional temperature CT is then reset to T/T_0 to allow calculation of the species thermal properties by calling subroutine THERM.

Next, EQCALC computes initial estimates of the species mole fractions and related quantities. The mole fractions for the independent species (I = 1 to ISC) are set to q_i = QM(I), defined by equation I(10). If any of the QM(I) are zero or negative (due to roundoff errors), the corresponding ZCAP(I) values are reset to 10^{-3}. Subroutine THERM is then called to compute the thermal properties of all the species at temperature T. The mole fractions for the dependent species (I = ISC + 1 to ISS) are then estimated using equation I(224), which can be written in the logarithmic form

$$\ell n\, X_i = \text{CHII(i-c)} + \sum_{j=1}^{c} \bar{\nu}_{i-c,j}\, \ell n\, X_j \qquad (66)$$

where

$$\text{CHIJ(i-c)} = -\frac{\mu_i^{\circ}}{R_0T} + (\nu^*_{i-c} - 1)\ell n\, p + \sum_{j=1}^{c} \bar{\nu}_{i-c,j}\, \frac{\mu_j^{\circ}}{R_0T} \qquad (67)$$

In this calculation, SKIL(i) is used as an intermediate variable, first to represent $\ell n\, X_{i-c}$, then X_{i-c}. If the number of dependent species, ISMC = n-c, is zero, then the calculation of

the dependent species mole fractions is skipped here and elsewhere in EQCALC.

Once initial estimates of the X_i and $\ln X_i$ have thus been determined, EQCALC enters the Newton-Raphson iteration based on equations I(277) to I(233). The iterative loop runs from statement 70 down to (but not including) statement 220. The constant terms F_j, equation I(227), in the system of linear equations I(233a) for the corrections h_n^r are loaded into the $M1^{th}$ column of the matrix AA by the DØ loop ending at statement 90. Note that, in statement 80, X_i for i = c + 1 to n is denoted by SKIL(i). Then the matrix of coefficients $-X_n^r(\partial F_j/\partial X_n)^r$, equation I(233b), is loaded into the first ISC rows and columns of AA by the nest of DØ loops ending at statement 140. Subroutine DSMSØL is then called to solve the system of equations for the correction factors h_n^r. Upon the return from DSMSØL, the solution values for these quantities are obtained from the $M1^{th}$ column of the matrix AA, where M1 = ISC + 1.

The corrected values of the species mole fractions are then calculated. The mole fractions for the independent species (i = 1 to c) are obtained using equation I(234):

$$X_k^{r+1} = (1 + h_k^r) \cdot X_k^r \qquad (68)$$

The factor $1 + h_k^r$ is denoted by ZA. If ZA is zero or negative, the corrected mole fraction is estimated by halving the previous value:

$$X_k^{r+1} = X_k^r/2 \qquad (69)$$

These calculations are done in the DØ loop ending at statement 180.

The corrected mole fractions for the dependent species are calculated from the equilibrium relations I(224), based on the corrected X_i for the independent species. Equations (66) and (67) are again used, in the DØ loop ending at statement 200.

The convergence test for the Newton-Raphson iteration is that the absolute values of all the h_k^r be less than or equal to a criterion value TEST, which is preset (in subroutine INIT) to 10^{-6}. If convergence has not been achieved after NTEST = 100 iterations, the DUMP routine is called to terminate the case and produce diagnostic output.

When convergence has been achieved, EQCALC computes a number of gas properties based on the temperature T, the pressure P, and the equilibrium mole fractions determined by the Newton-Raphson solution of equations I(227). The nondimensional molar entropy of the gas mixture, ZSEN = S/R_0, is calculated using equation I(30b). Note that the specific entropy of the gas is equal to

$$s = \left(\frac{S}{R_0}\right) \cdot \frac{R_0}{W} = \frac{ZSEN \cdot CRA}{ZCM} \qquad (70)$$

in cal/gm ^{o}K. The nondimensional molar enthalpy ZCH of the gas is calculated from equation I(238) and the molecular weight ZCM from equation I(236). Finally, the corrections for gas imperfections, equations I(79) and I(81), are applied.

3.19 Subroutine EXACT

This subroutine computes the derivatives of the species concentrations and the other dependent flow variables for use in the nonequilibrium solution. EXACT consists of two sections. The first section (down to statement 100) calculates the derivatives required in the perturbation solution; the second gives those used in the nonequilibrium integration.

3.19.1 Derivatives for the Perturbation Solution

The derivatives required in the perturbation solution (Section 7.3 of Vol. I) are those of the equilibrium flow variables. These derivatives are computed in EXACT by solving a system of n+2 linear equations of the form

$$\sum_{j=1}^{n+2} A_{ij} X_j = B_i \qquad (71)$$

The unknowns X_j are

$$X_j = \frac{d \bar{\nu}_j}{dx} \qquad \text{for } j = 1 \text{ to } n \qquad (72a)$$

$$X_{n+1} = \frac{1}{T_0} \frac{d\bar{T}}{dx} \tag{72b}$$

$$X_{n+2} = \frac{d \ln \bar{\rho}}{dx} \tag{72c}$$

The system of equations is

$$\text{i=1 to c:} \quad \sum_{j=1}^{n} \alpha_{ji} \frac{d \bar{\gamma}_j}{dx} = 0 \tag{73a}$$

$$\text{i=c+1 to n:} \quad \sum_{j=1}^{c} \frac{\bar{\nu}_{i-c,j}}{\bar{\gamma}_j} \frac{d \bar{\gamma}_j}{dx} - \frac{1}{\bar{\gamma}_i} \frac{d \bar{\gamma}_i}{dx}$$

$$+ \frac{T_0}{\bar{T}} \left[\frac{\bar{H}_i / R_0 T_0}{\bar{T}/T_0} - 1 - \sum_{j=1}^{c} \bar{\nu}_{i-c,j} \left(\frac{\bar{H}_j / R_0 T_0}{\bar{T}/T_0} - 1 \right) \right] \frac{d(\bar{T}/T_0)}{dx}$$

$$+ (\nu^*_{i-c} - 1) \cdot \frac{d \ln \bar{\rho}}{dx} = 0 \tag{73b}$$

$$\text{i=n+1:} \quad \sum_{j=1}^{n} \frac{d \bar{\gamma}_j}{dx} + \frac{T_0}{\bar{T}} \left[\sum_{j=1}^{n} \bar{\gamma}_j \right] \frac{d(\bar{T}/T_0)}{dx}$$

$$+ \left[\frac{1}{\bar{W}} - \frac{\bar{u}^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{T_0}{\bar{T}} \right] \frac{d \ln \bar{\rho}}{dx}$$

$$= \frac{\bar{u}^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{T_0}{\bar{T}} \cdot \frac{d \ln A}{dx} \tag{73c}$$

$$\text{i=n+2:} \quad \sum_{j=1}^{n} \frac{\bar{H}_j}{R_0 T_0} \frac{d \bar{\gamma}_j}{dx} + \left[\sum_{j=1}^{n} \bar{\gamma}_j \frac{\bar{C}_{pj}}{R_0} \right] \frac{d(\bar{T}/T_0)}{dx}$$

$$- \frac{\bar{u}^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \frac{d \ln \bar{\rho}}{dx} = \frac{\bar{u}^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{d \ln A}{dx} \tag{73d}$$

Equation (73a) is I(363) with the meanings of the indices interchanged; equation (73b) is I(367) with the signs of all terms reversed and the index replacements $j \rightarrow i$, $\ell \rightarrow j$; equation (73c) is I(299) applied to the equilibrium flow; and (73d) is I(293) applied to the equilibrium flow. Note that

$$\frac{\bar{H}_i}{R_0 T_0} = \text{SHJ(i)} \tag{74a}$$

$$\frac{\bar{T}}{T_0} = \text{CT} \tag{74b}$$

$$\left(\frac{\bar{u}}{u_s}\right)^2 = \text{SU2} \tag{74c}$$

$$\frac{u_s^2}{R_0 T_0} = \frac{R_0 T_0}{W_0} \cdot \frac{1}{R_0 T_0} = \frac{1}{W_0} \tag{74d}$$

Equation I(296) is used in the coding of the quantity 1/W in (73c).

The coefficients A_{ij} for these equations (73) are loaded into the array AAA(I,J) for I = 1 to n + 2 and J = 1 to n + 2. The constants B_i are loaded into AAA(I, n + 3) for I = 1 to n + 2. The system of equations is then solved by calling subroutine DØBØL, and the resulting values for the derivatives (72) are retrieved from the (n + 3)th column of AAA.

3.19.2 Derivatives for the Numerical Integration

During the numerical integration of the rate equations, the derivatives of the species concentrations are given explicitly by equation I(287). These derivatives are computed in the nest of DØ loops ending at statement 120; they are denoted by DGJ(J).

The nature of the remaining derivatives to be determined depends upon the region of solution (upstream or downstream) and the type of gas model being used. The unknowns solved for in the various cases are summarized in Table I. When an electronic nonequilibrium model is being used (NT=2), there is a fourth dependent variable in the integration, the total enthalpy h_0. However, its derivative is given explicitly by equation I(333), so that it is not involved in the simultaneous solution for the derivatives listed in Table I.

For a given type of gas model, the same set of equations is solved regardless of whether the solution is in the upstream region (inverse method) or the downstream region (direct integration). However, the matrix of coefficients and the vector of constants differ in the two cases because of the difference between the sets of unknowns being solved for. Figure 8 is a flowchart of the solution for the derivatives. By following this flowchart, one can easily write out the equations being solved in each of the four cases listed in Table I, and verify that they are identical with the text equations listed in the table. It should be noted that NE = NT + 1 is the number of unknowns and equations in each case. For NT = 2, the variable SCPGH contains the sum $\sum_{j=2}^{n} C_{pj} \gamma_j$, but for NT = 1 the sum is over all species (see subroutine CØMM).

3.20 Function EXP

The operating system on the IBM 360/75 at Avco produces several lines of diagnostic output whenever the Fortran function DEXP(X) underflows to give a value 0. To avoid such interruptions of the normal output, the IBM 360 version of NATA contains a Fortran-coded exponential function, EXP(X), which checks the argument X and sets the result to zero for $X \leq -180$. For $X > -180$, EXP calls DEXP to compute EXP. This routine is not used in the UNIVAC version of NATA because the UNIVAC system EXEC II does not produce a diagnostic message for exponential underflow.

TABLE I

DERIVATIVES DETERMINED BY EXACT DURING THE NUMERICAL INTEGRATION

Region	IUPD	Model Type	NT	Unknown Derivatives	Equations Solved
Upstream	1	Chemical	1	$\frac{d(T/T_0)}{dx}$, $\frac{d\ell n\,\widetilde{A}}{dx}$	I(293), I(299)
Upstream	1	Electronic	2	$\frac{d(T/T_0)}{dx}$, $\frac{d\ell n\,\widetilde{A}}{dx}$, $\frac{d(T_e/T_0)}{dx}$	I(328), I(331), I(332)
Downstream	0	Chemical	1	$\frac{d(T/T_0)}{dx}$, $\frac{d\ell n\rho}{dx}$	I(293), I(299)
Downstream	0	Electronic	2	$\frac{d(T/T_0)}{dx}$, $\frac{d\ell n\rho}{dx}$, $\frac{d(T_e/T_0)}{dx}$	I(328), I(331), I(332)

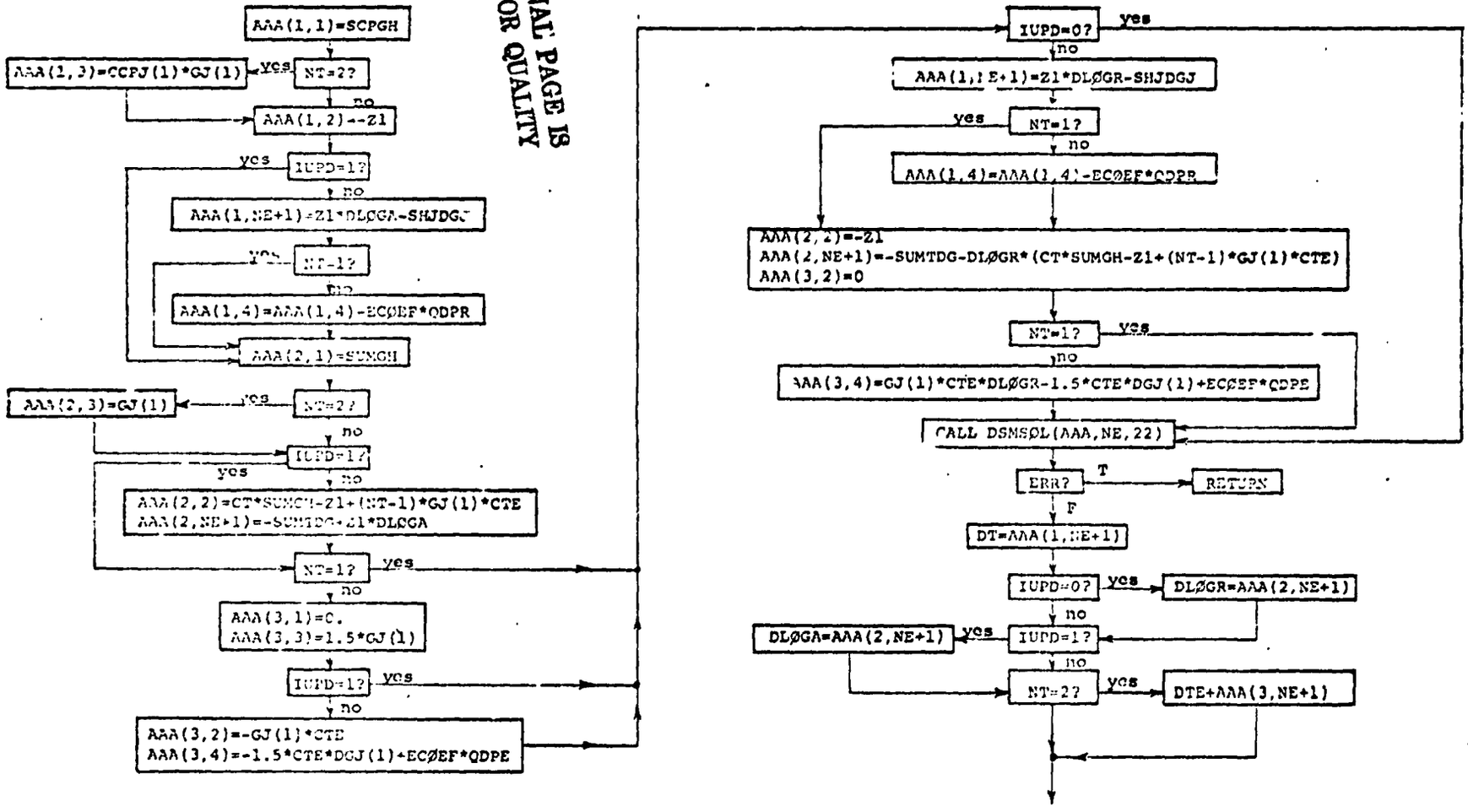


Figure 8. Flowchart of the Solution for the Derivatives of the Flow Variables in the Numerical Integration

3.21 Subroutine FINDX

The purpose of this subroutine is to determine the value of x at which the geometric area ratio has a given value A. An indicator (UPDØWN) specifies whether the upstream or the downstream solution is desired. An entry point FINDXC(A,MBL,X) permits determining the value of x at which a rectangular channel has a half-width A, in centimeters, on its MBLth profile.

The solution is obtained by a Newton-Raphson iterative procedure. The independent variable in the iteration is taken to be $V = |x|$.

If the area ratio is given (normal entry point), then its value is tested before the solution is attempted. If A is less than 0.99999, the case is terminated by calling DUMP, because by its definition the area ratio cannot be less than unity. If A is between 0.99999 and 1.0, the discrepancy is assumed to be due to a numerical (e.g., round-off) error elsewhere in the program; in this case, X is set to 0.0 (the value for A = 1.), and the RETURN is executed.

If A is greater than 1.0, V is initialized to 1.0 and the Newton-Raphson procedure is started. The geometric area ratio AR and its derivative DADX at the point X = V·sign (UPDØWN) are calculated by calling GEØMAR. If AR agrees with A to within one part in 10^5, the iteration has converged. If not, a new estimate of V is calculated from the Newton-Raphson formula

$$V = V_o + \frac{A-AR}{|DADX|} \qquad (75)$$

However, V is not allowed to increase by more than a factor of 2 in any step; this is to prevent V from becoming very large after a step in which V was very close to the throat (where DADX = 0.). A maximum of 50 iterative steps is allowed.

If the channel width is given (entry FINDXC), then V is initialized to 100 cm, a value typical of axial positions where test panels might be installed in a channel. For each trial value of V, the profile half width AR = Z(MBL) and slope DADX = DZDX(MBL) are computed by calling the entry point GMAR3 of GEØMAR. The same Newton-Raphson formulas and convergence test are used as in the area ratio case.

3.22 Subroutine FRØZEQ

FRØZEQ is the routine which controls the computation of the frozen and equilibrium flow solutions. The frozen flow solution is generated when the routine is called through entry FRØZEN. Entry EQUIL(IPASS) gives the equilibrium solution.

3.22. Nondimensionalization of the Flow Variables

Throughout subroutine FRØZEQ, the calculations are carried out with most of the flow variables in nondimensional form. The relation between the Fortran symbols and the dimensional flow variables is summarized in the following formulas:

AFNX	A_e, effective area ratio
CH	$\frac{W_0 h}{R_0 T_0}$, nondimensional specific enthalpy
CHA	$\frac{W_0 h_0}{R_0 T_0}$, nondimensional enthalpy in the reservoir
CT	T/T_0, nondimensional temperature
CX	x, axial nozzle coordinate (cm)
FLUX	$\frac{\rho u}{\rho_0 u_s}$, nondimensional mass flux

PRES $\frac{p}{p_0}$, nondimensional pressure

RHØ $\frac{\rho}{\rho_0}$, nondimensional density

SU $\frac{u}{u_s}$, nondimensional velocity

In these definitions, the subscript 0 denotes conditions in the reservoir, R_0 is the universal gas constant, and u_s is a velocity defined by

$$u_s = \sqrt{R_0 T_0 / W_0} \tag{76}$$

where W_0 is the molecular weight in the reservoir and R_0 is the universal gas constant in mechanical cgs units (erg/mole ^oK).

3.22.2 Determination of Frozen Sonic Conditions

When entry FRØZEN is called, subroutine FRØZEQ determines the sonic conditions in the frozen flow before starting the main frozen flow solution.* This calculation is done by the statements from entry FRØZEN down to entry EQUIL. A flowchart of this part of the subroutine is shown in figure 9. The criterion for locating the sonic point is that the mass flux m be a maximum. This condition follows from the continuity equation I(243) and the fact that the flow area A is a minimum at the sonic throat. The temperature at the sonic point is determined as follows: the temperature is initialized to its reservoir value, and is then decremented repeatedly by the amount DELT1·T_0, where DELT1 is normally 0.01. At each of these temperatures, the corresponding mass flux in frozen flow is calculated by calling subroutines THERM and PRØP. This procedure is continued until the mass flux passes its maximum. Then the temperature step is reduced and an iteration is carried out to locate the sonic-point temperature with high accuracy.

*The calculation of sonic conditions for the equilibrium solution is done in a separate subroutine, NRMAX.

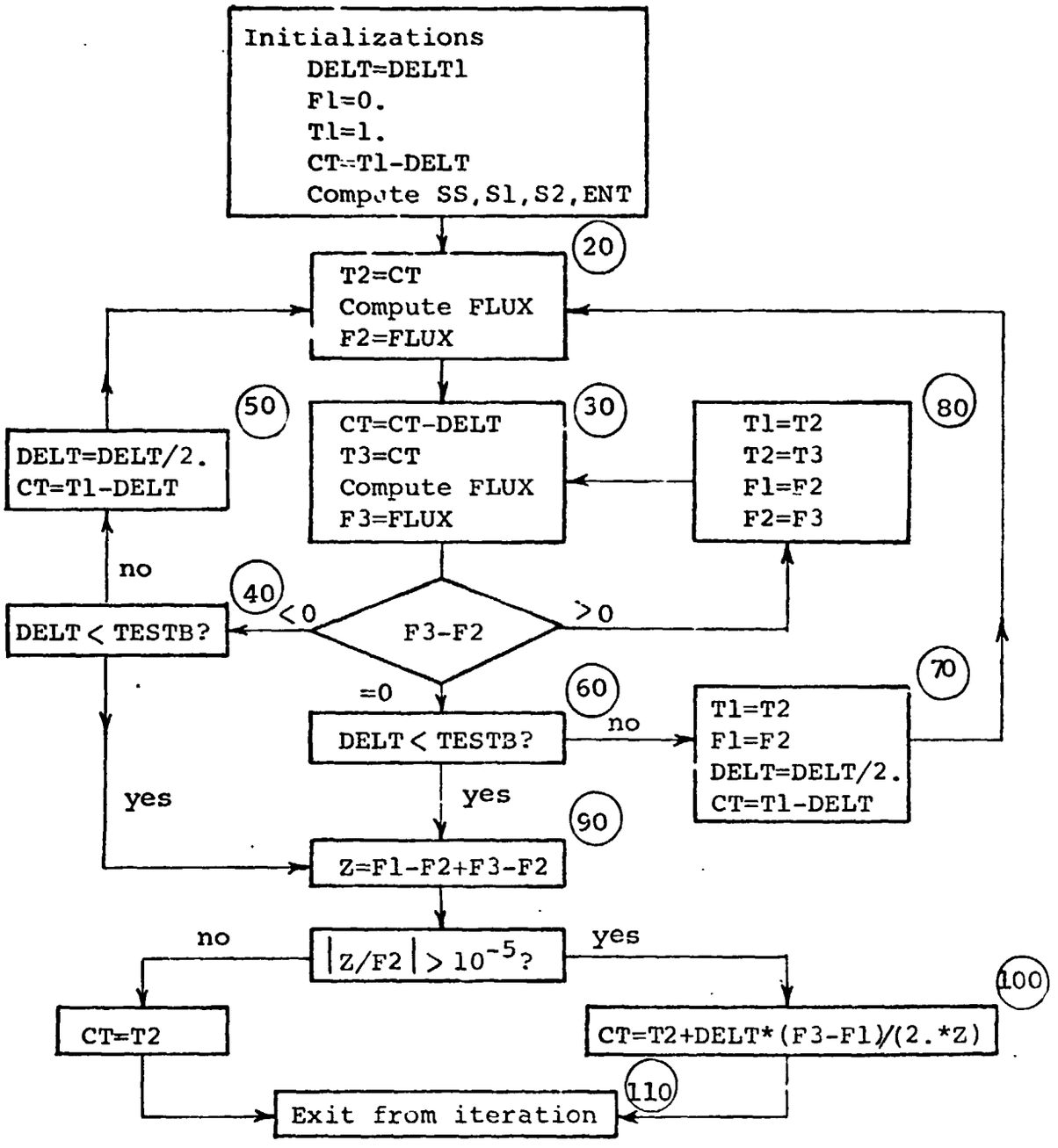
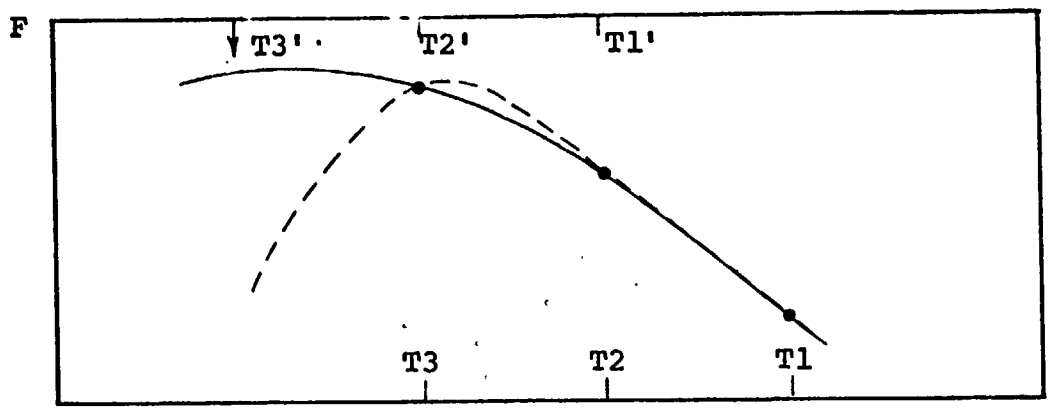


Figure 9. Flow Chart for Determination of Sonic Conditions

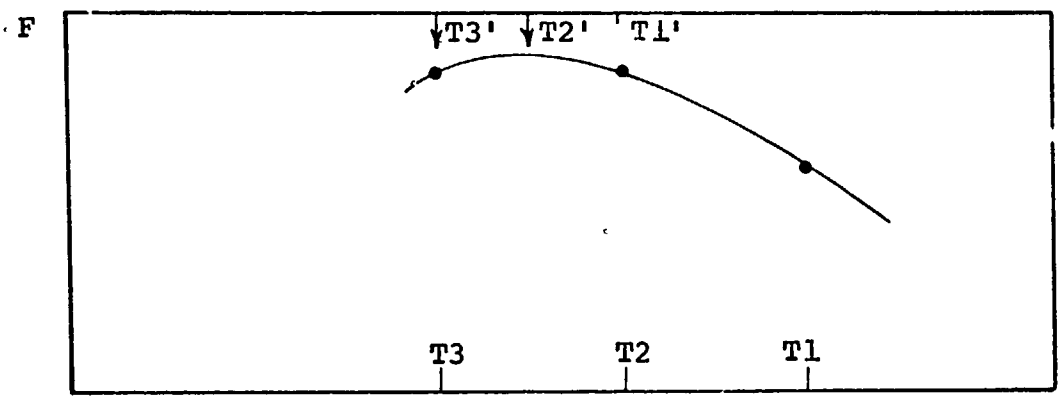
The algorithm for determining the sonic-point temperature (figure 9) is based upon consideration of a sequence of three equally-spaced temperatures T1 > T2 > T3 and the corresponding mass flux values F1, F2, F3. T1 and T2 are always chosen in such a way that F2 > F1; thus, T1 is definitely known to be higher than the sonic temperature. At each stage of the iteration, the rules for selecting the next triplet of temperatures T1', T2', T3' depend upon the value of F3-F2. There are three cases, which are illustrated schematically in figure 10. In case A (F3 > F2), the three mass flux values form a monotonic sequence F3 > F2 > F1. In this case, the available data provide no indication that the point of maximum mass flux has been reached, although this may actually be the case, as shown by the dashed curve in the top section of the figure. In case A, T1' is set equal to T2, T2' to T3, and T3' is obtained by applying the same decrement T1' - T2' to T2'. In case B (F3 = F2), the sonic point is known to lie between T2 and T3. In this case, T1' is set equal to T2, the temperature decrement DELT = ΔT is cut in half, and the remaining two temperatures are set to T2' = T1' - ΔT and T3' = T2' - ΔT = T3. In case C (F3 < F2), the sonic point could lie between T1 and T2 or between T2 and T3. To ensure that T1' lies above the sonic temperature, it is set equal to T1. The decrement ΔT is halved as in case B, and T2' = T1' - ΔT, T3' = T2' - ΔT = T2. If the sonic point actually lies between T2 and T3, as illustrated by the solid curve in the bottom part of figure 10, the situation in the next step of the iteration will be as in case A.

At each step, the magnitude of DELT is compared with the criterion value TESTB = 10^{-5}. If DELT < TESTB, the iteration has converged. A final improvement in the value of the sonic temperature is then obtained by fitting a parabola to the mass flux versus temperature data for the final triplet of points, and determining the maximum analytically based on this quadratic relation. The form assumed is

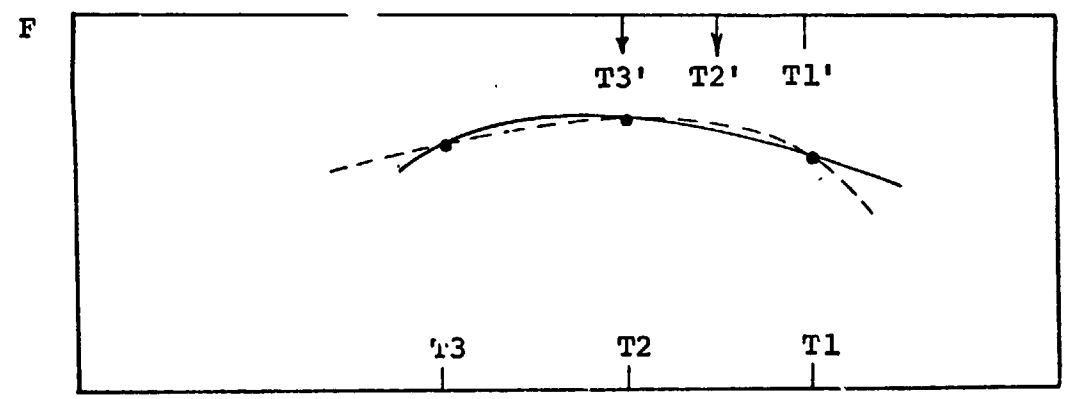
$$F(T) = A + B(T-T_2) + C(T-T_2)^2 \qquad (77)$$



Case A: F3 > F2



Case B: F3 = F2



Case C: F3 < F2

Figure 10. Three Cases in the Algorithm for Finding the Throat Temperature

Substitution of the data points (T_1,F_1), (T_2,F_2), and (T_3,F_3) and solution for the coefficients A, B, C give

$$F = F_2 - \frac{F_3 - F_1}{2(\Delta T)}(T - T_2) + \frac{F_1 + F_3 - 2F_2}{2(\Delta T)^2}(T - T_2)^2 \quad (78)$$

Setting dF/dT = 0 then leads to the final result

$$T_* = T_2 + \frac{(F_3 - F_1)(\Delta T)}{2(F_1 + F_3 - 2F_2)} \quad (79)$$

If the quantity $Z/F_2 = (F_1 + F_3 - 2F_2)/F_2$ is less than or equal to 10^{-5}, equation (77) is not used. In this case, T_* is simply set equal to T_2, to avoid possible errors due to loss of accuracy on the subtraction in Z (or a divide check in (79) if Z = 0).

3.22.3 Main Flow Solution

The basic operation performed by subroutine FRØZEQ is the calculation of the frozen and equilibrium flow solutions. In its essentials, this calculation is quite simple. The nondimensional temperature CT is decremented repeatedly, starting from its reservoir value (CT = 1.). At each value of the temperature, the other flow variables are computed, using subroutines THERM and PRØP in the case of frozen flow and subroutine NEWRAP in the case of equilibrium flow. Then the position x in the nozzle at which the flow point occurs is determined from the known nozzle geometry and the value of the area ratio at the flow point, and the results are printed by calling entry ØUT2 of subroutine ØUT1.

The actual complexity of FRØZEQ, as illustrated by its flowchart (figure 11) is due to provisions for treating various options and regions of the solution. The control variables pertaining to some of these cases and conditions will be defined before the subroutine is discussed in detail:

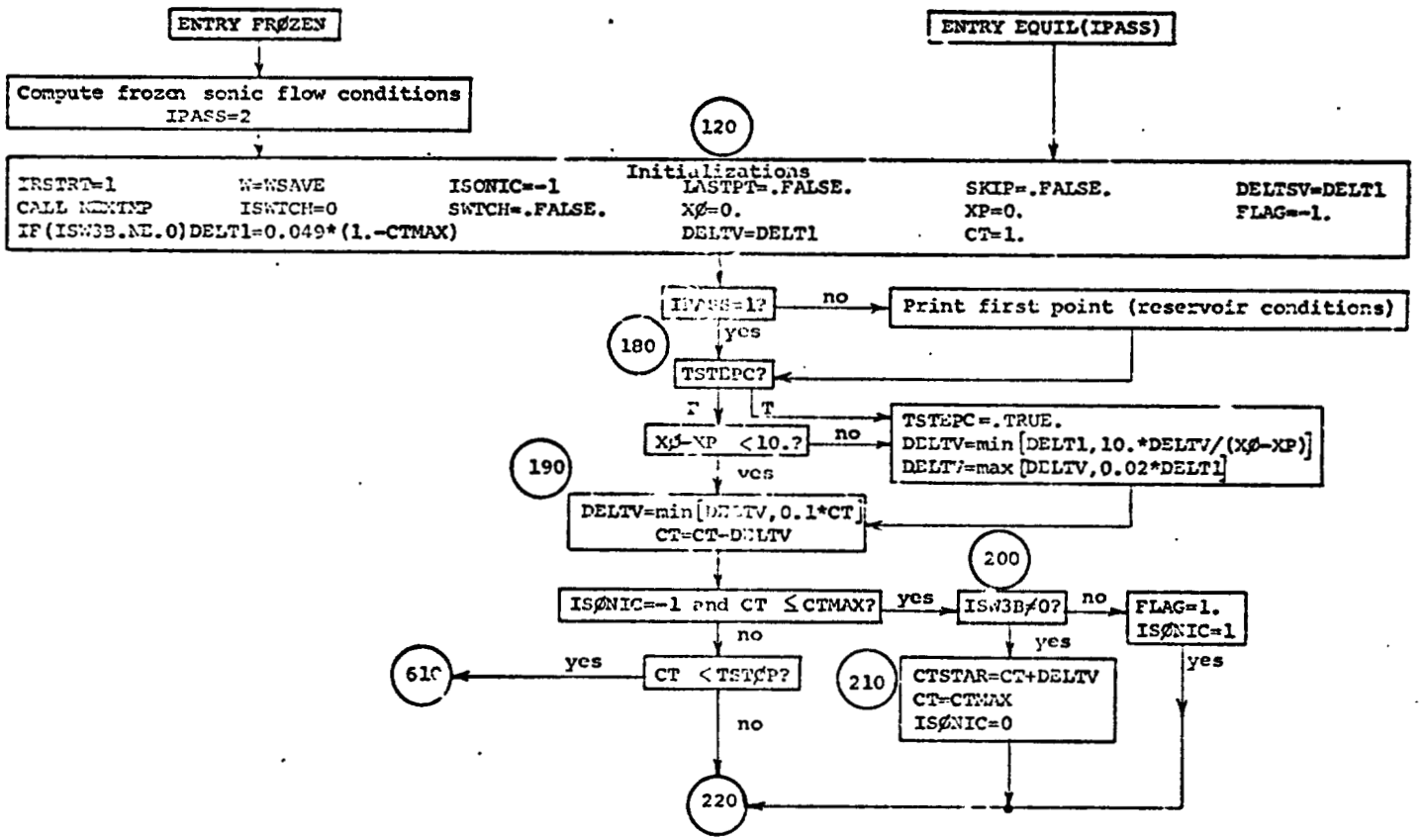


Figure 11a. Flowchart of Subroutine FRØZEQ (Part a)

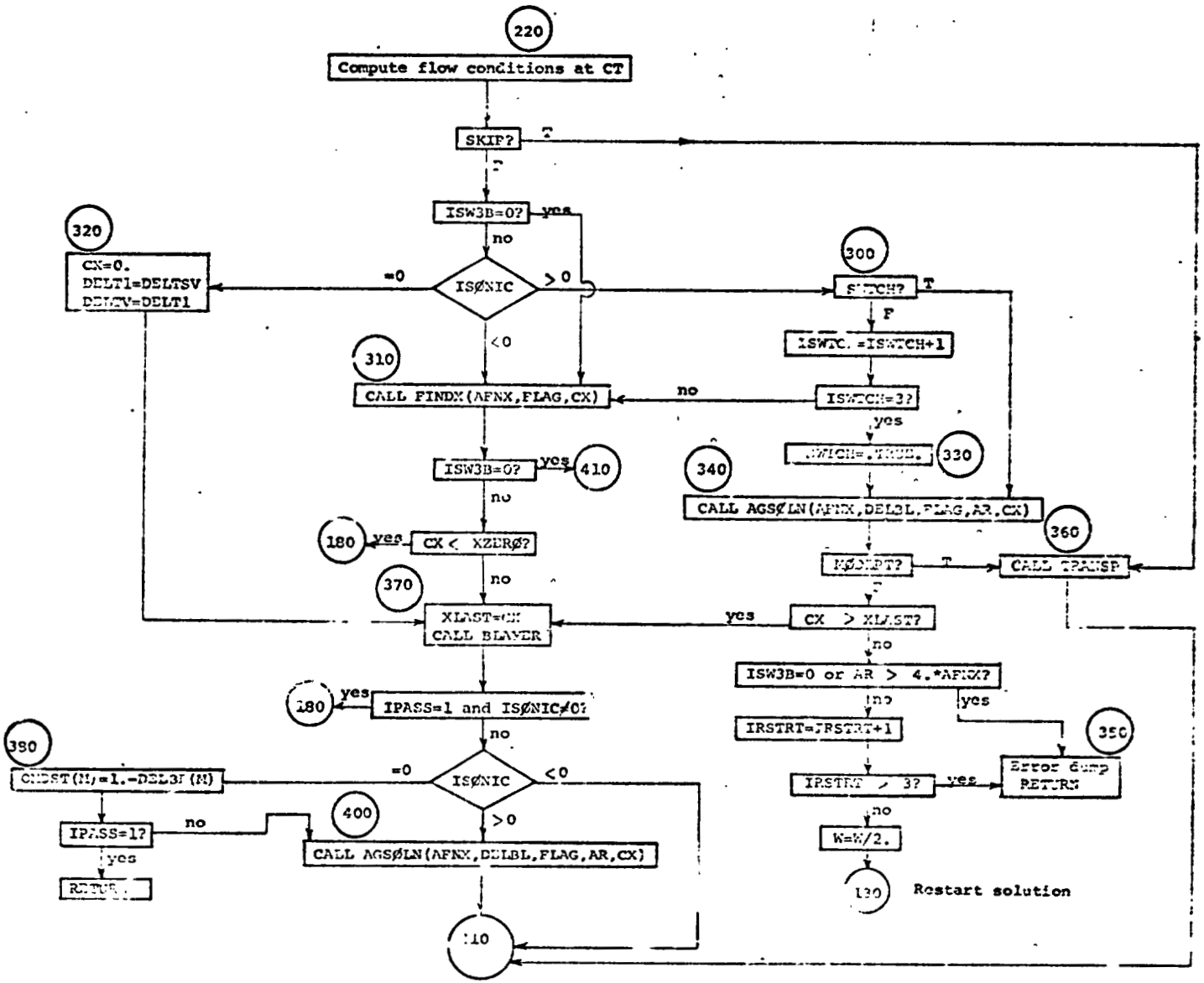


Figure 11b. Flowchart of Subroutine FRØZEQ (Part b)

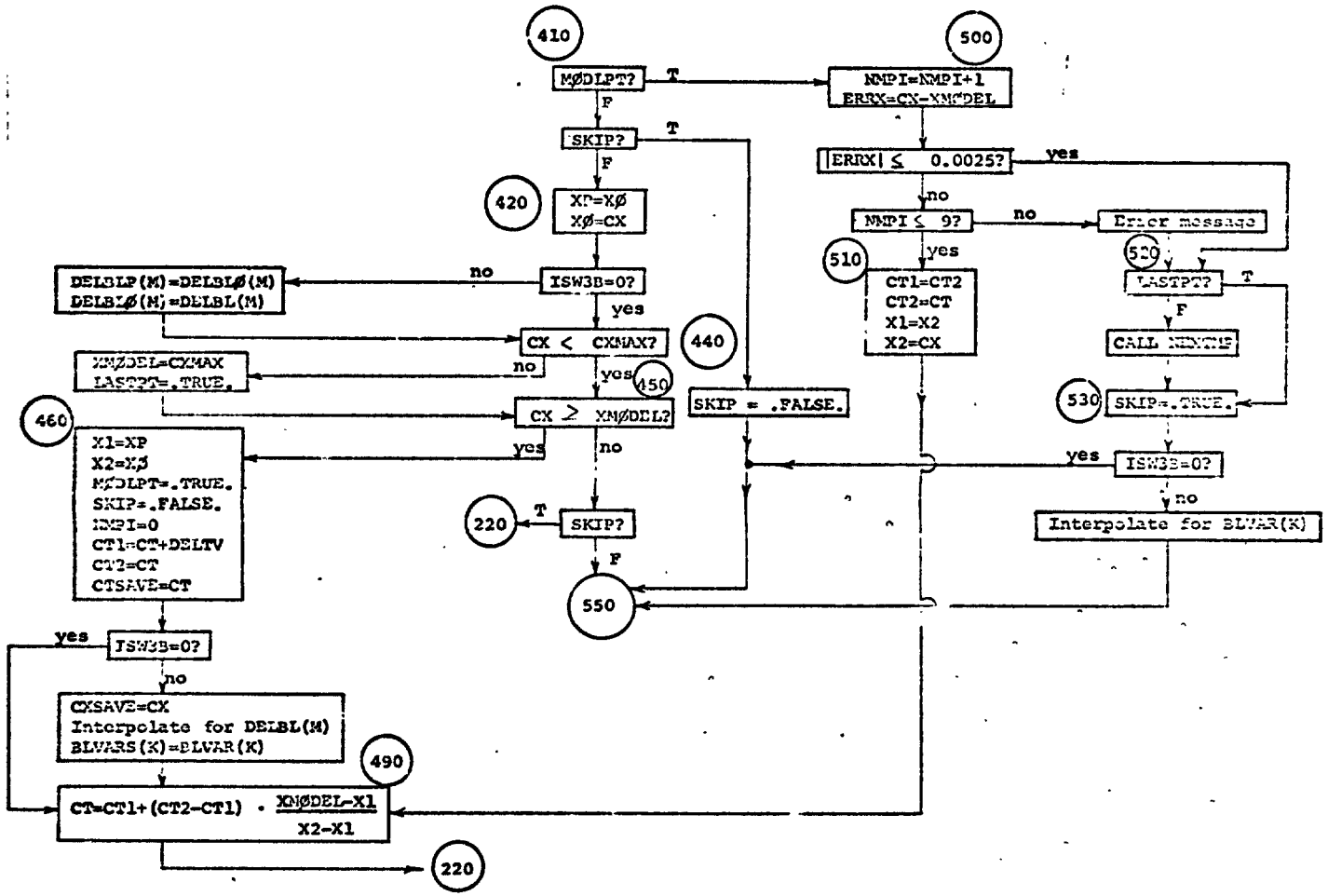


Figure 11c. Flowchart of Subroutine FRØZEQ (Part c)

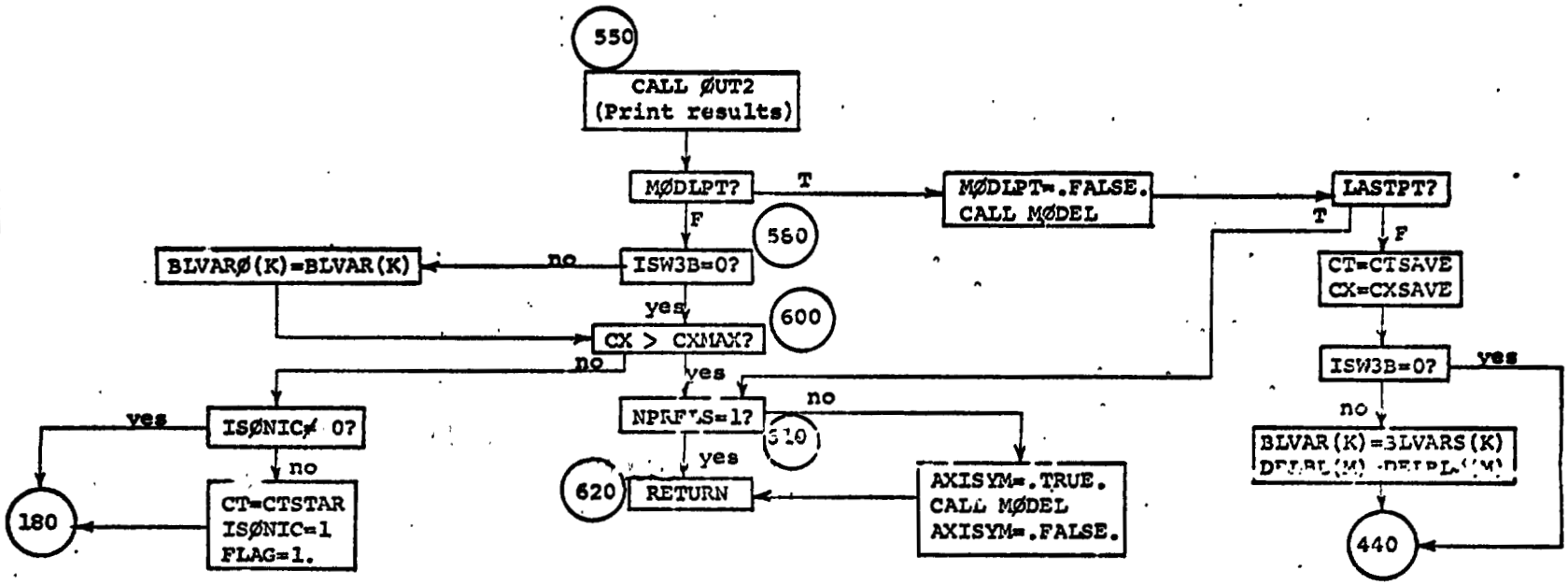


Figure 11d. Flowchart of Subroutine FRØZEQ (Part d)

ISW3B = 0	Boundary layer is omitted.
ISW3B ≠ 0	Boundary layer is included.
IPASS = 1	Preliminary equilibrium solution from the reservoir to the sonic point, to determine the boundary layer displacement thickness at the throat for use in reservoir condition calculations.
IPASS = 2	Final flow solution.
ISØNIC = -1	Region upstream of the throat.
ISØNIC = 0	Sonic point.
ISØNIC = 1	Region downstream of the throat.
MØDLPT = .FALSE.	Regular point of the flow solution.
MØDLPT = .TRUE.	Iteration to find the conditions at a specified model point is in progress.
SKIP = .FALSE.	Regular flow point or model point.
SKIP = .TRUE.	Regular flow point just after a model point.
LASTPT = .FALSE.	Regular flow point or model point.
LASTPT = .TRUE.	Last point of solution.

The simplest case is that of a solution without the boundary layer and without any model points. In this case, FRØZEQ generates a point of the solution by traversing the loop outlined by the statement numbers 180, 190, 220, 310, 410, 420, 440, 450, 550, 580, 600. The temperature decrement DELTV is equal to the input value of DELT1 (normally 0.01) in this case, until the difference XØ - XP in spatial position between two successive flow points exceeds 10 cm. Once this has happened, the unnumbered statements between 180 and 190 reduce DELTV so as to maintain a spatial step size of approximately 10 cm. This variation of DELTV occurs far downstream of the throat, in the region where the free stream temperature has fallen to a small fraction of the reservoir

temperature. DELTV is not allowed to fall below 0.02 DELT1. These adjustments of DELTV in the region far downstream are also made when the boundary layer is included in the solution.

The second statement below 190 tests whether the temperature CT is below the sonic temperature CTMAX, i.e., whether the solution has passed the throat. If it has, in the case without the boundary layer, ISØNIC is reset from -1 to +1, and the indicator FLAG for upstream and downstream solutions in the geometry routines is reset from -1. to +1. Once the flow conditions for the temperature CT have been computed, the spatial position in the nozzle is determined by calling subroutine FINDX(AFNX, FLAG,CX) (statement 310). This subroutine determines the upstream (FLAG = -1.) or downstream (FLAG = 1.) position CX at which the geometric area ratio is equal to AFNX.

If the boundary layer is included, the input value of DELT1 is saved in the location DELTSV, and DELT1 is reset to 0.049(1. - CTMAX) during initialization. This ensures that there will be 20 flow points upstream of the throat. In some cases with CTMAX $\sim$ 0.9, the normal input value of DELT1 = 0.01 would give only about 10 points above the throat, which would adversely affect the accuracy of the numerical integration in the boundary layer routine BLAYER. Once the throat has been reached, DELT1 is restored to its original value by the statements below 320.

When the boundary layer is included, the effective area ratio A_e = AFNX is no longer equal to the geometric area ratio A_g = AR, because of the boundary layer displacement thickness, δ^*. The relation between these two area ratios (Section 4.4 of Volume I) involves the displacement thickness at the throat, δ^*_*. Since this is not known initially, the spatial position CX corresponding to each flow point is calculated on the assumption that $A_g \simeq A_e$ until the throat has been reached. These calculations are done using subroutine FINDX, as in the case without the boundary layer. A special flow point is computed at the throat (ISØNIC = 0, CT = CTSTAR, CX = 0.) in order to determine the quantity

$1 - \delta_*^* =$ ØMDST (statement 390) for the nozzle (or two such values in the case of a channel). Then, beginning at the third point beyond the throat, CX is computed taking the displacement thickness into account, using subroutine AGSØLN. In statement 340, AGSØLN is called with the effective area ratio AFNX for the current step and the displacement thickness DELBL for the preceding step. The resulting CX value is used in the boundary layer calculations (statement below 370). Subroutine AGSØLN is then called again, with the same AFNX and the new DELBL produced by the call to BLAYER. This second call to AGSØLN gives an improved value of CX.

In some problems, at large distances downstream of the throat, the displacement thickness becomes large enough to have a major effect on the effective area ratio. Under such circumstances, the coupling between the boundary layer and the inviscid flow solution can lead to computational instability. The boundary layer routine BLAYER contains a smoothing algorithm intended to suppress such instability (Section 5.11 of Volume I). This algorithm involves a parameter W, preset to a value WSAVE which usually prevents instability. However, occasionally instability develops in spite of the smoothing. In the frozen and equilibrium solutions, the symptom of such instability is that CX begins to decrease with decreasing temperature. When this occurs, subroutine FRØZEQ attempts to produce a valid solution by cutting the stabilizing parameter W in half and restarting the solution in the reservoir. This operation is accomplished by coding between statements 340 and 350. If instability has been encountered after three successive restarts, the solution is abandoned and error dumps are written. Also, restarting is not attempted if the geometric area ratio is more than 4 times greater than the effective area ratio at the point where CX decreases, as in this case the code is probably predicting actual choking of the flow by boundary layer growth.

3.22.4 Model Points

Model points are input-specified spatial locations, in the downstream region of the nozzle, at which special calculations are to be performed. Since model points are

specified by values of CX, while points of the equilibrium or frozen flow solution are specified by values of the temperature, an iteration is required to find the temperature CT corresponding to each model point.

During the initialization, and after the completion of each model point calculation, subroutine NEXTMP is called to determine the location XMØDEL of the next model point. During the main flow solution, after CX has been determined for each flow point, statement 450 tests whether CX is greater than or equal to XMØDEL. If not, the flow solution proceeds normally. If CX $\geq$ XMØDEL, the iteration to determine the CT value corresponding to CX = XMØDEL is started. While this iteration is in progress, the indicator MØDLPT is .TRUE..

To avoid complicating the programming of subroutine BLAYER, and to save computer time, BLAYER is not called during the model point iteration. Instead, the boundary layer properties at the model point are approximated by linear interpolation of the data at the preceding and current main flow points, and at each cycle of the iteration subroutine TRANSP is called (statement 360) to determine the transport properties at the current temperature.

The iterative algorithm for improving the estimate of CT corresponding to XMØDEL is a numerical Newton-Raphson technique embodied in statement 490:

$$T = T_1 + (T_2 - T_1) \cdot \frac{X_m - X_1}{X_2 - X_1} \qquad (80)$$

where T_2 is the temperature for which calculations were done in the current step, T_1 is the temperature in the preceding iterative step, X_2 and X_1 are the CX values corresponding to T_2 and T_1, and X_m is XMØDEL.

The convergence criterion for the iteration is that CX differ from XMØDEL by no more than 0.0025 cm or 0.001 inch. A maximum of 10 iterations is allowed. If convergence has not been achieved after 10 iterations, a diagnostic message

is written and the data from the final iteration are used. When the iteration has thus been terminated by either convergence of nonconvergence, the results for the flow conditions at the model point are printed by calling ØUT2, MØDLPT is reset to .FALSE., and subroutine MØDEL is called to perform the required calculations of conditions on models. Also, the indicator SKIP is set to .TRUE., and the next XMØDEL is determined by calling NEXTMP. Before resuming the main flow solution, FRØZEQ checks whether the next XMØDEL is less than or equal to the CX for the current main solution point. If so, the model point iteration is started again to find the flow conditions at the new XMØDEL.

Once all of the XMØDEL values less than or equal to the CX for the current main flow point have been dealt with, the flow conditions for the current flow point are recomputed and printed. The .TRUE. value of the indicator SKIP provides for this recomputation by causing a transfer to statement 220 instead of 550 just below statement 450. Also for SKIP = .TRUE., the determination of CX and the call to BLAYER are bypassed below 220. SKIP is then reset to .FALSE., the results are printed by calling ØUT2, and the main flow solution is resumed.

When a CX greater than or equal to CXMAX is reached, XMØDEL is set equal to CXMAX and the indicator LASTPT is set to .TRUE.. The model point iteration then provides the flow conditions at the position CXMAX. After the iteration has converged and the results have been printed, for LASTPT =.TRUE., control is transferred to statement 610. For nozzle flow problems (NPRFLS = 1), the RETURN is then executed. For channel problems (NPRFLS = 2), subroutine MØDEL is first called with AXISYM = .TRUE. to provide calculation of the stagnation conditions in the channel exit plane. Stagnation condition calculations are suppressed for channels, with this one exception, because stagnation point models and Pitot probes are not used inside channels.

3.23 Subroutine GEØM

The function of GEØM is to calculate the density, the effective area ratio and the derivative of one or the other of these quantities at each position x along the nozzle during

the nonequilibrium flow solution. Two different methods of calculation are used in different parts of the flow solution. During the solution by the perturbation method, and during the direct integration of the rate equations in the region beyond the call to THRØAT, the calculation of the effective area ratio is based directly on the nozzle geometry and the boundary layer displacement thickness, if any, and the density is then calculated from the continuity equation. During the numerical integration by the inverse method upstream of the call to THRØAT, the density is obtained from the analytical density-area relation I(383):

$$\left(\frac{\rho}{\rho_0} A_e\right)^2 \left[1 - \left(\frac{\rho}{\rho_0}\right)^{\alpha}\right] = C \qquad (81)$$

and the effective area ratio is then obtained from the continuity equation I(266). The constants α and C in (81) are determined in the main program from sonic-point conditions, prior to the first call to GEØM, as explained in Section 7.4 of Volume I and in Section 3.1 of the present report.

The overall logical structure of GEØM is diagrammed in figure 12. Immediately upon entry, the subroutine calls GEØMAR to compute the geometric area ratio S1 and its derivative S2 at the current position CX. If the nonequilibrium flow solution being computed includes the boundary layer (ISW3B $\neq$ 0), then AESØLN is called to convert S1 into the effective area ratio and S2 into the derivative of the effective area ratio, based on the current displacement thickness.

If the flow solution is beyond the call to THRØAT (IUPD = 0) or if the perturbation technique is in use (INEQ = 0), the effective area ratio AFNX is set equal to S1 and the nondimensional density RHØ is calculated from AFNX using the continuity equation. Also, the derivative DLØGA = $d\ell n\, A_e/dx$ is computed from S1 and S2.

On the other hand, if the numerical integration technique and the inverse method are being used to generate the flow solution (INEQ = 1 and IUPD = 1), then the nondimensional

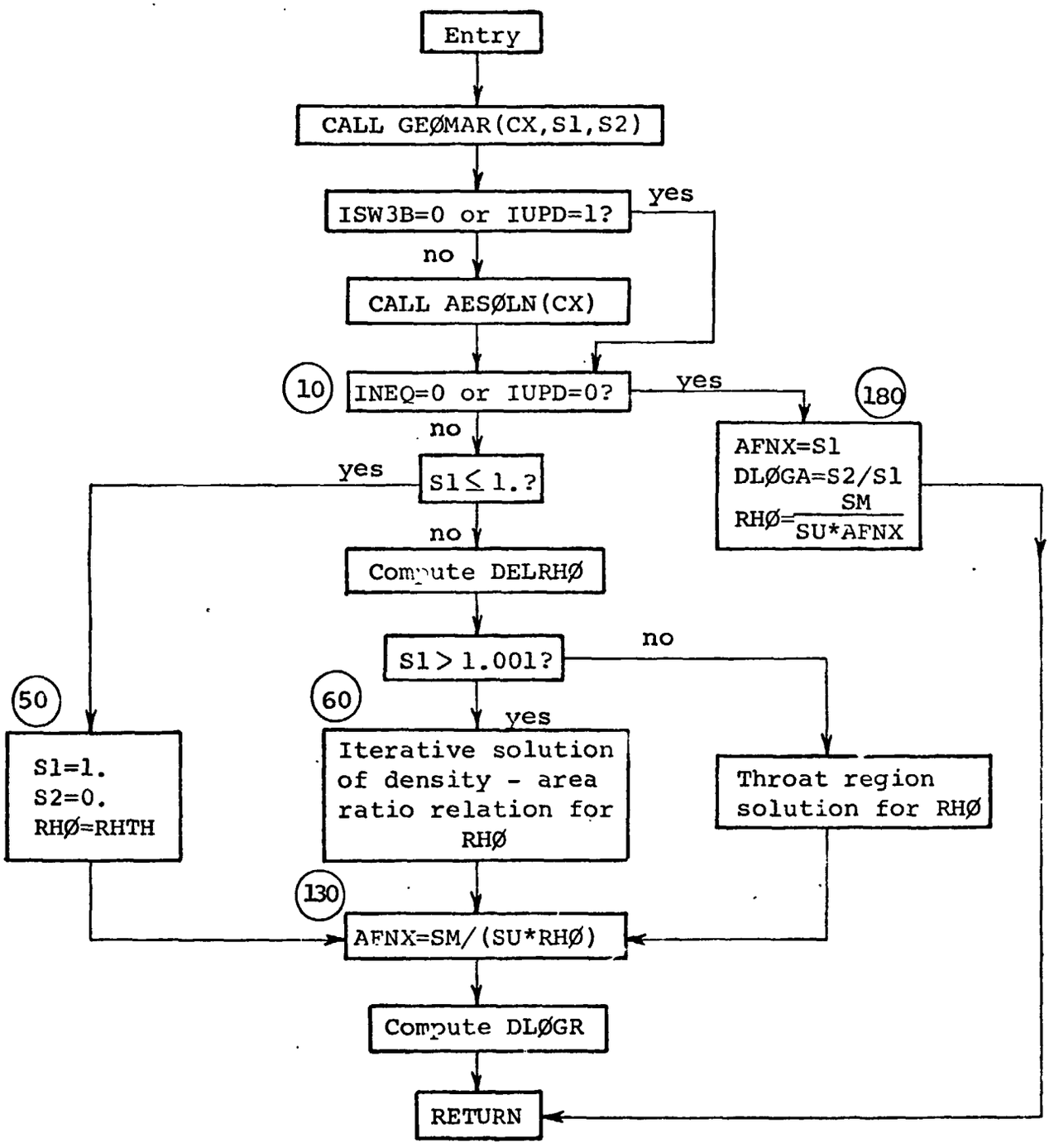


Figure 12. Flowchart of Subroutine GEØM

density RHØ is calculated from S1 by solving the density-area ratio relation (81), and the effective area ratio AFNX is recomputed from RHØ using the continuity equation. The derivative DLØGR = $d\ln(\rho/\rho_0)/dx$ is then computed for use in the rate equations (see the discussion of subroutine EXACT above).

The calculation of RHØ from equation (81) is based on the following analysis. Equation (81) can be solved for A_e:

$$A_e = \frac{\rho_0}{\rho}\sqrt{\frac{C}{1-(\rho/\rho_0)^{\alpha}}} \tag{82}$$

The form of this relation is illustrated in figure 13, for the case $\alpha = 0.2$, $C = 0.035049$. The corresponding density ratio at the sonic point is $\rho_*/\rho_0 = 0.62092$. The basic computational problem in GEØM is to solve (81) or (82) for ρ/ρ_0, based on a given value of A_e. Except near the throat, the solution is carried out using the Newton-Raphson method; i.e., from a given estimate ρ_p of ρ, an improved estimate ρ_n is obtained as follows:

$$A_p = \frac{\rho_0}{\rho_p}\sqrt{\frac{C}{1-(\rho_p/\rho_0)^{\alpha}}} \tag{83a}$$

$$\rho_0\left(\frac{dA}{d\rho}\right)_p = \frac{\rho_0 A_p}{\rho_p}\cdot\frac{(1+\tfrac{1}{2}\alpha)(\rho_p/\rho_0)^{\alpha}-1}{1-(\rho_p/\rho_0)^{\alpha}} \tag{83b}$$

$$\rho_n = \rho_p + \frac{A_e - A_p}{(dA/d\rho)_p} \tag{83c}$$

Here A_p is the A_e value calculated from (82) for $\rho = \rho_p$; $(dA/d\rho)_p$ is the derivative of (82) evaluated at $\rho = \rho_p$; and equation (83c) is the Newton-Raphson formula.

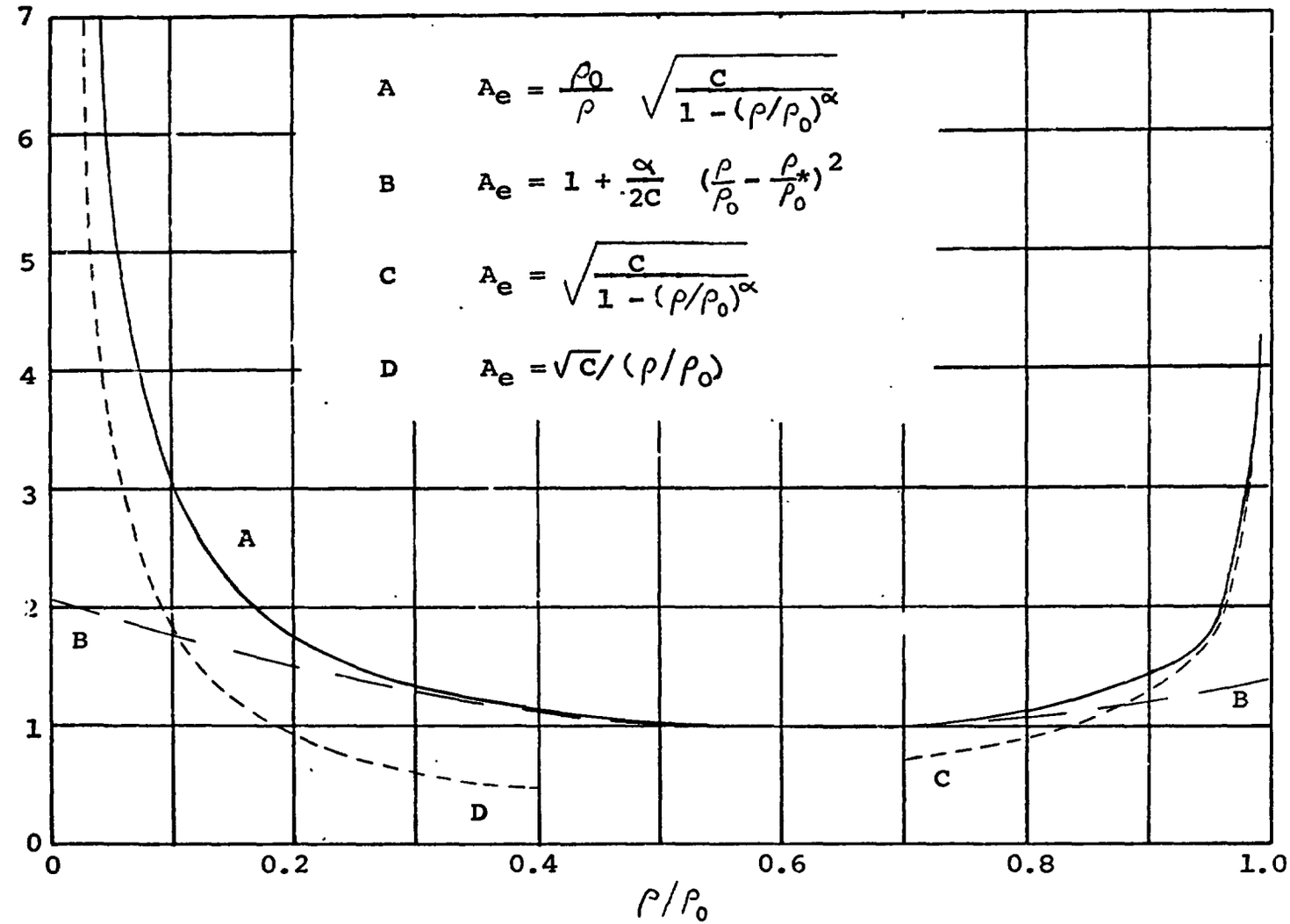


Figure 13. Approximations to the Density-Area Ratio Relation

Examination of equations (83) reveals several potential problems. At the sonic point ($A_e = 1$), the derivative $dA_e/d\rho$ is zero, so that the second term in (83c) is undefined. Near the throat, $(dA/d\rho)_p$ is very small. If a trial value ρ_p is near the throat, and nearer the throat than the solution of (82), the Newton-Raphson formula (83c) can give an "improved" ρ_n which is much less accurate than ρ_p and which may even be outside the permissible range $0 \leq \rho \leq 1$. To deal with these problems reliably, GEØM uses the following algorithm:

(1) If $A_e \leq 1$, $\rho = \rho_*$ (throat point)

(2) If $1 < A_e \leq 1.001$ (throat region), ρ is calculated from a Taylor series expansion of (82) around the throat point:

$$A_e = 1 + \tfrac{1}{2}\left(\frac{d^2A_e}{d\rho^2}\right)_{\rho=\rho_*}(\rho - \rho_*)^2 + \frac{1}{6}\left(\frac{d^3A_e}{dx^3}\right)_{\rho=\rho_*}(\rho-\rho_*)^3$$

$$= 1 + \frac{\alpha}{2C}\left(\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0}\right)^2 + \frac{(\alpha+1)\alpha^{3/2}}{6C\sqrt{C(\alpha+2)}}\left(\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0}\right)^3 \qquad (84)$$

The coefficients in this equation have been evaluated by differentiation of (82) and use of the sonic-point conditions $\rho = \rho_*$, $A_e = 1$, $(d A_e/d\rho)_* = 0$. In the throat region, where (84) is used, the cubic term is small. If this term were neglected, the solution of (84) would be

$$\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0} = S\sqrt{\frac{2C}{\alpha}(A_e - 1)} \equiv S\cdot(\delta\rho) \qquad (85)$$

where S is +1 upstream and -1 downstream of the throat. When the cubic term is taken into account, the solution of (84) can be approximated by

$$\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0} = S\left[\frac{2C}{\alpha}(A_e-1) - \frac{S(\alpha+1)}{3}\sqrt{\frac{\alpha}{C(\alpha+2)}}\cdot(\delta\rho)^3\right]^{\frac{1}{2}} \quad (86)$$

where $(\delta\rho)$ is defined by the second equality in (85). Equation (86) is used to calculate ρ/ρ_0 when $A_e < 1.001$.

(3) If $A_e > 1.001$ (general case), the Newton-Raphson procedure (83) is used. To ensure its stability, pains are taken to select a first estimate of ρ which is farther from ρ_* than the solution point. Since the relation (82) is everywhere concave upward (figure 13), the Newton-Raphson formula undercorrects ρ whenever the estimate ρ_p lies too far from ρ_*. Thus, if the first estimate ρ_p is too far from ρ_* the iteration converges to the solution without ever overshooting it. Under these circumstances, no instability can develop. To obtain such an initial estimate, the subroutine uses three approximations to (82), all of which lie below (82). One of these approximations is (85). The second is an approximation for $\rho/\rho_0 \sim 1$, namely

$$A_e \simeq \sqrt{\frac{C}{1-(\rho/\rho_0)^\alpha}}$$

from which

$$\frac{\rho}{\rho_0} \simeq \left(1 - \frac{C}{A_e^2}\right)^{1/\alpha} \quad (87)$$

The third is an approximation for $\rho \sim 0$,

$$A_e \simeq \sqrt{C}/(\rho/\rho_0)$$

from which

$$\frac{\rho}{\rho_0} \simeq \sqrt{C}/A_e \quad (88)$$

In the upstream region ($dA_e/dx < 0$), the initial estimate is chosen to be

$$\frac{\rho_p}{\rho_0} = \min\left[\frac{\rho_*}{\rho_0} + \sqrt{\frac{2C(A_e-1)}{\alpha}},\ \left(1 - \frac{C}{A_e^2}\right)^{1/\alpha}\right] \quad (89)$$

In the downstream region ($dA_e/dx > 0$), it is

$$\frac{\rho_p}{\rho_0} = \max\left[\frac{\rho_*}{\rho_0} - \sqrt{\frac{2C(A_e - 1)}{\alpha}},\ \frac{\sqrt{C}}{A_e}\right] \tag{90}$$

Once the solution for ρ has been obtained, the derivative $d\ell n\,\rho/dx$ is calculated. Normally this is determined from the formula

$$\frac{d\ell n\rho}{dx} = \frac{2C}{\alpha\left(\frac{\rho A_e}{\rho_0}\right)^2 - (\alpha + 2)C} \cdot \frac{d\ell n\ A_e}{dx} \tag{91}$$

which can be obtained by differentiating (82). The factor $d\ell n\ A_e/dx$ on the right is evaluated as S2/S1, based on the specified nozzle geometry and displacement thickness. The throat point ($A_e = 1$) is a special case, because there the denominator of (91) is zero and $dA_e/dx = 0$, so that (91) is of the form 0/0. At the throat, $d\ell n\rho/dx$ must be evaluated from (91) using L'Hospital's rule:

$$\left(\frac{d\ell n\rho}{dx}\right)_{A_e=1} = 2C \lim_{A_e \to 1} \frac{d^2\ell n\ A_e/dx^2}{\frac{2\alpha}{\rho_0}\left(\rho\frac{dA_e}{dx} + A_e\frac{d\rho}{dx}\right)}$$

Since $(dA_e/dx)_* = 0$, this gives

$$\left(\frac{d\ell n\rho}{dx}\right)_* = -\sqrt{\frac{C(d^2\ell n\ A_e/dx^2)_*}{\alpha\,\rho_*/\rho_0}} \tag{92}$$

The area derivative on the right hand side has to be evaluated from the specified nozzle geometry. In general,

$$\left(\frac{d^2\ell n\ A_e}{dx^2}\right)_* = \left(\frac{d^2A_e}{dx^2}\right)_* \tag{93}$$

by virtue of the conditions $A_{e*} = 1$, $(dA_e/dx)_* = 0$. To proceed further, it is necessary to consider the three geometry options separately:

(1) Two-dimensional nozzle

In this case, from equation I(126) with y_* replaced by y_0 as usual,

$$\left(\frac{d^2A_e}{dx^2}\right)_* = \frac{\frac{1}{y_0}\left(\frac{d^2y}{dx^2}\right)_* - \frac{1}{y_0}\left(\frac{d^2\delta^*}{dx^2}\right)_*}{1 - \delta^*_*/y_0} \qquad (94)$$

The term $(d^2\delta^*/dx^2)_*$ is small, because the displacement thickness δ^* is small and slowly varying at the throat. This term is therefore neglected. Since the throat sections of the profile fit are always of the form I(122) (shape 2, circular arc convex toward the axis), the remaining term in the numerator of (94) is given by

$$\frac{1}{y_0}\left(\frac{d^2y}{dx^2}\right)_* = \frac{1}{y_0}\left(\frac{d^2y}{dx^2}\right)_0 = \frac{1}{y_0 P_3}$$

The second equality is obtained by noting that $P_2=0$ for a throat section. Thus,

$$\left(\frac{d^2A_e}{dx^2}\right)_* = \frac{1}{y_0 P_3 (1 - \delta^*_*/y_0)} \qquad (95)$$

(2) Axisymmetric nozzle

From equation I(131) since $(dA_e/dx)_* = 0$ and $A_{e*} = 1$,

$$\left(\frac{d^2A_e}{dx^2}\right)_* = \frac{1}{1 - \delta^*_*/y_0}\left[\frac{1}{\sqrt{A_{g*}}}\left(\frac{d^2A_g}{dx^2}\right)_* - \frac{1}{2A_{g*}^{3/2}}\left(\frac{dA_g}{dx}\right)_*^2 - \frac{2}{y_0}\left(\frac{d^2\delta^*}{dx^2}\right)_*\right] \qquad (96)$$

Now dA_g/dx is very small at the sonic point (which is quite near the throat), so that the square of this quantity can be neglected. The quantity $(d^2 \delta^*/dx^2)_*$ is also neglected for reasons discussed above. Hence,

$$\left(\frac{d^2A_e}{dx^2}\right)_* \simeq \frac{(d^2A_g/dx^2)_*}{\sqrt{A_{g*}}(1 - \delta^*_*/y_0)} \qquad (97)$$

Evaluation of $(d^2A_g/dx^2)_*$ using I(116) and I(122) then gives

$$\left(\frac{d^2A_e}{dx^2}\right)_* \simeq \frac{2}{y_0 P_3 (1- \delta^*_*/y_0)\sqrt{A_{g*}}} \qquad (98)$$

(3) Rectangular channel

From equation I(134),

$$\left(\frac{d^2A_e}{dx^2}\right)_* = \frac{1}{D}\left[(y- \delta^*_1)\left(\frac{d^2z}{dx^2} - \frac{d^2\delta^*_2}{dx^2}\right) + (z- \delta^*_2)\left(\frac{d^2y}{dx^2} - \frac{d^2\delta^*_1}{dx^2}\right) + 2\left(\frac{dy}{dx} - \frac{d\delta^*_1}{dx}\right)\left(\frac{dz}{dx} - \frac{d\delta^*_2}{dx}\right)\right]_* \qquad (99)$$

where

$$D = (y_0 - \delta^*_{1*})(z_0 - \delta^*_{2*}) \qquad (100)$$

At the sonic point, the quantities $\left(\frac{dy}{dx} - \frac{d\delta^*}{dx}\right)$ and $\left(\frac{dz}{dx} - \frac{d\delta^*}{dx}\right)$ are small, so that their product is considered negligible. The quantities $(d^2\delta^*_i/dx^2)_*$ are also neglected, as in the other options. Also, $y_* \simeq y_0$ and $z_* \simeq z_0$. Hence,

$$\left(\frac{d^2A_e}{dx^2}\right)_* \simeq \frac{(y_0 - \delta_{1*}^*)(d^2z/dx^2)_* + (z_0 - \delta_{2*}^*)(d^2y/dx^2)_*}{(y_0 - \delta_{1*}^*)\ (z_0 - \delta_{2*}^*)} \qquad (101)$$

Evaluation of $(d^2z/dx^2)_*$ and $(d^2y/dx^2)_*$ from I(122) gives, finally,

$$\left(\frac{d^2A}{dx^2}\right)_* \simeq \frac{1}{(z_0 - \delta_{2*}^*)\ P_{3z}} + \frac{1}{(y_0 - \delta_{1*}^*)\ P_{3y}} \qquad (102)$$

where P_{3z} is the P_3 parameter for the z profile and P_{3y} that for the y profile.

3.24 Subroutine GEØMAR

GEØMAR is the basic geometry routine in the NATA code. Through its various entries, it provides calculations of the geometric area ratio and its derivative and of profile ordinates and their derivatives both for nozzles and for rectangular channels.

The subroutine has 4 entries, with the following calls:

GEOMAR(X,ARATIØ, DERIVA) (IENTRY = 1)

GMAR(X,Y) (IENTRY = 2)

GMAR2(X,Y,Z) (IENTRY = 3)

GMAR3(X,DYDX,DZDX,Y,Z) (IENTRY = 4)

The arguments have the following meanings:

X = axial coordinate in nozzle (cm)

ARATIØ = geometric area ratio

DERIVA = derivative of geometric area ratio (cm^{-1})

Y = profile ordinate (cm)

Z = ordinate for second profile in a channel (cm)

DYDX = derivative of profile ordinate

DZDX = derivative of ordinate for second profile in a channel

For each profile (I) the subroutine first determines the profile section (J) in which the specified value of X lies. This is the first section for which X < ATPI(J,I). The routine then calculates the profile ordinate from equation I(121), I(122), or I(123) depending on ISHAPE(J,I), and if required also calculates the derivative of the profile ordinate using the derivatives of these formulas:

(1) Straight Line, ISHAPE(J,I) = 1

$$\frac{dy}{dx} = P_2 \qquad (103)$$

(2) Circular Arc Convex Downward, ISHAPE(J,I) = 2

$$\frac{dy}{dx} = \frac{x - P_2}{\sqrt{P_3^2 - (x - P_2)^2}} \qquad (104)$$

(3) Circular Arc Concave Downward, ISHAPE(J,I) = 3

$$\frac{dy}{dx} = - \frac{x - P_2}{\sqrt{P_3^2 - (x - P_2)^2}} \qquad (105)$$

The subroutine then calculates the geometric area ratio, if required, from equation I(118) for a two-dimensional nozzle (NPRFLS = 1, JDIM = 0), I(116) for an axisymmetric nozzle (NPRFLS = 1, JDIM = 1), or I(120) for a rectangular channel (NPRFLS = 2). It calculates the derivative of the area ratio from the derivatives of these formulas, i.e.,

$$\frac{dA_g}{dx} = \frac{1}{y_0} \frac{dy}{dx} \qquad \text{(2D nozzle)} \qquad (106)$$

$$\frac{dA_g}{dx} = \frac{2y}{y_0^2} \frac{dy}{dx} \qquad \text{(axisymmetric nozzle)} \qquad (107)$$

$$\frac{dA_g}{dx} = \frac{y \frac{dz}{dx} + z \frac{dy}{dx}}{y_0 z_0} \qquad \text{(channel)} \qquad (108)$$

Before executing the RETURN, the routine rescales all of the computed results using the rescaling factor RSA. The area ratio and its derivative are multiplied by RSA in the case of a 2D nozzle, but in an axisymmetric nozzle or channel they are multiplied by $\sqrt{RSA}$. The factor RSA is initialized to 1.0 and is never changed in equilibrium and frozen flow solutions. In nonequilibrium flow solutions, RSA is reset in subroutine THRØAT so as to make the effective area ratio continuous between the upstream solution by the inverse method and the downstream direct solution.

3.25 Subroutine INGAS

This subroutine has two entry points, INGAS and INTA. The first part of the routine, which is entered by calling INGAS, sets up several arrays which are used in the chemical description of the gas mixture, as discussed in Section 2.1 of Volume I. The mathematical and Fortran notations for these arrays correspond as follows:

A_{ki}	AIN(K,I)	
$\bar{\nu}_{i-c,j}$	CDIJ(I,J)	(I = i - c)
$-1 + \nu^*_{i-c}$	BET(I)	(I = i - c)
q_i	QM(I)	
W_i	CGI(I)	

In the subroutine, the array AIN is first initialized to the square submatrix $\bar{\alpha}_{ij}$ of the matrix α_{ij} with i = 1 to c, where α_{ij} is the number of atoms of the j^{th} element

per molecule of the i^{th} species, and c denotes the number of elements in the system. The inverse matrix A_{ki} of $\bar{\alpha}_{ij}$ is then computed and stored in AIN by calling subroutine MATINV.

Next, the matrix $\bar{\nu}$, which gives the composition of the dependent species in terms of the components, is then computed by evaluating equation I(6), and is stored in CDIJ. Then the singly subscripted array $-1+\nu^*_{i-c}$, where ν^* is defined by equation I(15), is computed in BET. The array q_i giving the invariant composition of the gas in terms of the independent species is calculated using equations I(9) and I(10); it is denoted by QM(I). Finally, the species molecular weights W_i = CGI(I) are computed as follows:

$$c_i = \sum_{j=1}^{c} \alpha_{ij} \; A_j \qquad (109)$$

where A_j is the atomic weight of the j^{th} element;

$$W_i = c_i \qquad (i = 1, \ldots, c) \qquad (110a)$$

$$W_i = \sum_{k=1}^{c} \bar{\nu}_{i-c,k} \; W_k \qquad (i = c + 1, \ldots, n) \qquad (110b)$$

For the rationale of equation (110b), see equation I(5) and its associated discussion. The molecular weights of the dependent species i = c + 1,..., n could also be calculated directly using equation (109), in place of (110b).

The purpose of the calculations following entry INTA is to compute the equilibrium conditions in the upstream reservoir. The actual thermochemical equilibrium calculation is performed by subroutine EQCALC(CTAP,PRESA). The argument CTAP is the reservoir temperature in degrees Kelvin, and PRESA is the reservoir pressure in atmospheres.

The results of the calculations are communicated from EQCALC to INGAS through the common blocks /EQC/ and /EQC2/.

3.26 Subroutine INIT

Subroutine INIT initializes a number of control parameters and nondimensionalizes the species thermochemical data using the reservoir temperature. Since the coding is straightforward, it will not be discussed.

3.27 Subroutine KANDMU

Subroutine KANDMU calculates quantities proportional to the viscosity and the translational thermal conductivity of the gas mixture from equations I(85) to I(87). The first section of the subroutine, down through statement 20, computes the quantities $(25/8)\sqrt{T}\ A_i^{(K)}$ and $(25/8)\sqrt{T}\ A_i^{(\mu)}$ defined by equation I(87), for all species i in the mixture. The value of $(25/8)\sqrt{T}\ A_i^{(\mu)}$ is obtained simply by dividing the factor ZM2(I) out of the quantity B(I,2) computed previously in subroutine KINT, equation (113) below, and is stored again in the array B(I,2), while the quantity $(25/8)\sqrt{T}\ A_i^{(K)}$ is computed directly from I(87) using the matrix elements Q(3,I,J) calculated previously in subroutine TRANSP. The value of $(25/8)\sqrt{T}\ A_i^{(K)}$ is stored in the location B(I,1), replacing the previous quantity set in KINT, which is no longer needed. The reciprocals of the elements B(I,L) are also computed and stored in BR(I,L).

The next section of subroutine KANDMU, from statement 20 down through statement 70, computes the quantities

$$A(1) = (25/8)\sqrt{T}\ \bar{a}^{(K)}$$

and (111)

$$A(2) = (25/8)\sqrt{T}\ \bar{a}^{(\mu)}$$

from equation I(86), using the values of the matrix elements $a_{ij}^{(K)}$ and $a_{ij}^{(\mu)}$ which were previously stored in the lower halves (I > J) of the matrices Q(1,I,J) and Q(2,I,J)

by subroutine TRANSP. Since the values of $\bar{a}^{(\alpha)}$ for the viscosity and thermal conductivity are both computed from the same equation, the same coding can be used here for both properties, with the index L = 1 indicating values computed for the thermal conductivity and L = 2 values computed for the viscosity. In connection with this computation, it may be noted that the sum in equation I(86) need only be carried over values of I > J, since the matrix $a_{ij}^{(\alpha)}$ is symmetric and the diagonal terms do not contribute to the sum.

The final loop in subroutine KANDMU down through statement 90 computes the values of the subroutine arguments ZK(1) and ZK(2), which are proportional respectively to the translational thermal conductivity and the viscosity of the gas mixture. These values are computed from equation I(85), using the values of the elements B(I,L) and A(L) which were computed earlier in the subroutine

3.28 Subroutine KINT

The statements down through statement 10 of subroutine KINT compute the quantities

$$B(I,1) \equiv \frac{1}{k}\sqrt{T} \sum_{j=1}^{n} x_j \Delta_{ij}^{(1)} \qquad (112)$$

and

$$B(I,2) \equiv \frac{5}{6k}\sqrt{T} \sum_{j=1}^{n} x_j \Delta_{ij}^{(2)}$$

$$= \frac{25}{8}\sqrt{T}\ ZM2(I) \sum_{j=1}^{n} x_j A_{ij}^{(\mu)} \qquad (113)$$

which are required in the transport property calculations. Because of the way in which the data $\Delta_{ij}^{(\ell)}$ are stored in the Q array by subroutine TRANSP, it is necessary to treat the terms with i < j and with i > j separately in these

computations; thus the statement preceding 10 carries out the sum over terms with $i > j$ and statement 10 the sum over terms with $i < j$. The diagonal terms with $i = j$ were added previously during the initialization of the sum by subroutine TRANSP.

Following the computation of the B(I,1) and B(I,2), subroutine KINT uses the computed values of the B(I,1) in the small loop ending at statement 30 to calculate the value of the argument ZKINT $\equiv K_{int}/\sqrt{T}$ in units of milliwatts/cm-$^{o}K^{3/2}$, where K_{int} is the internal component of the thermal conductivity of the gas given by equation I(95).

3.29 Subroutine LIST

Subroutine LIST prints out certain portions of the input data for each case so as to provide a description of the problem which is to be solved by the code. Subroutine READ also provides some output of similar character.

Portions of the output normally produced by LIST are omitted if they would simply repeat the corresponding output for a previous case. If SUPGØ = .TRUE., the gas model is the same as in the preceding case, and the description of the gas model is skipped. If SUPGØ is .FALSE., the following tables are printed:

(1) The elemental composition of the gas in terms of "atom fractions", defined as the number of gram-atoms of each element per mole of the mixture of cold species. This table also lists the atomic weights assumed for the elements.

(2) The data defining the forward rate constant for each of the reactions. This table also includes the criterion value CCHI for switching from the perturbation technique to numerical integration, and the third body matrix for the reactions involving third bodies.

(3) A table which defines the chemical formula for each species by giving the α_{ij} = LPIJ(I,J) matrix. A separate column of this table contains 1 if a thermo fit is available for the species, 0 if not.

(4) The matrix ν'_{ij} of the stoichiometric coefficients for the product side of all the reactions.

(5) The matrix ν_{ij} of the stoichiometric coefficients for the reactant side of all the reactions.

(6) A table listing the data for calculations of species thermal properties based on the physical model.

(7) A table listing the electronic energy levels and degeneracies for all the species for which physical model data are provided.

(8) A table of the thermo-fit coefficients for those species for which thermo fits are provided.

If an electronic nonequilibrium gas model is being used (NT=2), then a table giving the extra reaction data required by this model is inserted before the thermo-fit table (8).

After the gas model description has been printed out, LIST prints a summary of the input data for model and wedge calculations.

3.30 Subroutine MATINV

This is a matrix inversion subroutine. It is called by INGAS to compute the inverse A_{ki} of the matrix $\bar{\alpha}_{ij}$; see equation I(3) and associated discussion. MATINV computes the inverse of matrix A using the Gauss-Jordan reduction with the maximum pivot strategy (ref. 8).

3.31 Subroutine MØDEL

Subroutine MØDEL is the principal routine for calculations of conditions on models in the NATA code. MØDEL performs normal-shock calculations, computes conditions at the

inviscid stagnation point, calculates stagnation point heat fluxes, and provides output of these results. In addition, if conditions on wedge models are to be calculated, MØDEL calls subroutine WEDGE. The overall structure of MØDEL is diagrammed in figure 14.

3.31.1 General Description

Immediately upon entry, MØDEL performs some checks to determine whether model condition calculations should be done. Such calculations are skipped if both of the logical variables AXISYM and WEDGEM are false, or if the free-stream Mach number is less than 1.5. The latter restriction is imposed because the normal shock solution fails to converge when the Mach number is below some value in the range from 1.5 to about 2. Models are not normally tested at such low Mach numbers in arc-heated wind tunnels because the corresponding nozzle diameter is too small to accommodate models of reasonable size.

If model calculations are to be done, subroutine ELTIME is called with the argument IP = 0 to record the execution time at the beginning of the model calculations. ELTIME is called again, just prior to the RETURN, to determine the CPU time elapsed during the model calculations. This time is printed out. It is usually about 1 to 3 seconds.

Before beginning the normal shock solution, MØDEL sets a number of flow quantities which depend upon the free-stream flow conditions and are thus constant for a given call to MØDEL. When MØDEL is first called in a given flow solution, the equilibrium and frozen shock density ratios EPSLØN and EPSF have the value zero, set in the calling routine (FRØZEQ or NØNEQ). These ratios are reset to the value 0.01. When MØDEL is called with EPSLØN = 0, it computes the Prandtl number at the model surface temperature, PRW, based on the cold species mole fractions QPJ(K). This value of PRW thus assumes that the gas at the model surface is in a state of thermochemical equilibrium at the surface temperature.

Most of the calculations in MØDEL are contained within a large DØ loop with the DØ index ISØLN, beginning below

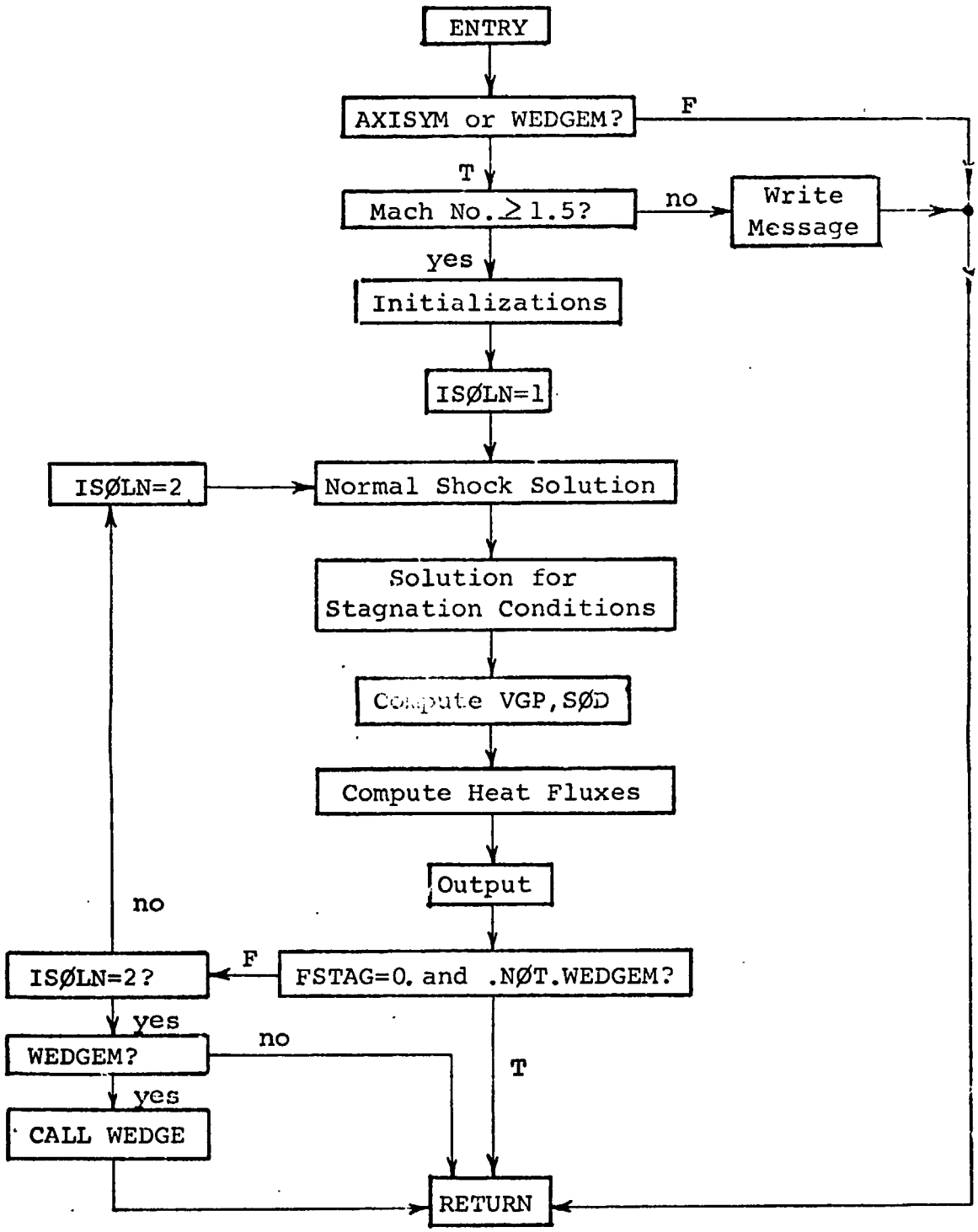


Figure 14. Overall Flowchart of Subroutine MØDEL

statement 40 and ending at 440. For ISØLN = 1, the normal shock is calculated assuming chemical equilibrium behind the shock. For ISØLN = 2, the flow across the shock is assumed to be chemically frozen. If the control variable FSTAG is 0, the frozen shock calculations are skipped unless wedge calculations are to be done (WEDGEM = .TRUE.). If wedge calculations are called for, the frozen shock solution is carried out, regardless of the value of FSTAG, to provide the frozen stagnation temperature which is required for use in subroutine WEDGE. If FSTAG is negative, the equilibrium shock calculations are skipped.

For either value of ISØLN, the conditions behind the normal shock and those at the inviscid stagnation point are calculated using methods discussed below in Sections 3.31.2 and 3.31.3. Then the stagnation-point velocity gradient parameter $(R_b/u_1)(du_e/dx)_s$, denoted by VGP(L), is computed for spherical or cylindrical nosed models (L = 1) and for flat-faced models (L = 2). The ratio SOD(L) of the shock standoff distance to the model radius is also calculated for L = 1 and 2. These calculations are based upon equations I(463) to I(465) for the velocity gradient parameter. and I(475) to I(478) for the standoff distance.

Next, the various heat flux values discussed in Section 8.1.4 of Volume I are computed. In preparation for these calculations, the dissociation enthalpy h_D of the gas at the inviscid stagnation point (denoted by HCF) is computed by summing the product of the mole fraction and the enthalpy of formation SHJA(I) over all neutral atomic species (see equation I(544)). In the case of an equilibrium shock, the mole fractions SAVEC(I) at the stagnation point are obtained from the common array ZCAP(I), which is set in subroutine EQCALC. In the case of a frozen shock, the mole fractions are the same as those in the free stream flow ahead of the shock, and are calculated from the concentrations GJ(I) using equation I(1). In the evaluation of HCF, the sum is restricted to neutral atoms by rejecting all species containing more than one element, more than one atom of an element, or a negative number of atoms of an element.* This algorithm

*Positive ions are represented as compounds containing a negative number of electrons.

accepts the electron species, which of course is not a neutral atom, but this does not affect HCF because the formation enthalpy of the electron is zero.

The transport properties μ_e(VISC), N_{Pr}(PR), N_{Le}(LE), and σ (SIGMA) at the inviscid stagnation point are computed by calling subroutine TRANSP. The heat fluxes are calculated using equations I(466), I(468) and I(469) with suitable conversion factors to obtain English engineering units in the output. For example, the numerical coefficient in the Fortran formula corresponding to equation I(466) is obtained as follows:

$$0.51 = \frac{1.8 \times (30.48)^{3/2}}{453.6} \times 0.763$$

The factor of 1.8 converts the enthalpies from internal code units of cal/gm to Btu/lb; $(30.48)^{3/2}$ converts $cm^{\frac{1}{2}}/cm^2$ to $ft^{\frac{1}{2}}/ft^2$; and $(453.6)^{-1}$ converts g to lb.

After the heat fluxes have been computed, the results of the calculations are printed out. To this end, the results are first loaded into the array SCØUT, to allow use of a relatively compact output format. If a binary output tape is being prepared during the run (DATAPE = .TRUE.), certain of the model condition results are also written on tape unit ITPØUT.

3.31.2 Normal Shock Solution

Figure 15 diagrams the method used in the normal shock solution. This technique is explained in analytical terms in Section 8.1.1 of Volume I. The solution is based upon an iteration to determine the temperature T2 behind the shock. The value of T2 obtained during the previous call to MØDEL is used as a first estimate; this value is stored as T2E (equilibrium shock) or T2F (frozen shock) in common block /STAG/. Prior to the first call to MØDEL, these values are initialized to the reservoir temperature.

The criterion for selection of T2 is that the static enthalpy behind the shock, calculated from T2 using equation

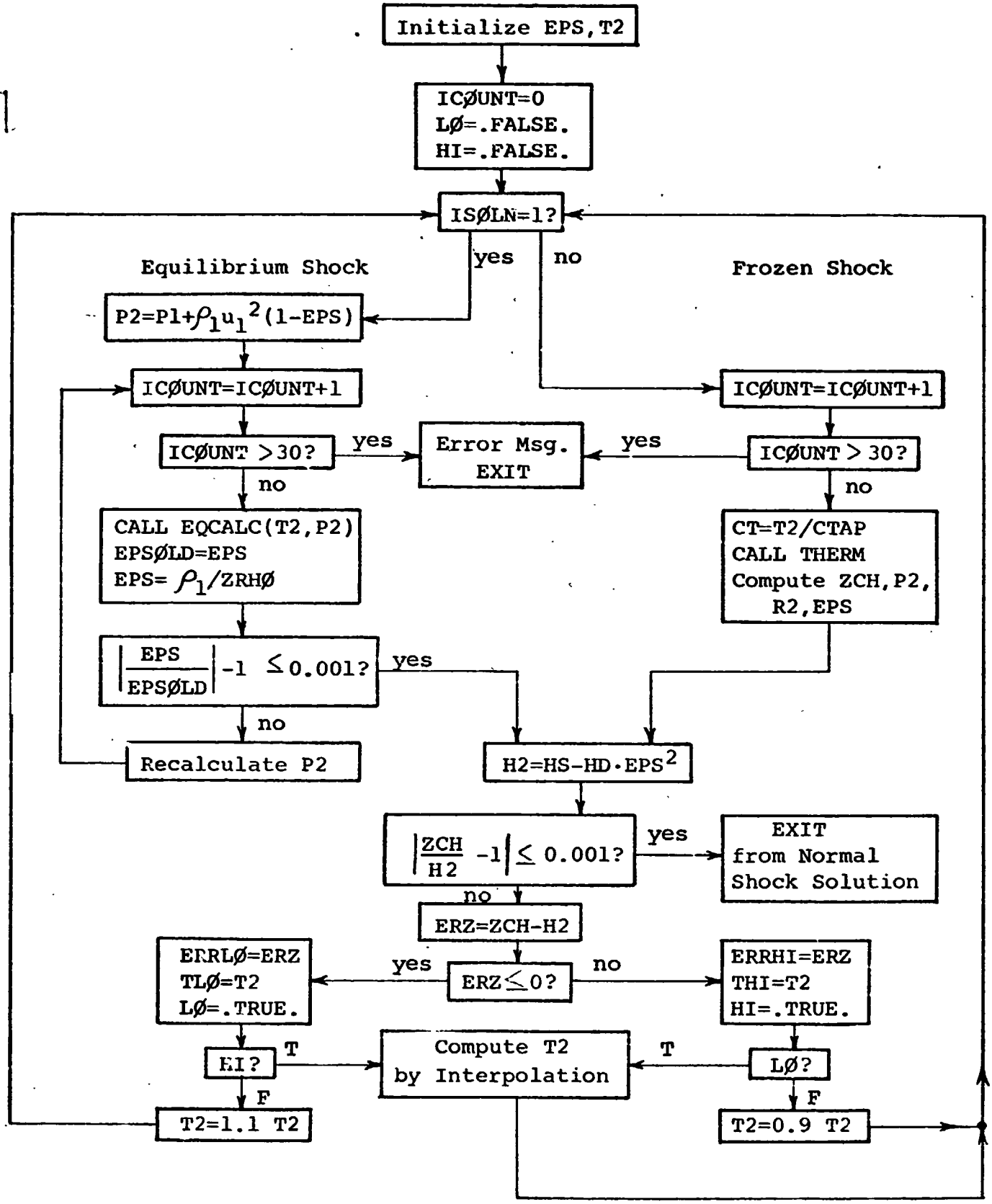


Figure 15. Flowchart of the Normal Shock Solution

I(429b), be equal to the value H2 obtained from the Rankine-Hugoniot energy equation for the shock, equation I(432b). In the case of a frozen shock, the gas is thermally perfect because the mole fractions are constant. The static enthalpy ZCH is calculated, in this case, from species enthalpies obtained by calling subroutine THERM after the nondimensional temperature CT has been set to the value T2/CTAP. The pressure P2 behind the shock is then computed from I(437), and the shock density ratio from I(434).

The equilibrium shock solution is more complicated because in this case the density ratio ϵ is not given as an explicit function of T2. Thus, a double iteration is required. For each trial value of T2, ϵ and the pressure P2 are calculated by an inner iteration as described in the discussion around equation I(438). In this case, the static enthalpy ZCH and density ZRHØ behind the shock are obtained by calling subroutine EQCALC. The outer iteration to determine T2 is the same as for the case of a frozen shock.

The object of the outer iteration is to make ERZ = ZCH - H2 equal to zero, to within an accuracy of 0.1 percent. The temperature T2 satisfying this criterion is obtained by iterative linear interpolation:

$$(T2)_{new} = (T2)_{old} - ERZ \cdot \frac{THI-TL\emptyset}{ERRHI-ERRL\emptyset} \qquad (114)$$

Here $(T2)_{new}$ is the improved estimate resulting from the linear iterpolation, $(T2)_{old}$ is the previous estimate, ERZ is the enthalpy error obtained at $(T2)_{old}$, ERRHI is the last positive error ERZ obtained and THI the corresponding T2 estimate, and ERRLØ is the last negative error obtained with TLØ the corresponding temperature. Of course, this formula cannot be used unless points with both positive and negative ERZ have been computed. This requirement is implemented with the aid of logical variables LØ and HI which are initialized to .FALSE.. LØ is set to .TRUE. when a negative ERZ is obtained, and HI is set to .TRUE. when a positive ERZ is found. If ERZ is negative and HI is "false", T2 is increased by 10 percent. If ERZ is positive and LØ is "false", T2 is

decreased by 10 percent. The interpolation formula (114) is not used until both LØ and HI are .TRUE..

3.31.3 Conditions at the Inviscid Stagnation Point

After the normal shock calculation has converged, the conditions at the inviscid stagnation point are computed as explained in Section 8.1.1 of Volume I. A flowchart of the coding is given in figure 16. The logic is similar to that of the normal shock solution (figure 15), but with two simplifications:

(1) The stagnation pressure is calculated from the incompressible Bernoulli equation (428b) with no iteration. Thus, only a single iterative loop is required even for the case of equilibrium flow.

(2) The initial estimate of the stagnation temperature TS is taken to be the temperature T2 just behind the normal shock. Since T2 is known to be lower than TS, the logical variable LØ is not needed in the stagnation-condition solution. Linear interpolation is begun as soon as a positive ERZ is obtained, where ERZ is the difference between the known stagnation enthalpy HS and the local enthalpy at the stagnation point (ZCH) computed from TS and PS.

3.32 Subroutine NEWRAP

The function of subroutine NEWRAP is to compute the flow conditions at a point in the equilibrium flow solution, using the method explained in Section 6.2 of Volume I. First, the system of (c + 1) equations I(227) and I(252) is solved for the species mole fractions X_i and the pressure p. Then the remaining flow variables are computed from these data.

The Fortran notation in NEWRAP is similar to that in EQCALC, already discussed in Section 3.18, but there are some differences and additions. ISMCNR denotes the number of dependent species. Initially, this is equal to n-c=ISS-ISC.

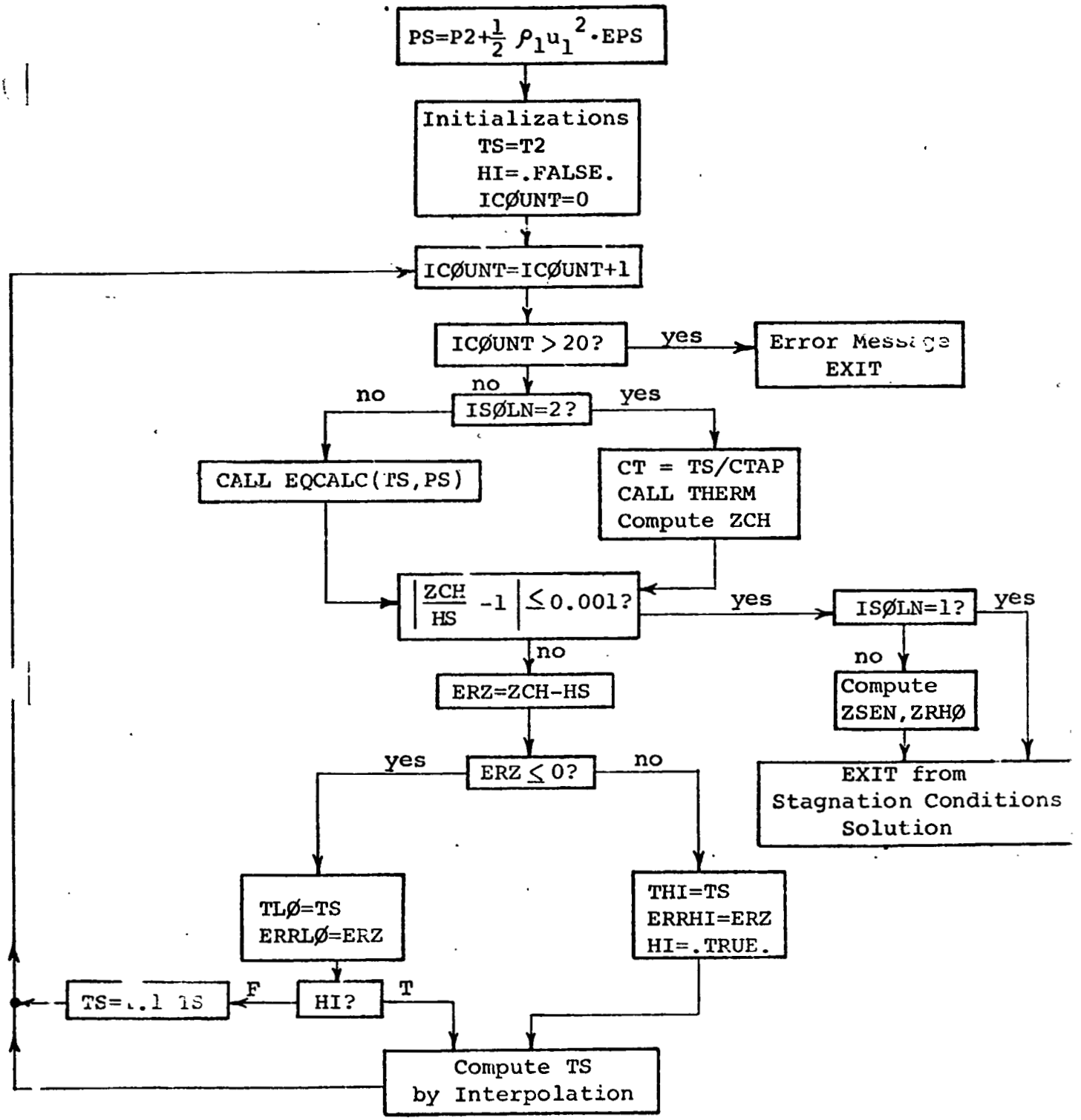


Figure 16. Flowchart of the Solution for Conditions at the Inviscid Stagnation Point

The array CHII is redefined as the logarithm of the equilibrium constants I(225) for the dependent species:

$$\text{CHII(i-c)} = -\frac{\mu_i^o}{R_0 T} + \sum_{j=i_m}^{c} \bar{\nu}_{i-c,j} \frac{\mu_j^o}{R_0 T} \qquad (115)$$

where i_m = IM is the index of the first independent species, initially equal to 1. A new array PGJ is defined by

$$\text{PGJ(i)} = \frac{S_i^o}{R_0} - \text{CRRB} \cdot W_i \qquad (116)$$

where S_i^o denotes the molar entropy of the i^{th} species at the standard pressure, W_i is the molecular weight of the i^{th} species, and*

$$\text{CRRB} = \frac{S_0}{R_0 W_0} \qquad (117)$$

where S_0/R_0 is the nondimensional molar entropy of the gas in the reservoir and W_0 the reservoir molecular weight.

Immediately upon entry, NEWRAP calls subroutine THERM to compute the species thermal properties at the temperature CT·T_0 corresponding to the current value of the nondimensional temperature CT. NEWRAP then sets up the arrays CHII and PGJ, defined above, and obtains a set of initial estimates of the species mole fractions X_i = CAPX(I) from equation I(1):

$$\text{CAPX(i)} = W \gamma_i \qquad (118)$$

The molecular weight W = CM and species concentrations γ_i = GJ(i) are the values in storage. For the first call to NEWRAP, these are reservoir values. In subsequent calls, they are the final values for the preceding flow point. Also,

*CRRB is set in subroutine INGAS following the entry INTA.

SKIL(I) is initialized to CAPX(I), and ZP to ℓn(PRES), where PRES is the stored value of the nondimensional pressure.

The Newton-Raphson iterative loop starts at statement 70. The system of linear equations to be solved in each step of the iteration is I(258) . The constant terms $F_j(Y_k^r)$ are loaded into the $M2^{th}$ column of the matrix AA, where M2 = c + 2. For j = 1 to c, these terms are given by equation I(227), and are computed in the DØ loops ending at statement 90. The constant term for j = c + 1 is given by I(252), and is loaded into AA(M1,M2) by the statements between 90 and 100. In the loop "DØ 100", the upper limit ISSNR is the current number of species included, initially equal to ISS.

The matrix of coefficients for the system of equations I(258) is loaded into the first (c + 1) rows and columns of AA. The coefficients I(233b) for j = i_m to c and k = i_m to c go into the first c rows and columns, and are set by the nested DØ loops ending with statement 150. The coefficients I(261b) for j=c + 1 and k = i_m to c are computed in the DØ loops ending at statement 170. The coefficient I(261c) for j = c + 1 and k = c + 1 is set up by the loop DØ 180. Finally, the coefficients I(261a) for j = i_m to c and k = c + 1 are loaded into AA by the DØ loops ending at statement 210.

If the number of dependent species ISMCNR is zero a logical indicator NØDEPS is set to .TRUE. at the beginning of NEWRAP. Tests on NØDEPS then cause the subroutine to skip all of the DØ loops with DØ indices running from 1 to ISMCNR, in which sums from c + 1 to n are evaluated.

During an equilibrium expansion to high Mach numbers, the mole fractions of ionized species can become extremely low as the gas temperature falls to a small fraction of its value in the reservoir. To avoid computational problems arising when the electron mole fraction becomes exceedingly small, the electrons and all ionized species are eliminated from the equilibrium calculation whenever the electron mole fraction drops below 10^{-20}. The test for this condition is in statement 220. If CAPX(1) is found to be $\leq 10^{-20}$, the following

actions are taken:

(1) The indicator JJK is reset from its initial value of 0 to 1. The latter value signifies that the elimination of charged species has already been performed.

(2) The number of ions (IC) is subtracted from ISMCNR and ISSNR.

(3) IM is increased from 1 to 2.

(4) AA(1,1) is set to 1, and all the other entries in the first row and first column of AA are set to 0.

(5) The electron mole fraction, CAPX(1), is set to 0.

(6) The mole fractions of all the ion species are set to 0.

Regardless of whether the charged species have been eliminated, the system of equations I(258) is solved by calling subroutine DSMSØL. Improved values for the species mole fractions and the pressure are then obtained using equations I(262). The correction factor $1 + h_n^r$ in I(262) is denoted by ZB for n = 1 to c (i.e., for the mole fractions), and by ZC for n = c + 1 (the pressure). If any of these correction factors turns out to be 0 or negative, the corresponding quantity (mole fraction or pressure) is adjusted by dividing it by 2. Based on the improved values of p and the X_i, the other quantities in the calculation ($ZP = \ell n\, p/p_0$ and $CGMU(K) = \ell n\, X_k$), are then recomputed. Next, the mole fractions for the dependent species are computed from the equilibrium relations I(224), which are written in the logarithmic form

$$\ell n\, X_i = \ell n\, K_{pi} + (\nu^*_{i-c} - 1)\, \ell n\, p + \sum_{\ell=1}^{c} \bar{\nu}_{i-c,\ell} X_\ell \quad (119)$$

The term $\ell n\ K_{pi}$ is denoted in NEWRAP by CHII(i), equation (115). The $\ell n\ X_i$ are computed in the array SKIL(i-c) and the resulting values are saved in CGMU(i). Then the X_i themselves are computed by taking exponentials and are stored in SKIL(i-c) and in CAPX(i).

The convergence criterion for the iteration is that the absolute magnitudes of all of the relative corrections h_n^r be smaller than or equal to TEST = 10^{-6}. When convergence has been achieved, the other flow variables are computed from the pressure, the mole fractions, and the known reservoir conditions. The relations used in these calculations are equations I(236), I(237), and I(242) for the molecular weight, density, and specific enthalpy, equations I(79b) and I(81) for the effects of gas imperfections, equation I(263) for the flow velocity, and equation I(264) for the mass flux. As noted in Section 3.22.1, the flow variables are computed in nondimensional form.

3.33 Subroutine NEXTMP

The function of subroutine NEXTMP is to determine the location (XMØDEL) of the next point at which model condition calculations are to be done. In the case of channel flow solutions, the routine determines the location corresponding to the next specified channel width.

The first two arguments of NEXTMP specify which of the model points called for in the input have already been passed in the current solution. ITS is the index of the next of the specified test-section diameters or channel widths, while XMØD1 is the x-coordinate the next model point in the geometric sequence of model points.

In the case of channel flow (NPRFLS = 2), the geometric sequence is not used. NEXTMP simply calls subroutine FINDXC to determine the position XMØD2 at which the width of the wide (MBLth) profile has the specified value TCAR(ITS), and sets XMØDEL equal to XMØD2.

In the case of nozzle flow (NPRFLS = 1), the next model point may be either the next point (XMØD1) in the geometric sequence or the point with the next specified area

ratio. In this case, NEXTMP calls FINDX to determine the location (XMØD2) of the point with the area ratio TSAR(ITS), and sets XMØDEL to the lower of the two values XMØD1, XMØD2. If the next point turns out to be at the specified area ratio, ITS is incremented by 1 to signify that TSAR(ITS) has been used. If the next point is a point in the geometric sequence, XMØD1 is replaced by FACMP times XMØD1, which is the next subsequent value of the sequence.

3.34 Subroutine NØNEQ

NØNEQ is the controlling routine for the nonequilibrium flow solution. The overall method of solution has been explained in Section 7.2 of Volume I. Briefly, the solution is started upstream of the throat, near the reservoir, using the perturbation method. At each step in the perturbation solution, the quantities $\delta\gamma_i$ (which are measures of the departures of the reactions from equilibrium) are calculated and tested. When any one of these quantities reaches a specified size, the perturbation technique is abandoned and the solution is continued by numerical integration. If the integration is begun well downstream of the throat, it is carried out using the normal, direct method in which the specified nozzle geometry is taken to be a condition on the flow. However, if the integration is started upstream of the throat, an inverse method is used, in which the equilibrium density distribution $\rho(x)$ is used to define the flow. In this case, a switch is made from the inverse method to the direct method when the solution reaches a point slightly beyond the throat.

Figure 17 is an overall flowchart of subroutine NØNEQ. This flowchart shows how the various methods are organized to produce the entire solution. The operations enclosed in double boxes are represented by separate flowcharts and discussed below. IUPD is an indicator which is initialized to 1 (in subroutine READ), and is reset to 0 (in subroutine THRØAT) at the switch from the upstream to the downstream region. INEQ is another indicator, initialized to 0, and set to 1 when the numerical integration is started. Figure 17 does not show any exits from the control loops that generate the solution, because the case termination controls are within the boxes labelled "Numerical integration" and "Perturbation solution". The statements involving MØDLPT, XMSET, and subroutine NEXTMP pertain to the calculation of conditions at model points, and are discussed below.

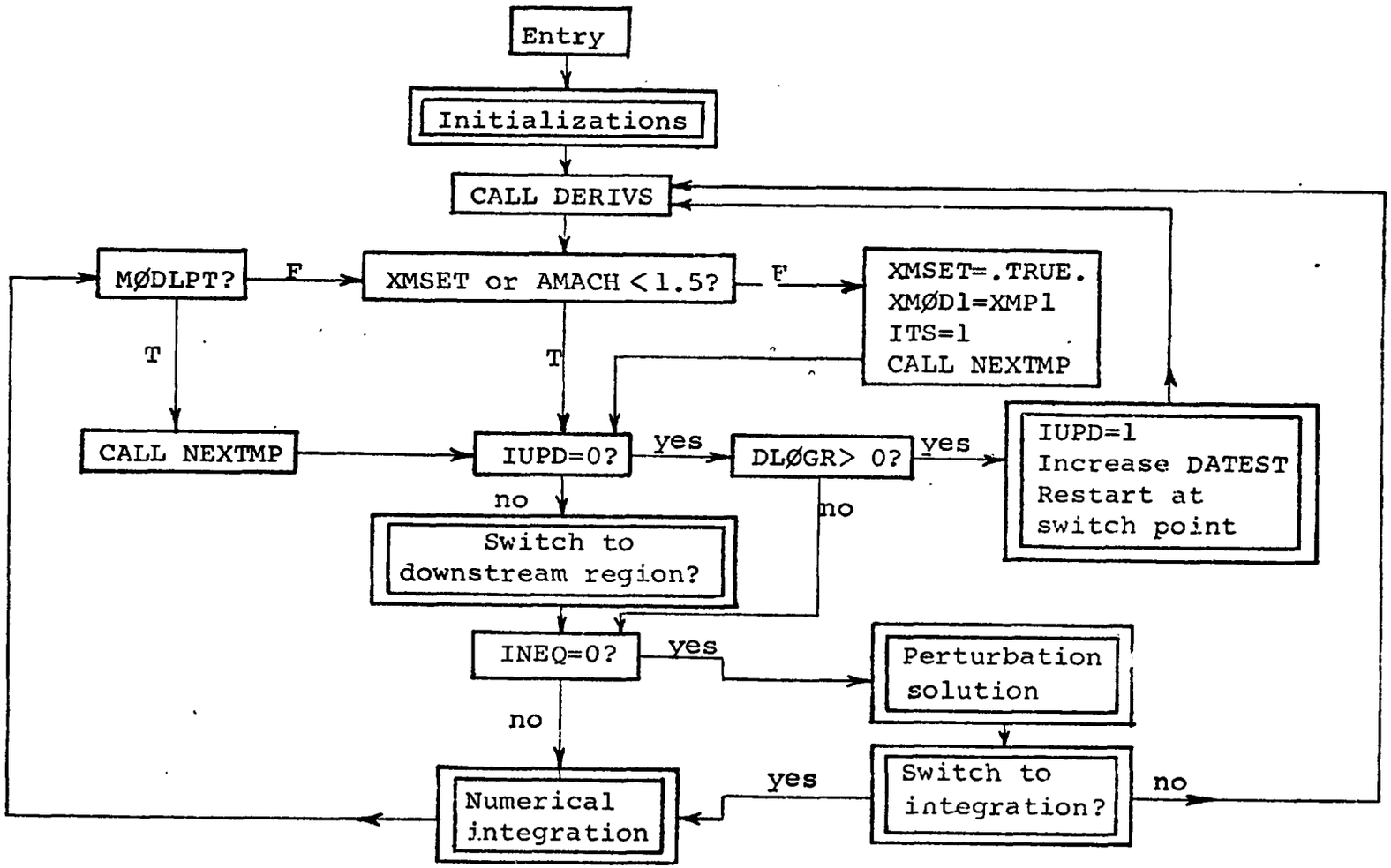


Figure 17. Flowchart of Subroutine NØNEQ

The remaining parts of this section discuss individual portions and aspects of subroutine NØNEQ in detail.

3.34.1 Initializations

Figure 18 is a flowchart of the initializations section of NØNEQ, represented by the uppermost double box in figure 17. This flowchart shows only the initializations of control variables. In the subroutine, several physical variables are also set, but their values are not needed for following the logic.

The perturbation technique, which is used to start the solution, proceeds by taking steps in temperature, ΔT. The initial temperature decrement is given by $\Delta T/T_0$ = DELT1, where the variable DELT1 is under input control and has a preset value of 0.01. This preset value of DELT1 gives fairly large steps in the perturbation solution. In cases including the boundary layer(ISW3B $\neq$ 0), to allow more accurate evaluation of the boundary layer integral I(172), DELT1 is reset to 0.049 (1 - CTMAX), where CTMAX is the nondimensional sonic temperature T^*/T_0 based on the equilibrium solution. This value gives 20 steps upstream of the throat if the perturbation technique is used all the way to the throat. If the numerical integration is started upstream of the throat,as is usually the case,the number of steps is larger because the initial step size in the integration is much smaller than the initial perturbation step size.

The temperature, species concentrations, and gas properties are initialized for the nonequilibrium solution in the main program. Before calling NØNEQ, the main program sets the nondimensional temperature CT equal to 1. - DELT, where DELT is set by the same rules used in calculating DELT1 in NØNEQ. MAIN then calls NEWRAP to determine the equilibrium flow conditions at this temperature, calculates the effective area ratio, and determines the corresponding position in the nozzle by calling FINDX.

As explained in Section 7.3 of Volume I, the system of equations used in the perturbation technique is solvable only if the β_{ij} matrix has a rank of n-c. The initializations section of NØNEQ computes the rank IX of BETA(I,J) and executes an error exit if IX is not equal to n-c.

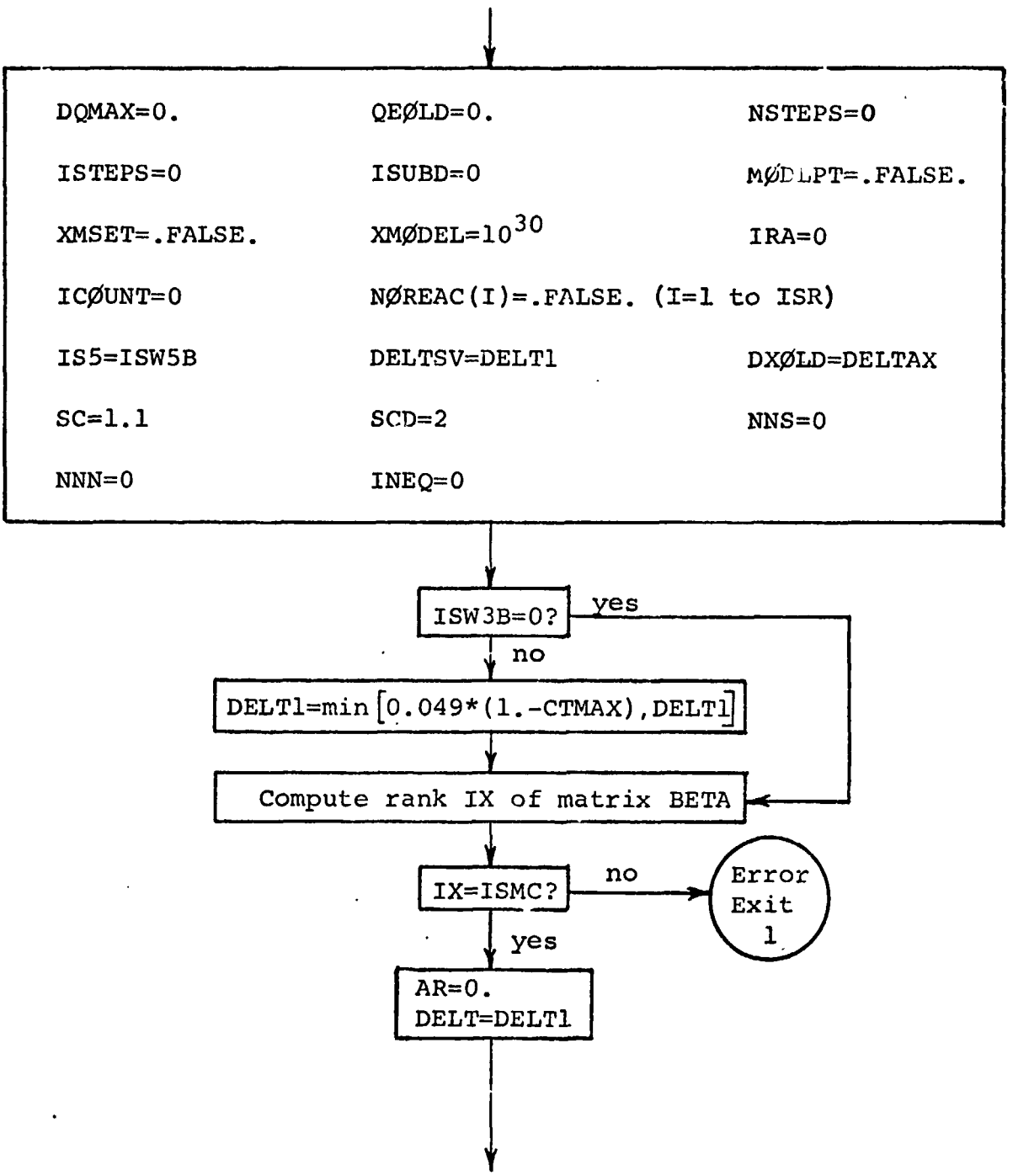


Figure 18. Flowchart of Initializations Section of Subroutine NØNEQ

3.34.2 Switch to Downstream Region

Figure 19 is a flowchart of the operations represented by the box labelled "Switch to downstream region?" in figure 17. The coding shown in figure 19 determines whether the conditions for switching to the downstream solution have been satisfied, and if so, carries out the switch by calling subroutine THRØAT. The conditions are as follows:

(1) DLØGA = $d \ln \tilde{A}/dx$, where $\tilde{A}$ is the area ratio computed using the inverse method, must be nonnegative.

(2) The position of the flow point must lie beyond the geometric throat (CX > 0).

(3) The area ratio $\tilde{A}$ = AFNX must be greater than or equal to a value DATEST which is preset (in subroutine INIT) to 1.01.

When all of these conditions have been satisfied, a large number of flow variable and control parameters are saved in the arrays TB, ITB, and BLBK, and the switch to the downstream region is executed by calling THRØAT. In THRØAT, IUPD is reset to 0 and RSA is set to a value which rescales the actual nozzle geometry to make it consistent with $\tilde{A}$.

In cases including the boundary layer, the coding in figure 19 also sets the ØMDST(I) array based on the displacement thickness DELBL(I) at the first point where the conditions (1) and (2) are satisfied.

3.34.3 Restart of the Upstream Solution

Figure 20 is a flowchart of the operations represented by the box labelled "IUPD = 1, Increase DATEST, Restart at Switch Point" in figure 17. This section of NØNEQ is entered when a positive value of $d \ln \rho/dx$ = DLØGR is encountered in the downstream solution. If this condition arises so far beyond the upstream/downstream switch point that the effective area ratio A_e = AFNX exceeds DATEST by 5 percent of DATEST or more, an error exit is executed. Otherwise,

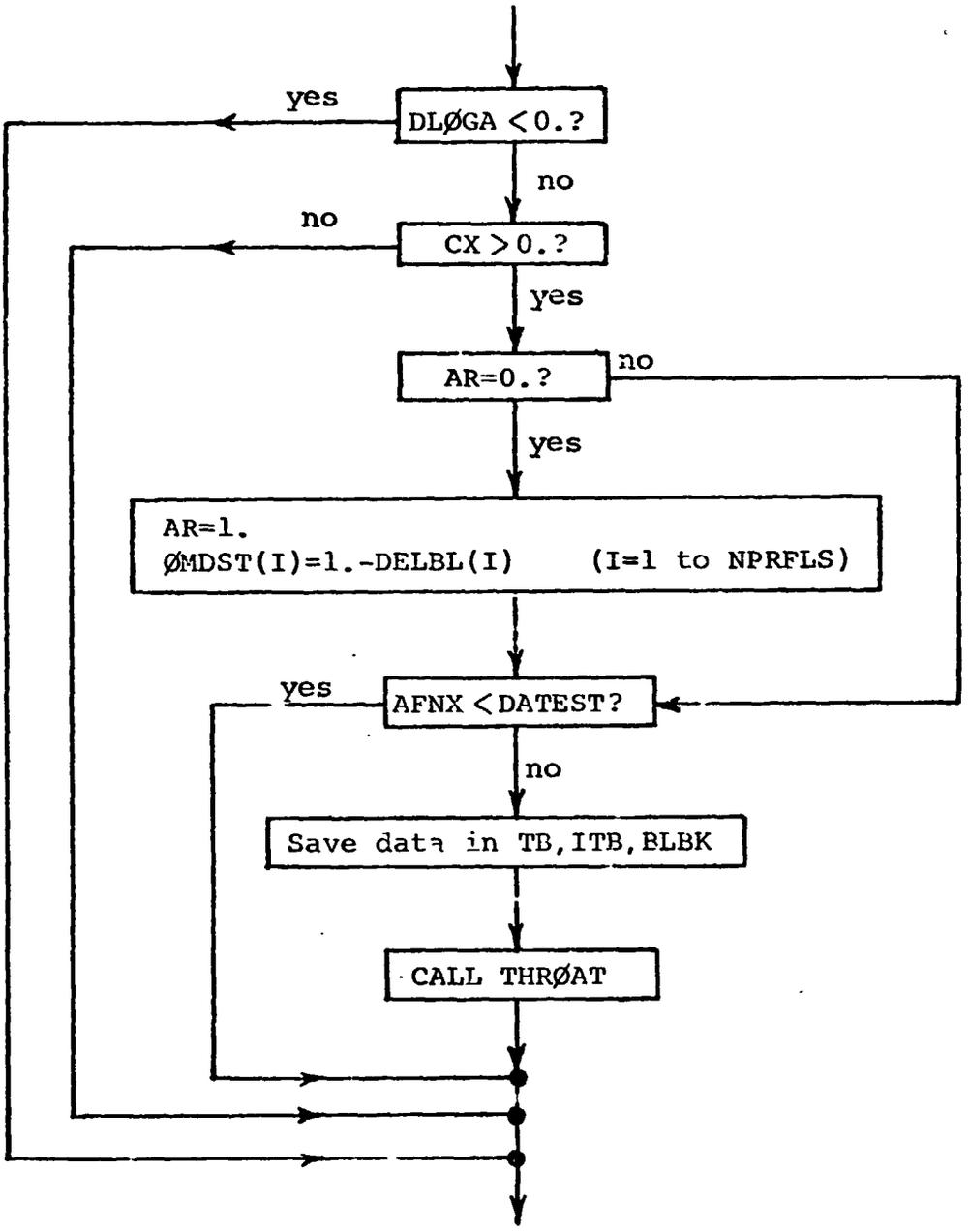


Figure 19. Flowchart of Switch to Downstream Region in Subroutine NØNEQ

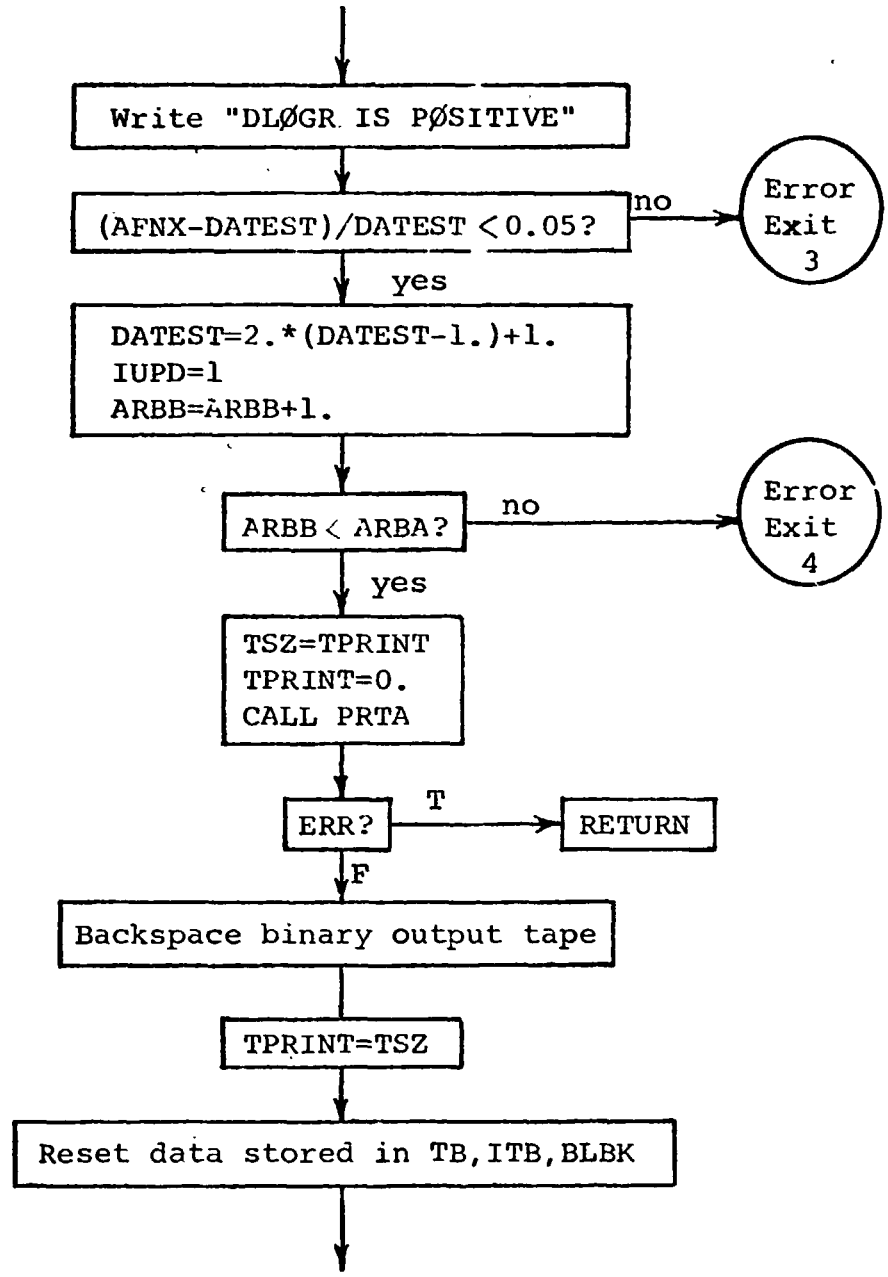


Figure 20. Flowchart of Restart of the Upstream Solution at the Switch Point in Subroutine NØNEQ

the program assumes that the positive $d\ln\rho/dx$ value is a result of switching to the downstream solution too near the throat, and attempts to recover the correct solution. To this end, the excess of DATEST over unity is doubled, IUPD is reset to 1 (signifying the upstream solution), the flow variables and control parameters are all reset to their values at the previous switch point, and the solution is restarted at that point. A counter ARBB (initialized to 0. in subroutine INIT) is also incremented by 1. If the $d\ln\rho/dx > 0$ condition is encountered repeatedly after successive restarts, the effort to continue the solution is abandoned after four attempts (ARBB $\geq$ ARBA = 5., set in INIT). Also, before the restart is executed, the conditions at the final point of the invalid downstream solution are printed out by setting TPRINT = 0. and calling PRTA. If a binary output tape is being produced for subsequent use by the NATA plot program, the records containing data from the invalid solution beyond the switch point are eliminated by backspacing the tape.

3.34.4 Perturbation Solution and Switch to Integration

Figure 21 is a flowchart of the operations represented by the boxes labelled "Perturbation solution" and "Switch to integration?" in figure 17. These operations may be summarized as follows:

(1) Subroutine PERT is called to compute the perturbations PCT (in the nondimensional temperature), PERTGJ(J) (in the species concentrations), and SDCHI(I) (in the reaction parameters χ_i).

(2) If the solution is in the downstream region (IUPD = 0) and the nondimensional temperature decrement DELT is still equal to the reduced value of DELT1 (see part 3.34.1 above), DELT is reset to the input value of DELT1, which was saved in DELTSV. This is done because the step size in the perturbation solution is sufficiently small in the throat and near downstream region, even when the larger decrement is used.

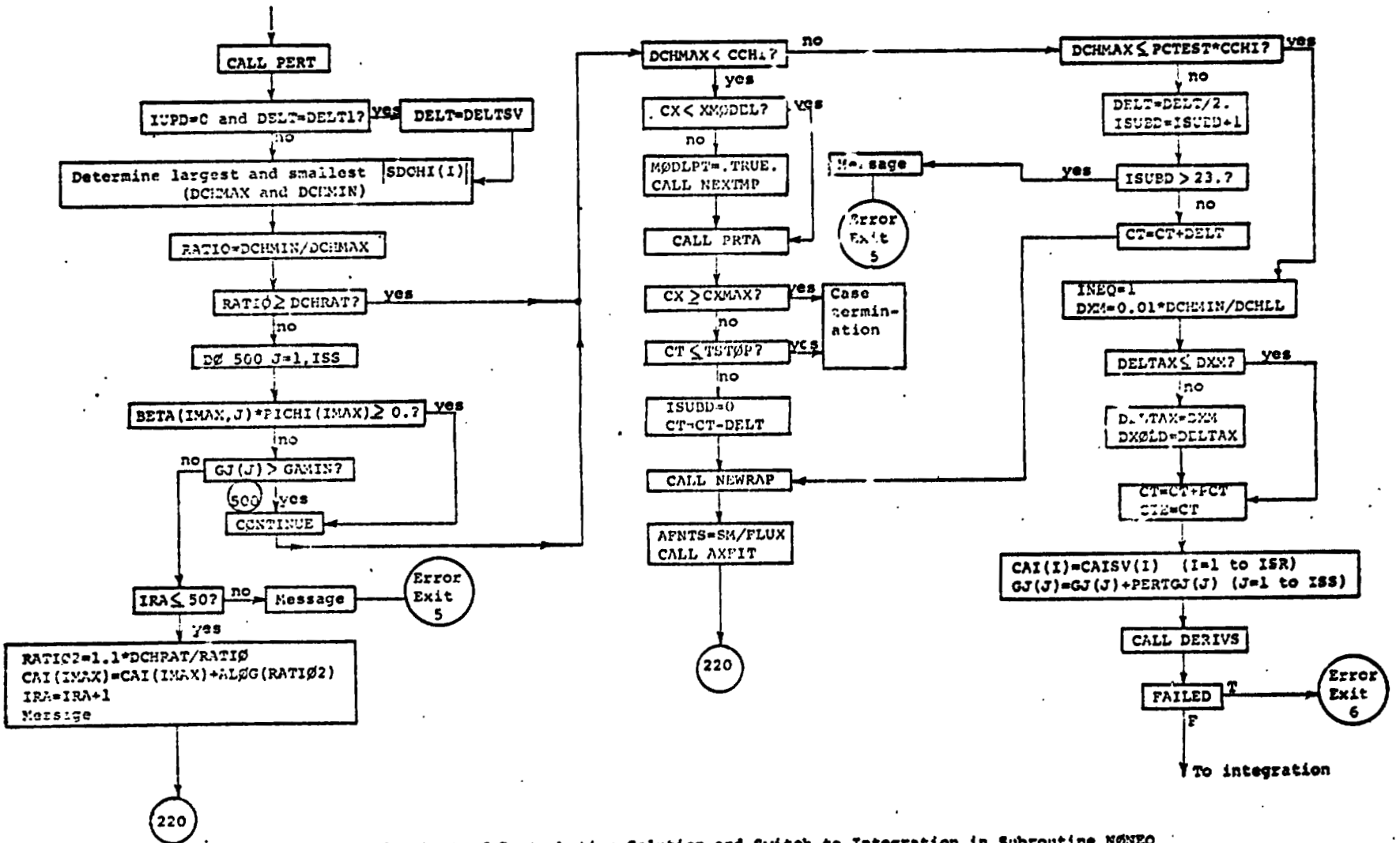


Figure 21. Flowchart of Perturbation Solution and Switch to Integration in Subroutine NØNEQ

(3) The largest SDCHI(I) value (DCHMAX) and the smallest (DCHMIN) are determined.

(4) The ratio RATIØ = DCHMIN/DCHMAX is examined. If this ratio is larger than or equal to a value DCHRAT (preset in BLØCK DATA to 10^{-4}), NØNEQ proceeds to the checks for the switch to numerical integration, item (5) below. If RATIØ is less than DCHRAT, the reaction system assumed in the gas model will lead to a switch to integration at a point where some of the reactions are still very near equilibrium. The value of RATIØ is approximately indendent of the size of DCHMAX; thus, the program responds to an excessively small value of RATIØ immediately, even though DCHMAX may be much smaller or much larger than the value CCHI at which the switch to integration is desired. To prevent the premature start of the numerical integration, NATA artificially increases the rate constant for the reaction (I = IMAX) for which |SDCHI(I)|= DCHMAX, if this can be done without significantly affecting the solution. This procedure is discussed near the end of Section 7.3 in Volume I. If RATIØ < DCHRAT, NØNEQ determines whether this reaction involves a minor species (concentration less than or equal to GAMIN, preset to 10^{-10}) whose concentration is being decreased by the reaction. If so, the rate constant for the IMAXth reaction is increased by a factor of RATIØ2 = 1.1 *DCHRAT/RATIØ, and the perturbation calculation for the previously assumed temperature is repeated. If not, the perturbation solution is continued without adjusting the rate constant.

(5) The next step is to check whether DCHMAX has reached the value CCHI at which the switch to the numerical integration is desired. If DCHMAX is less than CCHI, the perturbation solution continues. If the axial coordinate

CX is greater than or equal to the current value of XMØDEL (the coordinate of the next model point), the indicator MØDLPT is set to .TRUE., and subroutine NEXTMP is called to determine a new value for XMØDEL. In any event, subroutine PRTA is called to provide output of the computed flow conditions at the current point of the perturbation solution. If MØDLPT is .TRUE., PRTA also calls subroutine MØDEL to calculate test conditions on a model.

(6) After the return from PRTA, the case termination tests are applied. If CX is greater than or equal to CXMAX, or if the nondimensional temperature CT is less than or equal to TSTØP, control is transferred to statement 660. The flow conditions and model conditions at the final point are then printed and the RETURN is executed.

(7) If case termination has not been reached, the temperature CT is decremented by DELT, and subroutine NEWRAP is called to compute the equilibrium flow conditions at the new temperature. The equilibrium effective area ratio AFNTS is then calculated from the continuity equation I(265), and subroutine AXFIT is called to determine the axial coordinate CX at this area ratio. Control is then transferred to statement 220 ("CALL DERIVS" in figure 17).

(8) Once DCHMAX reaches or exceeds the criterion value CCHI for the switch to integration in step (5), NØNEQ commences an iterative adjustment of the nondimensional temperature CT in order to obtain a flow point at which DCHMAX= $|\delta\chi_i|_{max}$ satisfies both of the conditions in equation I(381). Usually the perturbation solution overshoots the second condition I(381), i.e., DCHMAX is greater than PCTEST*CCHI,where

PCTEST = 1.2 (set in subroutine INIT). When this occurs, NØNEQ cuts DELT in half, adds DELT to CT in order to obtain a temperature halfway between the current value and the preceding one, and computes the corresponding point in the perturbation flow solution by calling NEWRAP and AXFIT as in step (7) above. This procedure, together with the normal procedure for advancing the perturbation solution (step (7)), repeatedly subdivides the temperature interval known to contain the flow points satisfying the conditions I(381).

(9) When a point satisfying both of the conditions I(381) has been obtained, the switch from the perturbation technique to numerical integration is carried out. The indicator INEQ is reset from 0 to 1 and the perturbations PCT, PERTGJ(J) and PRHØ are added to the corresponding equilibrium flow quantities CT, GJ(J), and RHØ. The integration step size DELTAX is adjusted, if necessary, to make it no larger than DXM = 0.01* DCHMIN/DCHLL, where the input DCHLL is preset to 10^{-4} in BLØCK DATA. This adjustment provides a reduced initial step size when DCHMIN = $|\delta\chi_i|_{min}$ at the switch point is smaller than 10^{-4}, i.e., when some of the reactions are still very close to equilibrium. If this reduction in DELTAX were omitted, the normal step size controls of the integration would still bring DELTAX down to similar values at the cost of some additional computation. Any rate constants which have been artificially increased in step (4) are now restored to their correct values. Subroutine DERIVS is called to compute the derivatives and supplementary flow variables at the switch point, based on the perturbed temperature and concentrations, and control is transferred to the numerical integration section of NØNEQ. If the indicator FAILED is .TRUE. after the call to DERIVS, an error exit is executed. This indicator is .TRUE. if the flow conditions fail to pass certain validity checks in DERIVS and CØMM.

3.34.5 Numerical Integration

Figure 22 is a flowchart of the operations represented by the box labelled "Numerical integration" in figure 17. These operations may be summarized as follows:

(1) The case-termination checks are the same as in the perturbation solution. If CX $\geq$ CXMAX or CT $\leq$ TSTØP, control is transferred to statement 660. The flow conditions and model conditions at the final flow point are then printed, and the RETURN is executed.

(2) If the end of the case has not been reached, the counter NNN is incremented and subroutine PRTA is called to provide output of the conditions at the flow point reached during the preceding integration step.

(3) Next, the integration step size DELTAX and the factors SC and SCD used in changing DELTAX are adjusted. If NNN is equal to NQS (an input parameter, preset to 4), then DELTAX is multiplied by SC, NNS is incremented by 1, and NNN is reset to 0. Then if NNS is equal to NQS, SC is increased by 0.1, SCD is reset to the new SC value, and NNS is reset to zero. For NNN < NQS, no changes are made in these variables.

(4) In preparation for the next integration step, the current values for CX, CT, GJ(I), and other quantities are saved in the storage locations CXB, CTB, GJB(I), etc. If the forthcoming step should fail for any reason, these data will be used in restarting the solution at the beginning of the step with a smaller step size.

(5) Next, the anticipated position, CX + DELTAX, at the end of the forthcoming step is compared with XMØDEL and CXMAX. If CX + DELTAX is greater than either of these values, DELTAX is reduced to make the end of the step coincide with the lesser of

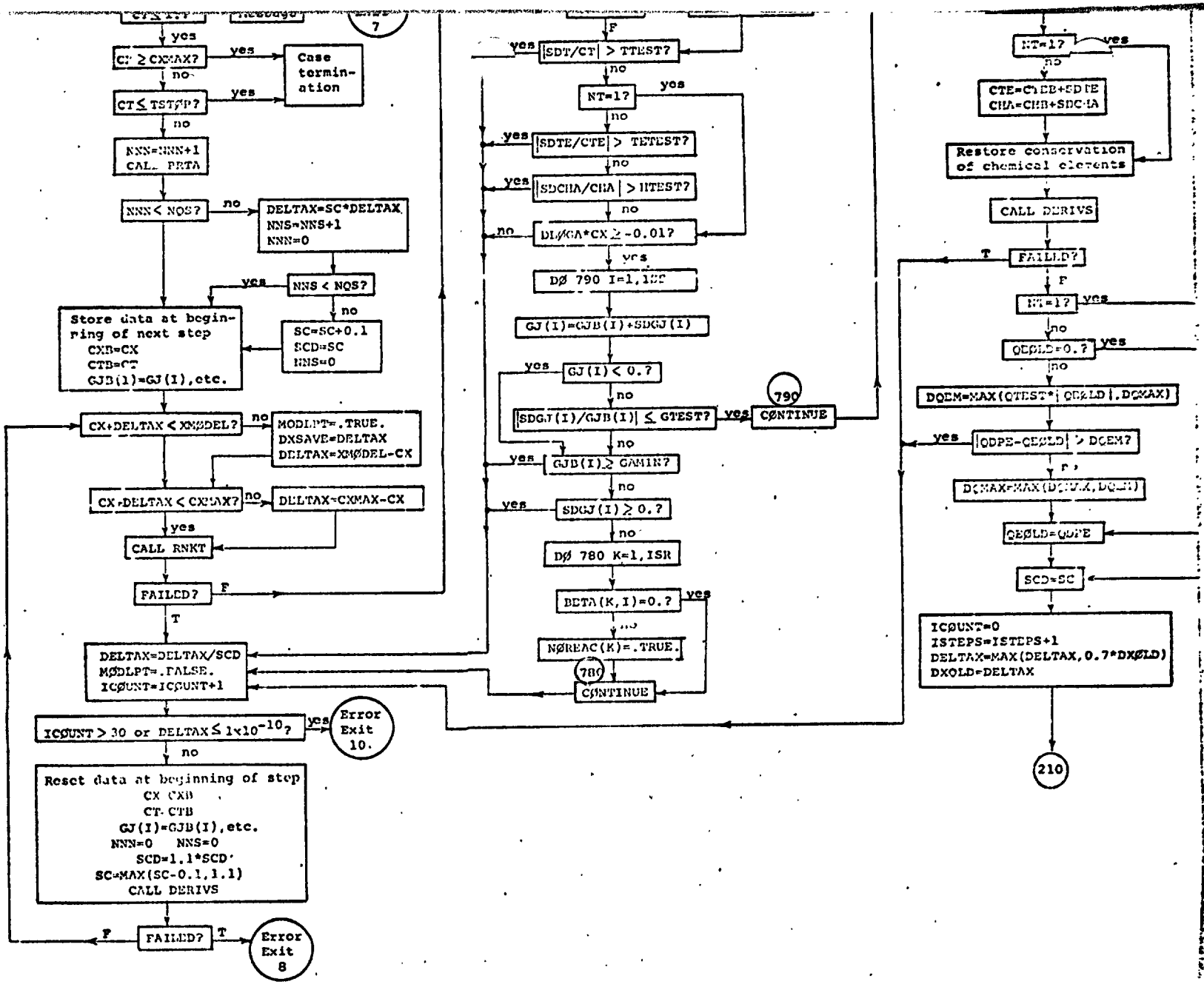


Figure 22. Flowchart of Controls for the Numerical Intogration in Subroutine NONEO

the two. If the end of the step will be at XMØDEL, then MØDLPT is set to .TRUE. and the DELTAX value before the reduction is saved in DXSAVE.

(6) Subroutine RNKT is called to compute the changes in the dependent variables over the integration step.

(7) The indicator FAILED is tested. A .TRUE. value of FAILED indicates that a validity check was not satisfied somewhere in RNKT, DERIVS, or CØMM. If FAILED = .TRUE., the step size is reduced by dividing DELTAX by SCD and the calculation is restarted at the beginning of the step, using the data saved in step (4). In the event of such a step restart, a counter ICØUNT is incremented. Also, MØDLPT is set to .FALSE., SCD is increased by a factor of 1.1, and SC is reduced by 0.1 (but not below a value of 1.1). In addition, NNN and NNS are both set to zero. The idea underlying these changes in the step control parameters is that the failure was caused by instability resulting from the use of an excessively large step size. Either the step has been increased too rapidly, or the solution is entering a region where the reaction system requires a smaller step size for stability. Thus, the step size itself is reduced, and the growth in step size is both temporarily arrested and reduced in rate. The increase in SCD allows the program to reduce the step size very rapidly if repeated failures are encountered in the same step, as sometimes happens in runs using electronic nonequilibrium models.

(8) If FAILED is .FALSE. after the call to RNKT, additional validity checks I(407) and I(408) are applied to the results of the integration step. If any of these conditions is violated, the step is restarted with a smaller step size as described above in item (7). Also an indicator IFAIL is set to a distinctive value for each possible type

of failure. IFAIL is also set when step failures occur in other subroutines. Each time a step fails and is restarted, the IFAIL value is saved in the integer array IF(ICØUNT). These values can be obtained in the printed output by setting ISW5B to a negative value in the code input. Output of these data shows what tests, species, etc., are controlling the integration step size.* If a failure occurs in one of the conditions I(407a), I(407c) on the species concentrations, then the program checks whether the procedure of freezing the concentration of a minor species (Section 7.5.3 of Volume I) is applicable. The conditions for use of this procedure are that the species I, whose concentration GJ(I) violates I(407a) or I(407c), should have a concentration GJB(I) at the start of the step less than GAMIN, and that the concentration change for this species over the step, SDGJ(I), be negative. If these conditions are satisfied, all of the reactions contributing to changes in the concentration of this species are frozen by setting the corresponding elements of the array NØREAC(K) to .TRUE.. The step is then restarted in the usual way.

(9) If the conditions I(407) and I(408a,b) are satisfied, the integration step is presumed to be valid subject to the condition I(408c), which is applied later. The changes SDT, SDTE, SDCHA in the dependent variables during the step are added to the values CTB, CTEB, CHB at the start of the step to obtain the corresponding values at the end of the step. The species concentrations are then adjusted to restore the original elemental composition of the gas mixture, as described in Section 7.1.3 of Volume I and in Section 3.34.6 below. Subroutine DERIVS is then called to compute the derivatives and supplementary flow variables at the end of the step.

---

*The numerical code used in IF(ICØUNT) is defined in Section 4.55.

(10) If NT = 2 (electronic nonequilibrium model), the condition I(408c) on the change in the energy transfer to the electron gas (QDPE or $\dot{q}_e$) is now applied to the final QDPE value resulting from this call to DERIVS. If this condition is not satisfied, the step is restarted with a reduced step size in the usual way.

(11) Once all of the validity checks have been passed, SCD is reset equal to SC and the step-restart counter ICØUNT is reinitialized to 0. The counter ISTEPS for successful steps is incremented by 1. The step size DELTAX is set equal to the larger of DELTAX itself and 0.7*DXØLD. Thus, if DELTAX has been reduced sharply by repeated step restarts, it is here returned to 70 percent of its value in the preceding step.

(12) Control is now transferred to statement 210 ("MØDLPT?" in figure 17).

3.34.6 Element Conservation

The adjustment of species concentrations to restore conservation of the chemical elements, explained in Section 7.1.3 of Volume I, is implemented by the coding from statement 800 down to 940 in NØNEQ. The programming is straightforward, and can easily be related to the formulas in Section 7.1.3 with the aid of the following summary of notations:

CJ	c_j'
DCI(J)	δc_j
GSQ(I)	$(\gamma_i')^2$
AIN(J,K)	A_{jk}
DCA(K)	$\frac{1}{(\gamma_k')^2} \sum_{j=1}^{c} \delta c_j \cdot A_{jk}$
AA(I,ISMCP1)	b_i (before call to DSMSØL)

CDIJ(I,K) $\bar{\nu}_{i-c,k}$

AA(I,L) $a_{i\ell}$

AA(I,ISMCP1) $\delta\gamma_{c+1}$ (after call to DSMSØL)

The system of equations I(346) is solved for $\delta\gamma_m$ by calling subroutine DSMSØL. After the adjusted concentrations of the dependent species (i = c + 1 to n) have been calculated from I(340) and those of the independent species (i = 1 to c) from I(338), the condition I(407a) is applied to the adjusted concentrations. If any of these concentrations is found to be negative, the step is restarted with a reduced step size in the usual way.

3.34.7 Model Points

The position of the next model point, at any point in the flow, is denoted by XMØDEL. Whenever a particular model point has been reached and the model calculations have been done, the new XMØDEL value is obtained by calling subroutine NEXTMP(ITS,XMØD1,XMØDEL). The argument ITS is the index of the next specified test section diameter, and XMØD1 is the coordinate of the next model point in the geometric sequence of model points. Subroutine NEXTMP, each time it is called, determines whether the next model point is at the test section diameter with index ITS or at XMØD1, sets XMØDEL accordingly, and updates ITS or XMØD1 as required. Before NEXTMP is called for the first time in NØNEQ, the initial values ITS = 1 and XMØD1 = XMP1 must be set. This is done by coding shown in figure 17 , at the first point where the Mach number AMACH has reached or exceeded a value of 1.5. A .TRUE. value of the indicator XMSET denotes that this initialization has already been performed.

The indicator MØDLPT, if .TRUE., denotes that the current flow point in the perturbation solution or the integration is a model point. If MØDLPT = .TRUE., PPTA calls entry ØUT2 of subroutine ØUT1 to print the flow conditions (even though the TPRINT criterion may not be satisfied) and calls MØDEL to compute and print the model test conditions.

When a model point is encountered during the numerical integration, DELTAX is reset to provide a flow point exactly at the model point. During the perturbation solution, MØDEL is called at the first solution point beyond XMØDEL. To obtain a perturbation solution point coinciding with XMØDEL would require an iteration, as in the equilibrium and frozen flow solutions, because the flow is generated by decrementing the temperature, not by incrementing the position coordinate. Such an iteration has not been provided in the coding because, in the normal applications of NATA, the switch from the perturbation technique to the numerical integration always occurs upstream of the point where the Mach number is equal to 1.5.

3.34.8 Error Exits

Subroutine NØNEQ contains ten error exits, marked conspicuously with comment cards in the listing. Whenever one of these exits is executed, a message giving the error exit number is written out before the DUMP routine is called to provide diagnostic data. This message facilitates determining the nature of the error that caused NØNEQ to fail.

3.35 Subroutine NRMAX

This subroutine determines the sonic flow conditions for the case of equilibrium flow. Its logic and structure are practically identical with those for the corresponding calculation for frozen flow, performed in subroutine FRØZEQ. The flowchart for the frozen calculation (figure 9) is applicable to subroutine NRMAX. The discussion in Section 3.22.2 is also applicable. In the present case, the mass flux at each temperature is determined by calling subroutine NEWRAP.

3.36 Subroutine ØUT

This routine simply prints a list of definitions of the labels used in the output of the flow conditions and test conditions on models and wedges.

3.37 Subroutine ØUT1

Subroutine ØUT1 is the principal output routine of NATA. When called through its first entry point, it produces a printed list of the conditions at a special point in the flow, such as the reservoir or the sonic point. When called through its entry ØUT2(ISØLN), it prints a more comprehensive specification of the conditions at a general flow point. In addition, if the indicator DATAPE is .TRUE., a call to ØUT2 causes a record to be written on the binary output tape (tape 8), for subsequent use by the NATA plot program. Figure 23 is an overall flowchart of the subroutine.

3.37.1 Entry ØUT1

ØUT1 is called (by MAIN) to print the conditions in the reservoir and at the frozen and equilibrium sonic points. Since a call to ØUT1 precedes each of the three flow solutions, ØUT1 is also used to initialize some quantities used by ØUT2.

The chemically frozen specific heat ratio GAMMA is computed in a section of coding which is used by both ØUT1 and ØUT2. First, the entry THERM1 of subroutine THERM is called to compute the nondimensional species specific heats C_{pi}/R_0 = CCPJ(I) at the current nondimensional temperature CT = T/T_0. The nondimensional molar specific heat of the gas mixture is then computed as

$$CP \equiv \frac{C_p}{R_0} = \sum_{i=1}^{n} X_i \frac{C_{pi}}{R_0} \qquad (120)$$

Here the species mole fractions X_i are obtained from the array SAVEC(I), where they have been placed by the calling routine. Then the thermodynamic relation

$$C_p - C_v = R_0 \qquad (121)$$

is used to obtain the specific heat ratio, γ :

$$\gamma = \frac{C_p}{C_v} = \frac{1}{1 - \frac{R_0}{C_p}} \qquad (122)$$

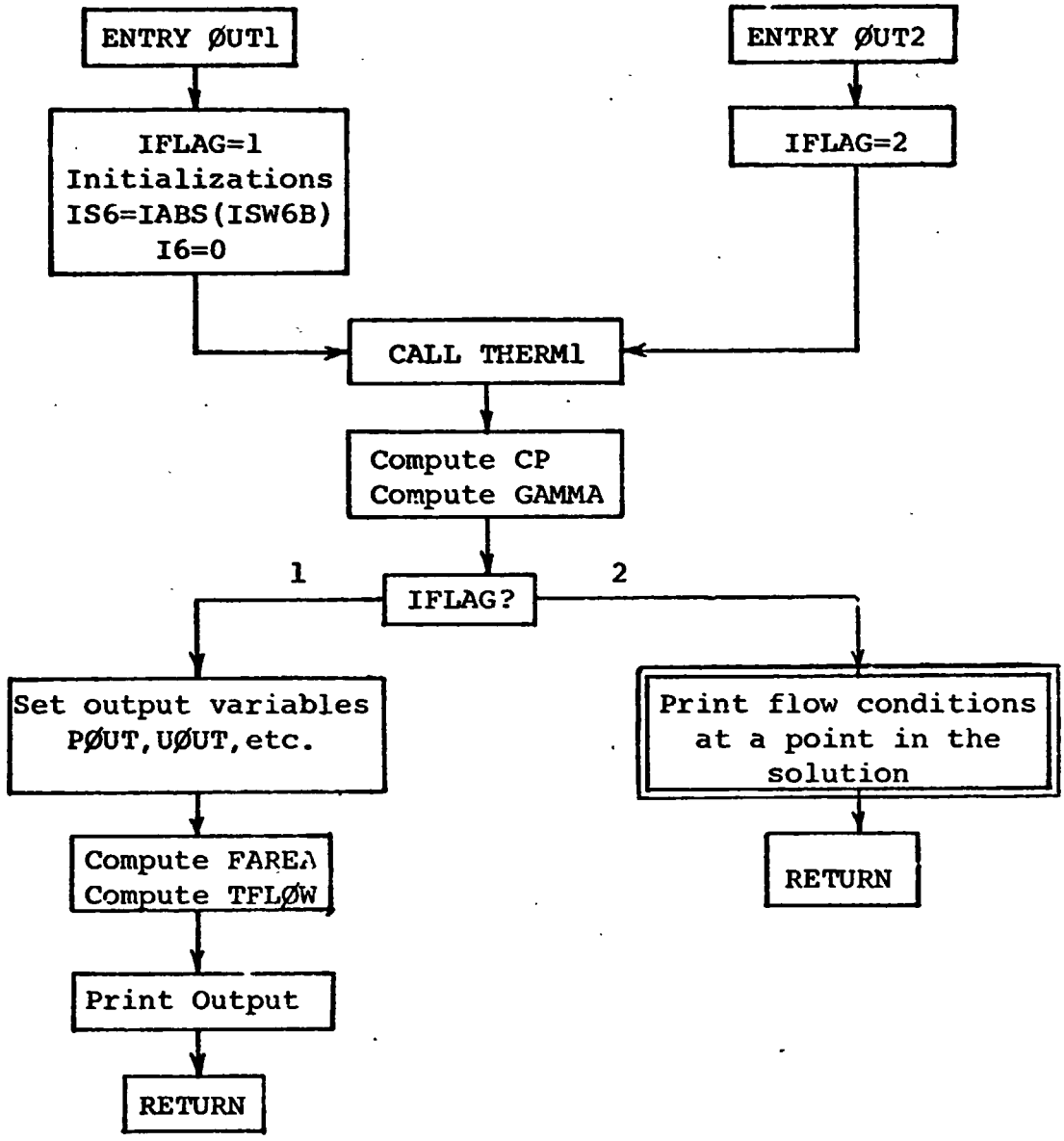


Figure 23. Overall Flowchart of Subroutine ØUT1

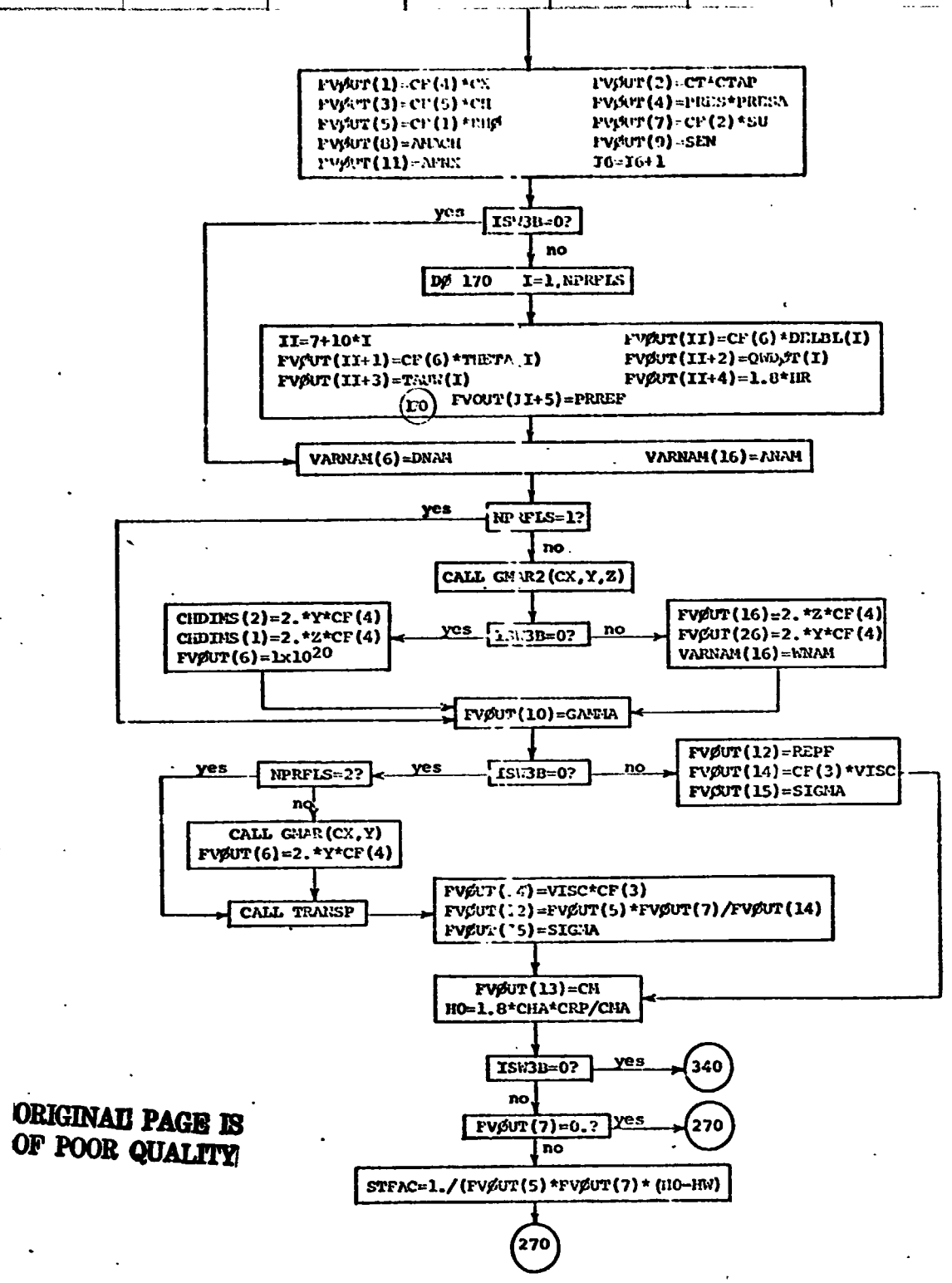
At the conclusion of this calculation, the indicator IFLAG is used to transfer control to the appropriate portion of the subroutine, depending on which entry point was called. If entry was through ØUT1, control passes to statement 30. The variables used for output of the flow conditions in ØUT1, namely PØUT, UØUT, etc., are then set. Next, the effective cross sectional area of the flow at the throat, FAREA, is computed. For ISW3B = 0 (boundary layer neglected), FAREA is based upon the geometric dimensions of the nozzle or channel at the throat. For ISW3B $\neq$ 0, the displacement thickness at the throat is taken into account using equation I(125), I(129), or I(133) as appropriate. After it has been calculated in square centimeters, FAREA is converted into square feet. The total mass flow is then computed by multiplying FAREA by the sonic mass flux in lbm/ft^2sec.

Finally, the flow conditions, including the calculated mass flow TFLØW, are printed out in the form of a list. This is followed by a list of species mole fraction.

3.37.2 Entry ØUT2

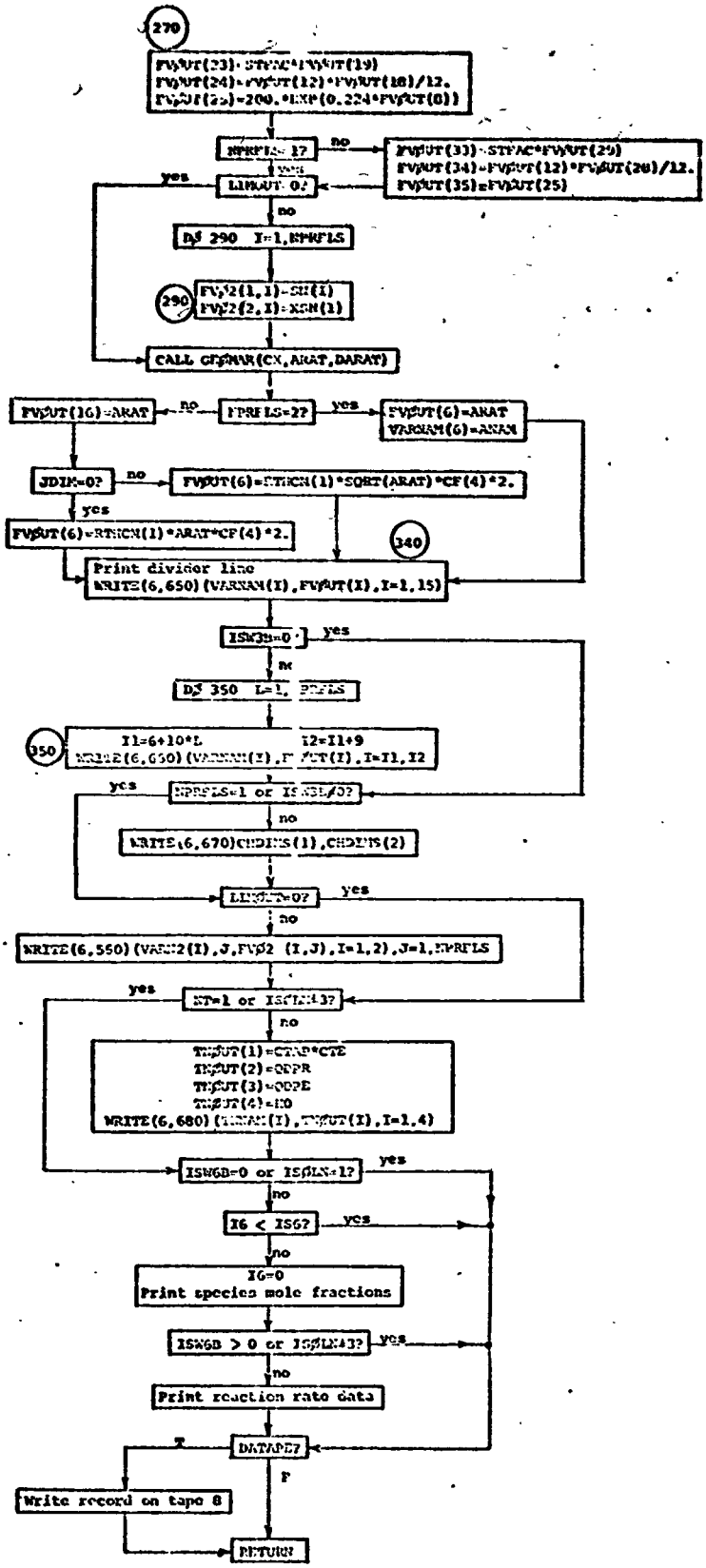
The operations performed when the entry ØUT2(ISØLN) is called, which are represented by a double box in figure 23, are flowcharted in detail in figure 24. The complexity of this portion of the subroutine results from the need to deal with various special cases:

ISW3B $\neq$ 0	Boundary layer included; output includes boundary layer displacement and momentum thicknesses, heat flux, shear stress, Stanton number, etc.
ISW3B = 0	Boundary layer neglected.
NPRFLS = 1	Flow is in a nozzle.
NPRFLS = 2	Flow is in a channel; two full sets of boundary layer outputs if ISW3B $\neq$ 0.
JDIM = 0	Two dimensional nozzle.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 24a. Flowchart of the Operations Produced by Calling Entry ØUT2 of Subroutine ØUT1 (Part a)



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 24b. Flowchart of the Operations Produced by Calling Entry SUT2 of Subroutine SUT1 (Part b)

JDIM = 1 Axisymmetric nozzle.

LIMØUT ≠ 0 Include output of SN, XSN boundary layer parameters.

LIMØUT = 0 Omit this output.

The printed outputs are, in most cases, expressed in different physical units from those used in the internal computations of the code. Thus, a major function of ØUT2 is to re-express these quantities in the desired output units. The converted flow variables in output units are loaded into an array FVØUT(I), dimensioned (35). This scheme allows the use of compact WRITE statements based on implied DØ loops. The converted quantities in FVØUT are also used in the calculations of conditions on wedge models; they are communicated to subroutine WEDGE through the common block ØUTPUT. Definitions of the FVØUT array elements are given in the glossary of Fortran symbols, Section 4.31. However, the definitions will be reviewed here to facilitate the study of the flowchart (figure 24).

The following elements of FVØUT always have the meanings indicated:

(1) Axial coordinate, x (inches)

(2) Temperature, T (^{o}K)

(3) Enthalpy, h (Btu/lb)

(4) Pressure, p (atm)

(5) Density, ρ (lbm/ft^3)

(7) Velocity, u (ft/sec)

(8) Mach number, M

(9) Entropy, s (Btu/lbm-^{o}R)

(10) Frozen specific heat ratio, γ

(11) Effective area ratio, A_e

(12) Reynolds number per foot

(13) Molecular weight (g/mole)

(14) Viscosity (lbm/ ft-sec)

(15) Electrical conductivity (mho/cm)

Element number 6 is absent from this list because its meaning varies with the type of case, as explained below. For ISW3B $\neq$ 0, the Reynolds number per foot, viscosity, and electrical conductivity are calculated in BLAYER, and ØUT2 obtains these data from common blocks /BLØUT/ and /TRPRØP/. For ISW3B = 0, BLAYER is not called and these quantities must be computed in ØUT2. The viscosity and conductivity are obtained by calling subroutine TRANSP.

If the boundary layer is included (ISW3B $\neq$ 0), the ten elements FVØUT(16) to FVØUT(25) are loaded with data relating to the boundary layer. If, in addition, a channel geometry is being used (NPRFLS = 2), these elements contain the data for the boundary layer on the first profile, and the ten elements FVØUT(26) to FVØUT(35) contain the corresponding data for the second profile. The meanings for the elements numbered 17 to 25 are as follows:

(17) Boundary layer displacement thickness (inch)

(18) Boundary layer momentum thickness (inch)

(19) Heat flux to nozzle wall (Btu/ft^2-sec)

(20) Shear stress on nozzle wall (lbf/ft^2)

(21) Recovery enthalpy (Btu/lbm)

(22) Prandtl number at the reference temperature

(23) Stanton number

(24) Reynolds number Re_θ based on momentum thickness

(25) Value of Re_θ for boundary layer transition

The data on geometric area ratio and nozzle diameter or channel widths is placed into FVØUT(6), FVØUT(16), and FVØUT(26). The arrangement of these data depends upon the type of case being treated:

A. If ISW3B = 0, NPRFLS = 1: FVØUT(6) = nozzle diameter (inch). In this case, the geometric area ratio is equal to the effective area ratio, FVØUT(11).

B. If ISW3B = 0, NPRFLS = 2: FVØUT(6)is set to a large value which exceeds the width of the F format used and thus prints as a row of asterisks. The width of the channel surface lying on the first profile is set to CHDIMS(2). The corresponding width for the surface with the second profile goes into CHDIMS(1). These CHDIMS(I) values are printed on a separate line in the block of output. The geometric area ratio is again equal to the effective area ratio.

C. If ISW3B $\neq$ 0, NPRFLS = 1: FVØUT(16)= geometric area ratio, FVØUT(6) = nozzle diameter (inch).

D. If ISW3B $\neq$ 0, NPRFLS = 2: FVØUT(6) = geometric area ratio; FVØUT(16) = width of channel surface lying on the first profile (inch); and FVØUT(26) = width of channel surface lying on the second profile (inch).

Note that in the channel case (NPRFLS = 2), the width of the surface lying on a given profile is twice the ordinate of the <u>other</u> profile.

After the data have been loaded into the output arrays, they are printed out. First, a line of asterisks is printed to separate the new block of output from the previous one. Imbedded in this line is a word ("frozen", "equilibrium", or "nonequilibrium") identifying the type of solution being

computed, based upon the entry-point argument ISØLN. In the case of a nonequilibrium solution, the number of integration steps since the last output of flow variables (NSTEPS) and the value of the indicator INEQ are also printed in this divider line. Note that NSTEPS is initialized in ØUT1 to the seven-digit value 1000000, so that when printed (with an I6 format) in the frozen and equilibrium solutions, it gives a row of six asterisks.

Following the divider line, the first fifteen elements of FVØUT(I) are printed by a single WRITE statement. The array VARNAM(I) appearing in this statement contains the Hollerith labels for the output variables. Then, if ISW3B is nonzero, the boundary layer data in FVØUT(16) to FVØUT(25) or FVØUT(35) are printed. For channel cases without the boundary layer, the channel widths CHDIMS(I) are printed on a separate line. If LIMØUT is nonzero, the boundary layer parameters stored in FVØ2(I,J) are printed on a separate line. For nonequilibrium solutions based on an electronic nonequilibrium model, the electron temperature, the radiative and electronic energy transfer terms, and the total enthalpy are loaded into TNØUT(I) and printed.

Next, if ISW6B is nonzero and ISØLN is not equal to 1, and if I6 is equal to IS6, the species mole fractions are printed. The test on I6 provides output of these species data every IS6th time ØUT2 is called in the equilibrium and nonequilibrium solutions, where IS6 = |ISW6B|. Finally, if ISW6B is negative and ISØLN = 3 (nonequilibrium solution) and if I6 = IS6, a table of reaction rate data is printed out. This table includes P_i, $\mathcal{X}_i$, and $P_i \mathcal{X}_i$ for all of the reactions (i = 1 to r), and $d \ln \gamma_j/dx$ for all the species (j = 1 to n).

If DATAPE is .TRUE., a record is written on tape 8 for subsequent use by the NATA plot program.

3.38 Subroutine PERT

The function of subroutine PERT is to compute the perturbations in the flow variables by solving the following system of n+2 linear equations:

Equation Index	Equation
1 to c	I(368) for j = 1 to c
c+1 to n	I(362) for j = c+1 to n
n+1	I(374)
n+2	I(380)

The unknowns are PERTGJ(J) = $\delta\gamma_j$ for j = 1 to n, PCT = $\delta T/T_0$ and PRHØ = $\delta\rho / \rho_0$, in the order listed. The simultaneous solution is obtained by calling subroutine DSMSØL.

The coefficients and constants for the above system of equations are set up in the common array AA. First, the region of AA which is to be used is zeroed. Then, in the DØ loops ending at statement 20, the coefficients for equations I(368) are set into AA(I,J) for I = 1 to ISC. The coding down to the statement following 110 then loads the coefficients of equations I(362) into AA(I,J) for I = ISC + 1 to ISS, and loads the coefficients for equation I(374) into AA(ISSP1,J). In the calculation of the coefficients for I(362), the Fortran index I corresponds to j in I(362), J corresponds to k and K corresponds to i. The coding from DØ 120 down to the second statement below 130 sets up the coefficients for equation I(380) in AA(ISSP2,I). Finally, the loop DØ 140 loads the constant terms of equation I(362) into AA(I,ISSP3) for I = ISCP1 to ISS. The remaining equations are all homogeneous, so that their constant terms have been set by the initial zeroing of the AA array.

After DSMSØL has been called, the values obtained for the perturbations are retrieved from the column AA(I,ISSP3). Then the $\delta\chi_i$ values, denoted by SDCHI(I), are calculated from equation I(360).

3.39 Function PIØMEG

This routine provides the Maxwell-averaged momentum transfer cross section for collisions of electrons with

neutral species as a function of the electron temperature, TE. The cross section is calculated by linear interpolation in a table stored in common block /TNEQ/.

3.40. Subroutine PRØP

The function of this routine is to compute the flow conditions at a specified temperature in the frozen flow solution. Since PRØP makes use of species thermal properties, a call to PRØP must always be preceded by a call to THERM.

The technique used by PRØP has been discussed in Section 6.4 of Volume I. The coding is straightforward. As noted in Section 3.22.1, the flow variables are computed in nondimensional form. In particular, equation I(267) for the pressure is written in the form

$$ZP = \ell n \left(\frac{p}{p_0}\right) = -\ell n\ p_0 + \sum_{j=1}^{n} X_{j0}\ \frac{s_j^o}{R_0} - \sum_{j=1}^{n} X_{j0}\, \ell n\, X_{j0} - \frac{W_0 s_0}{R_0} \tag{123}$$

The first sum on the right is evaluated in PRØP as

$$\sum_{j=1}^{n} X_{j0}\ \frac{s_j^o}{R_0} = W_0 \sum_{j=1}^{n} \gamma_{j0} s_j^o = CMA \cdot S4 \tag{124}$$

The remaining three terms on the right in (123) are computed as $W_0 \cdot$ ENT, where ENT has been calculated in FRØZEQ before PRØP is first called:

$$ENT = -\sum_{j=1}^{n} \gamma_{j0}\ (\ell n\, \gamma_{j0} + \ell n\, W_0) - \frac{\ell n\, p_0}{W_0} - \frac{s_0}{R_0} \tag{125}$$

3.41 Subroutine PRTA

This subroutine controls the output of the flow conditions and executes the calls to the MØDEL routine during

the nonequilibrium solution. It also calls subroutine BLCALL with the argument FINAL = .TRUE., to update the derivative of the boundary layer displacement thickness and provide permanent increments to the Cohen-Reshotko boundary layer integral. PRTA calls BLCALL even if the boundary layer is being neglected in the flow solution (ISW3B = 0), because the output and model routines need the species mole fractions, which are computed and loaded into common block /TEMPRY/ by BLCALL. For ISW3B = 0, the call to the boundary layer routine BLAYER is suppressed in BLCALL.

The step size in the nonequilibrium integration is often quite small, especially in the region upstream of the throat, where some of the reactions may still be nearly in equilibrium. To avoid the excessive output which would result if the flow conditions were printed out for every step in such regions of small step size, NATA prints the results of the nonequilibrium solution at intervals of TPRINT in the nondimensional temperature CT. The control parameter TPRINT is preset to 0.01, and is under input control (with the input name TPRNTI). A counter (NSTEPS) is used to determine the number of steps computed between successive printouts of the flow conditions. This information is included in the printed output.

If the flow solution is being generated by the perturbation technique (INEQ = 0), the temperature perturbation PCT is added to CT before subroutine ØUT2 is called to produce the output. Then PCT is subtracted to restore CT to its equilibrium flow value.

3.42 Subroutine PUTQIN

The computation of the cross sections $\bar{\Omega}_{ij}^{(\ell,s)}$ required in the transport property calculations for any particular conditions of gas temperature, pressure, and composition is carried out in the NATA code under the general control of subroutine PUTQIN. The subroutine selects data for each step of the computations from the edited cross section data in common block/TRANS7/, calls the appropriate subroutine to carry out the computations, and then returns the computed

cross sections for all species pairs to subroutine TRANSP in the Q(K,I,J) array of common block /TRANS1/ (see the discussion of subroutine TRANSP). The argument X of subroutine PUTQIN is the partial pressure of electrons in the specified gas mixture, in atmospheres, a quantity which is needed in the computation of effective Coulomb cross sections from equations I(100).

Before beginning the cross section computations, subroutine PUTQIN initializes Q(K,I,J) to zero for all pairs of species included in the cross section computations.* If the first entry in the V array is positive, as it should always be for proper operation of the code, Q(1,2,1) is set to 0.8 Q_c, where Q_c is the quantity defined by equation I(100b), for later use in the computation of effective Coulomb cross sections for the mixture. The remainder of the subroutine down to statement 240 then consists of a loop which goes through the steps L of the cross section computation one by one and accumulates the values of the cross sections for all species included in the computations in the array Q(K,I,J). In this computation, the index K = KQ(L) indicates the option to be used in performing the L^{th} step of the computations (see Section 4.6 of Volume II), M is the location of the first parameter for the L^{th} step of the computations in the parameter array V, and LQ1 and LQ2 are the locations respectively of the first and last pairs in the IQ, JQ array to which the computations of the L^{th} step are to be applied. For each step L of the computations, subroutine PUTQIN determines the option which is to be used for that step from the KQ array and then calls the appropriate subroutine to carry out the computations for that option, as indicated in Table II. For the Coulomb cross sections (KQ = 2) however, these computations are carried out in subroutine PUTQIN itself using the previously computed value $0.8Q_c$.

---

*Note that the number N of species in the cross section computations, which is stored in common block /TRANS8/, is not necessarily the same as the number of species in the transport calculations which is stored in common block /TRANS5/.

TABLE II

SUBROUTINES USED IN CROSS SECTION COMPUTATIONS

Option KKQ	Subroutine Used	Type
2	PUTQIN	1
3	QEXP	1
4	QEX	3
5	QTAP	1
6	QREPP	1
8	QLJ	1
9	QSAME	1
10	QMIX	2
11	Q11	2
12	Q12	1
13	Q13	2
14	Q14	2

As indicated in Table II, the subroutines called by PUTQIN may be divided into three general types, based on the manner in which they return their output. The first and most commonly used type, 1, returns the three cross section values $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$ and $B^* \bar{\Omega}^{(1,1)}$ computed for the step to subroutine PUTQIN in the three locations ØM(1), ØM(2), and ØM(3) respectively of the dimensioned variable ØM. The loop from statement 190 through 200 of PUTQIN then adds these computed values to the previously computed cross sections Q(K,IQ,JQ), for each pair of species in the IQ,JQ array to which the step applies. For the type 2 subroutines, on the other hand, the cross sections are loaded directly into the Q array by the subroutine which calculates them, and statements 190 through 200 in PUTQIN are bypassed. The reason for this difference is that for the type 2 subroutines the computed values of the cross sections for a species depend on the indices of the species, so that a single set of cross section values cannot be returned for all pairs of species to which the step applies.

The single type 3 subroutine in the code, QEX (charge exchange cross section), returns the computed cross section values for the step in the variable ØM as do the type 1 subroutines. However, in this case, the coding of subroutine PUTQIN offers a choice, determined by the input values of V for the option, as to whether the computed cross sections for the step will be added to the previous cross sections by transferring control to statement 190, as for a regular type 1 subroutine, or will be used to replace the previous values of $\bar{\Omega}^{(1,1)}$ and $B^* \bar{\Omega}^{(1,1)}$ by transferring control to statement 170 and bypassing 190.

The final section of coding between statements 210 and 240 of PUTQIN produces a voluminous dump of cross section data under control of the input parameter ISW8B. For ISW8B $\leq$ 0, this dump is not produced. For ISW8B > 0, the dump is produced the first time PUTQIN is called and every ISW8Bth time thereafter. Note that PUTQIN is called once each time subroutine TRANSP is executed to compute transport properties. For each step in the cross section calculations, the dump includes a line of output giving the number of the step; the code number KKQ of the cross section option used in the step; the positions in the IQ, JQ arrays of the first (LQ1) and the last (NQ(L)) pairs of species for which cross sections are calculated in the step; and the position M of the first unused parameter in the V array. Below this line of indexing information, the calculated values of the averaged cross sections $\bar{\Omega}_{ij}^{(1,1)}$ for all pairs of species in the mixture are printed out as they exist at the conclusion of the step. Together with the initial edit of the cross section calculations (Section 3.9 of Volume II), this PUTQIN dump allows detailed checking of the operation of the cross section calculations for a new gas model, since the effect of each step in the calculations upon each pair cross section $\bar{\Omega}_{ij}^{(1,1)}$ is clearly shown. The routine could easily be modified to print $\bar{\Omega}_{ij}^{(2,2)}$ and $B_{ij}^* \bar{\Omega}_{ij}^{(1,1)}$, in addition.

3.43 Subroutine QCØ

Subroutine QCØUL computes the factor $0.8Q_c$ in the Coulomb cross section from equation I(100b). Inputs for this computation are the temperature T in °K and the

electron pressure $X = n_e kT$ in atmospheres. The computed value of $0.8Q_c$ in 10^{-16} cm^2 is returned in ØM1. For $C \leq 0$, $\ell n(f\Lambda)$ in I(100b) is set to 1.0 while for $C > 0$ it is calculated correctly from I(100c) and I(100d). To prevent overflow when $n_e = 0$, a small quantity ($=10^{-50}T^4$) is added to X before carrying out the computations.

3.44 Subroutine QEX

Subroutine QEX computes the averaged cross sections $\bar{\Omega}^{(1,1)}$ and $B^*\bar{\Omega}^{(1,1)}$ in 10^{-16}cm^2 for a pair of species between which there is an exchange interaction (i.e., an interaction such as charge exchange in which the identities of the two species are interchanged) having a cross section of the form

$$Q_{ex} = (A - B \log_{10} v)^2, \tag{126}$$

where v is the relative velocity in cm/sec. The formula for the averaged cross sections $\bar{\Omega}^{(\ell,s)}$ is then (ref. 9):

$$\bar{\Omega}^{(\ell,s)} = 2\left[A - \alpha_s B - \tfrac{1}{2} B \ell n(T/W)\right]^2 + 2\beta_s B^2 \tag{127a}$$

where

$$\alpha_s = \tfrac{1}{2}\left[\ell n\,(4k/N_0) + \sum_{n=1}^{s+1}\left(\frac{1}{n}\right) - \gamma\right]\log_{10} e$$

$$\beta_s = \tfrac{1}{4}\left[\frac{\pi^2}{6} - \sum_{n=1}^{s+1}\left(\frac{1}{n^2}\right)\right](\log_{10} e)^2 \tag{127b}$$

$\gamma = 0.577216\ldots$ is Euler's constant, and W is the molecular weight. Putting the numerical values into equations (127) and I(91) then gives the formulas used for $\bar{\Omega}^{(1,1)}$ and $B^*\bar{\Omega}^{(1,1)}$ in the subroutine. Since the cross section $\bar{\Omega}^{(2,2)}$ is not affected by exchange (ref. 10), subroutine QEX sets the contribution ØM(2) to this cross section equal to zero.

3.45 Subroutine QEXP

Subroutine QEXP computes the averaged collision cross sections $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$, and $B^* \bar{\Omega}^{(1,1)}$ in 10^{-16} cm^2 for a pair of species interacting according to the exponential potential $\phi = Ae^{-r/\rho}$, where r is the distance between the two species in Å. These cross sections are obtained by linear interpolation in Monchick's tables (ref. 11) of α versus $I_{(1,1)}$, A^*, and B^*, where

$$\alpha = \ln (A/kT)$$

$$I_{(1,1)} = \bar{\Omega}^{(1,1)} / (4 \pi \alpha^2 \rho^2) \qquad (128)$$

and

$$A^* = \bar{\Omega}^{(2,2)} / \bar{\Omega}^{(1,1)},$$

which are stored in common block /TRANS4/. The table lookup and interpolation are carried out by a call to subroutine QINTRP. The coding assumes that there are 50 entries in these tables.

3.46 Subroutine QINTRP

Subroutine QINTRP is used by several of the other cross section subroutines for interpolating tabular data stored in common block/TRANS4/. In this interpolation, the independent variable is assumed to be stored in monotonically increasing order in the array TL, starting at location N1, and the corresponding values of the three dependent variables in the corresponding locations of the ØMEGA1, ASTAR and BSTAR arrays. The argument N of the subroutine represents the number of data points to be included in the table, and A and B are respectively the values of the independent and dependent variables for the interpolation. The subroutine searches the data in the TL array for the specified value A of the independent variable and, if A is found to lie within the range of the table, carries out a linear interpolation of the tabular data to determine the values B(J) of the three dependent variables corresponding to the specified value A of the independent variable. If, on the other

hand, A lies outside the range of the table, the last two data points in the table are linearly extrapolated to determine the values of the dependent variable B and, if ISW8B $\neq$ 0, a message is printed out indicating that an extrapolation of the data was required. This message gives the location in the TL array at which the extrapolation occurred, the desired value A of the independent variable, and the two values of the variable from the TL array used in the extrapolation.

3.47 Subroutine QLJ

Subroutine QLJ determines the collision cross sections $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$ and B* $\bar{\Omega}^{(1,1)}$ at temperature T for a pair of species interacting according to the Lennard-Jones 6-12 potential $\phi = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^{6}]$ by linear interpolation in tables of $T^* = kT/\epsilon$ versus $\bar{\Omega}^{(1,1)*} = \bar{\Omega}^{(1,1)}/\pi\sigma^2$, $A^* = \bar{\Omega}^{(2,2)}/\bar{\Omega}^{(1,1)}$, and B* which have been computed for this potential by Monchick and Mason (ref. 12). These tables contain data for 37 reduced temperatures T* from 0.1 to 100 and are stored in the arrays TL, OMEGA1, ASTAR, and BSTAR respectively in common block/TRANS4/, starting at location N1 = 501 in the arrays.

3.48 Subroutine QMIX

Subroutine QMIX computes cross sections from the empirical mixing rule I(102) and adds them to the previously computed cross sections Q(K,I,J) for a series of species pairs given in the IQ, JQ arrays in common block/TRANS7/. The arguments LQ1 and LQ2 of the subroutine indicate the positions of the first and last pairs in the IQ, JQ array to which these computations are to be applied.

To avoid computing the square roots required in equation I(102) several times in a single call to QMIX, these quantities are stored in the array

$$SQT(K,I) \equiv \tfrac{1}{2}\sqrt{Q(K,I,I)} \qquad (129)$$

Upon each entrance to the subroutine, the values of SQT(1,I) are set to 0.0 for all species I, to indicate that they have not yet been calculated in the current call to QMIX. (Note that values of SQT cannot be saved over from one call to QMIX to the next, since the values of the cross sections Q(K,I,I) may have changed between the two calls.) The remainder of the subroutine then consists of a loop over the species pairs I, J which are to be included in the computation. For each pair, the subroutine checks the values of SQT(1,I) and SQT(1,J) to determine if they have been calculated previously in the loop, and, if not, computes the required values of SQT from (129) for K = 1, 2, and 3. The SQT values for the pair are then used in equation I(102) and the results added to the previously computed cross sections Q(K,I,J) to obtain the final cross section values for the pair.

Since the edited cross section data prepared by subroutine XSECT may contain steps using the option KQ = 10 which do not apply to any species pairs, a provision to bypass the cross section computations completely when LQ1 > LQ2 is included in the subroutine QMIX to allow for this possibility

3.49 Subroutine QREPP

Subroutine QREPP computes the averaged cross sections $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$, and $B^* \bar{\Omega}^{(1,1)}$ in units of 10^{-16} cm^2 for a pair of species which interact according to an attractive or repulsive inverse power potential of the form $\phi = Ar^{-\eta}$, where r is the distance between the species in Å. The cross section formulas for this potential are (ref. 13):

$$\bar{\Omega}^{(1,1)} = \text{ØMEGA1(ITL)} \; / \; (T^{2/\eta})$$

$$\bar{\Omega}^{(2,2)} = \text{ASTAR(ITL)} * \bar{\Omega}^{(1,1)} \qquad (130)$$

$$B^* \bar{\Omega}^{(1,1)} = \text{BSTAR(ITL)} * \bar{\Omega}^{(1,1)}$$

where ØMEGA1(ITL), ASTAR(ITL), and BSTAR(ITL) are input quantities given by equations I(105).

3.50 Subroutine QSAME

Subroutine QSAME sets the cross sections for a species pair equal to a constant multiple of those computed previously for another pair (I,J). The coding of the subroutine permits a different constant factor to be used for each of the three cross sections $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$, and $B^* \bar{\Omega}^{(1,1)}$.

3.51 Subroutine QTAB

Subroutine QTAB obtains the values of the cross sections $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$, and $B^* \bar{\Omega}^{(1,1)}$ for a pair of species at a given temperature T by linear interpolation in tables of $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$ and B* versus temperature for the species which are included in the input. The tabulated data are assumed to be given in the arrays ØMEGA1, ASTAR, BSTAR, and TL, respectively, in common block/TRANS4/,with the tables beginning at location N1 in the arrays and containing NL data points. An option is also provided to multiply the tabluated cross sections by an arbitrary constant factor A in the computations.

3.52 Subroutine Q11

Subroutine Q11 multiplies the previously computed cross sections for a series of species pairs by the temperature dependent factor f(T) given by equation I(108). The subroutine computes the value of this factor at the specified temperature T from the input data for the subroutine and then calls subroutine Q14 with the appropriate inputs to carry out the actual multiplication of the cross sections by this factor.

3.53 Subroutine Q12

Subroutine Q12 computes the cross sections $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$, and $B^* \bar{\Omega}^{(1,1)}$ for a species pair from the generalized mixing rule given by equation I(109). As with the other type 1 subroutines discussed above, it returns the computed cross sections to subroutine PUTQIN in the dimensioned variable ØM.

3.54 Subroutine Q13

Subroutine Q13 computes the cross sections Q(K1,I,J) according to equations I(110) and I(111) for a fixed value of K1 specified in the cross section data ($1 \leq K1 \leq 3$) and for a series of species pairs I,J given in positions LQ1 through LQ2 of the IQ, JQ arrays. The computed cross sections then replace the previous values for these cross sections in the Q(K,I,J) array.

3.55 Subroutine Q14

Subroutine Q14 computes the cross sections $\bar{\Omega}^{(1,1)}$, $\bar{\Omega}^{(2,2)}$, and $B^* \bar{\Omega}^{(1,1)}$ for a series of species pairs by multiplying the previously computed values of these cross sections by a constant factor, as indicated in equation I(112). The computed cross sections then replace the previous values for these cross sections in the Q(K,I,J) array.

3.56 Subroutine RADIUS

This small routine performs geometric calculations that are required at several points in the computation of the boundary layer at the first flow point in subroutine BLAYER. The inputs to RADIUS are the type of nozzle geometry ITYPE, the axial coordinate X, and the profile index L. The quantities computed and returned to BLAYER are the profile ordinate R, the geometric area ratic AG, and the factor AGJ, which is proportional to r^{2j} in equation I(171).

For ITYPE = 1 (two-dimensional nozzle), R is obtained by calling the entry GMAR of subroutine GEØMAR. Then AG = R/R0 where R0 is the half-width of the throat gap, and AGJ = 1 since j = 0.

For ITYPE = 2 (axisymmetric nozzle), R is again obtained by calling GMAR, AG = $(R/R0)^2$, where R0 here is the throat radius, and AGJ = AG since j = 1.

For ITYPE = 3 (rectangular channel), the ordinates YZ(1), YZ'2) of the two profiles are obtained by calling the entry GMAR2 of subroutine GEØMAR, AG = YZ(1)· YZ(2)/ Y0Z0 where Y0Z0 is the product of the profile ordinates at the throat, R = YZ(L), and AGJ = $[YZ(M)]^2$, where M = 2 if L = 1 and M = 1 if L = 2.

3.57 Subroutine READ

READ is the input routine for NATA. It reads the input data, sets up the geometric description and gas model for the problem, and initializes most of the control parameters.

3.57.1 Input Operations

The subroutine begins by reading the input data for the current case. The routine contains four READ statements, of which the first two are always executed. The first of these statements reads a comment card into the array ACØM. This information is subsequently written out as part of a heading for the output for the case. The second READ statement reads data for the variables included in namelist INPUT. This is the main input operation for the code. Namelist INPUT contains all of the inputs to NATA except those required for defining chemical elements, species, and reactions (read under namelist EINPUT), and those required for calculating the transport cross sections of species (read under namelist TINPUT). When the code is run using the chemical and cross section data compiled into the BLØCK DATA routines, as it is normally, the statements READ(5, EINPUT) and READ(5, TINPUT) are not executed.

All of the variables in INPUT are printed out at the head of the output for the case by the statement WRITE (6,INPUT). If EINPUT and TINPUT are read, their variables are also printed out in namelist format. The reason for breaking the code inputs into these three groups is to minimize the number of pages of such output in normal usage, when EINPUT and TINPUT are not used.

3.57.2 Sonic Mass Flux Calculation

If ISW2B $\leq$ 0, NATA determines the equilibrium conditions in the reservoir from data on the reservoir pressure or stagnation enthalpy and the total mass flow. In these cases, the sonic mass flux SMASS is required in the calculations to determine the reservoir temperature. This quantity is computed in READ from the total mass flow $\dot{M}$ (denoted by FLØW) and the geometric dimensions of the throat. The coding treats three cases:

(1) Two-dimensional nozzle

Here $\dot{M}$ and FLØW denote the mass flow per unit length of the throat gap. Hence

$$\text{SMASS} = \frac{\dot{M}}{D} \qquad (131)$$

where D is the throat gap. Since SMASS is required in gm/cm^2-sec while $\dot{M}$ is in lb/in-sec and D in inches, a numerical conversion factor from lb/in^2-sec to gm/cm^2-sec is inserted on the right. This factor is $453.5924/(2.54)^2 = 70.3069$.

(2) Axisymmetric nozzle

In this case, $\dot{M}$ is the total mass flow in lb/sec, and

$$\text{SMASS} = \frac{4\dot{M}}{\pi D^2} \qquad (132)$$

Insertion of the conversion factor (70.3069) gives a numerical coefficient of $4 \times 70.3069/\pi = 89.5173$.

(3) Rectangular channel

The sonic mass flux here is

$$\text{SMASS} = \frac{\dot{M}}{D_1 D_2} \qquad (133)$$

where D_1 and D_2 are the throat gaps of the channel's two profiles.

3.57.3 Automatic Air Model Selection

If the input AAMS (acronym for "automatic air model selection") is .TRUE., and IGAS is either 1 or 2, subroutine READ resets IGAS to either 1 or 2 on the basis of an enthalpy or temperature criterion. IGAS = 1 gives the large air model, including five ion species, which is required in cases

with high reservoir temperatures. IGAS = 2 gives the small air model which includes only a single ion (NO^+) and which can be used at lower temperatures. If the reservoir temperature is read in (ISW2B > 0), the air model is selected to be IGAS = 1 or 2 according as the reservoir temperature CTAPI is greater than or less than 6000°K. On the other hand, if the reservoir temperature is to be determined from mass flow data (ISW2B $\leq$ 0), then its value is not yet known at the time when READ is executed. In this case, the air model selection is based on a stagnation enthalpy criterion of 8000 Btu/lb. For ISW2B < 0, the stagnation enthalpy is an input. For ISW2B = 0, the inputs are stagnation pressure and mass flow; in this case, the stagnation enthalpy is estimated from the Winovich correlation (ref. 14). The Winovich estimate of the stagnation enthalpy is computed from the formula

$$\text{HWINØV} = \left[\frac{136.7\ p_0}{m_*}\right]^{2.52} \qquad (134)$$

where p_0 denotes the reservoir pressure in atmospheres and m_* the sonic mass flux in g/cm^2 sec. The numerical factor on the right in (134) differs from the coefficient (280) in Winovich's equation because it includes a factor converting m_* from g/cm^2-sec to lb/ft^2-sec.

3.57.4 Initializations Based on Inputs

With few exceptions, the input variables of NATA are not altered by execution of the program. Also, the Namelist input system resets only those variables which are actually referenced in the input cards for a given case. Thus, when several cases are run in a single job, the input cards for the second and subsequent cases need contain data only for the variables which the user wishes to change, e.g., the pressure, mass flow, nozzle index, etc.

The input-determined variables which are not altered by program execution are included directly in the namelists INPUT, EINPUT, and TINPUT. However, some input-determined variables are altered by the program. In such cases, the variable which is altered and its corresponding input are given different names and thus are stored in separate

locations. The input name is the internal variable name (or a contraction thereof) with the letter I added at the end. At the cost of some increase in storage requirements, this arrangement protects the input variables against alteration during execution.

3.57.5 List of Test Sections

The data in the input array TSDIAM specify points in the flow at which special calculations are to be done. In the case of a two-dimensional or axisymmetric nozzle, the entries in TSDIAM are nozzle diameters at which model-condition calculations are to be perfcrmed. For a channel, the TSDIAM entries are specified channel widths at which free-stream flow and boundary layer calculations are to be performed.

Subroutine READ uses the data in the TSDIAM array to set up another array, TSAR, whose values are actually used by the program to select the points at which the special calculations are done. For a nozzle, the TSAR entries are geometric area ratio values. For a channel, the TSAR are widths in centimeters. In generating the TSAR array, READ rejects any TSDIAM values which are less than or equal to the throat diameter for the profile involved, and determines the number (NTS) of valid TSAR entries. All the entries of TSDIAM are preset (in BLØCK DATA) to the value 1×10^{20}. Thus, any values in excess of 1×10^{20} are assumed not to have been set by the user in input, and are not regarded as valid points.

3.57.6 Geometric Description

A nozzle is specified by one profile curvefit, a channel by two. For each profile, the parameters defining the geometry of the nozzle or channel may be either read in as part of the input or, if one of the standard geometries is desired, obtained from precoded data. For the Jth profile (J = 1 or 2), if NPRØFL(J) is nonzero, the geometric data are looked up in the ZPRP array, which is set in BLØCK DATA. This array occupies the common block /NØZZ/. If NPRØFL(J) is zero, the geometric data are set on the basis of the geometric input variables such as NSECTS, DIAM, and PARAMI.

In either case, after the geometric model has been set up, subroutine READ prints out a summary description of it, unless the standard geometry specified is the same as in the preceding case in the job.

3.57.7 Extra Chemical Elements

If additional chemical elements have been defined in the EINPUT input, the data for them are loaded into the EPRP array, which contains the precoded data for elements. The number of input-defined elements is NEELS. For the Ith such element, the atomic number is obtained from EEPRP(1,I) and the atomic weight from EEPRP(2,I). The index assigned to the element in the master list of elements is IEEP(I). The Hollerith atomic symbol for the element is obtained from the array ASYM, which is set in a data statement in READ.

3.57.8 Gas Model

The elements, species, and reactions present in the gas mixture are specified by the group of variables ISC, ISS, ISR, IC, IE(I), NSC, QPJ(I), JCS(I), IS(I), and IR(I). If IGAS=0, these variables are obtained from the input data. If IGAS is positive, they are looked up in the array of precoded data, GPRP(I,IGAS), which occupies common block /MIXT/. The physical definitions of the entries in this array are given in Section 4.5 of Volume II. If IGAS is negative, the input value is saved in the location IGS, and IGAS is reset to its own absolute value. In this case, all of the above-listed variables except QPJ(I) are again looked up in GPRP, but the QPJ (the mole fractions of the cold species) are obtained from input. This provision allows input control of the overall elemental composition of the standard gas mixtures containing more than one chemical element.

After these general specifications of the gas model have been determined, the required properties of the chemical elements, species, and reactions are looked up. The data for elements are obtained from the array EPRP(I,IE), where the second subscript is the index in the master list of elements. The species data are looked up in the array SPRP(I,IS), where IS is the index in the master list of species. The reaction data are obtained from RPRP(I,IR),

where IR is the index in the master list of reactions. The entries in these arrays are all defined in Section 4 of Volume II. The order of the elements, species, and reactions in a gas model is determined by the index arrays IE(I), IS(I), IR(I), and may differ from the order in the corresponding master lists.

3.57.9 Geometric Sequence of Model Points

Positions at which model condition calculations are to be done can be determined by specifying an initial point XMØDP1, a final point CXMAXI, and a number of points NMØDPT. From these data, subroutine READ computes a factor FACMP which is used to generate a geometric sequence of NMØDPT values of x extending from XMØDP1 to CXMAXI. This factor is calculated from the formula

$$FACMP = \left[\frac{CXMAXI}{XMØDP1}\right]^{1/(NMØDPT-1)} \qquad (135)$$

3.58 Subroutine RESET

This Fortran routine, used only in the IBM version of NATA, simulates the UNIVAC 1108 timing routines RESET and TIME. The time data are obtained by calling the library routine ACUCPU, which returns in its argument ICPU the number of milliseconds of CPU time remaining before the TIME parameter of the job step is reached. Subroutine RESET is called by the main program near the beginning of each NATA job. The initial value of ICPU is stored in IZERØ. Subsequent calls to the entry TIME then give the elapsed time I in milliseconds.

3.59 Subroutine RESTMP

RESTMP performs the iterative solution for the reservoir conditions when the second or third option for input specification of these conditions is used. These options have been described in Section 6.5 of Volume I. Usage of these options is controlled by the indicator ISW2B. For ISW2B = 0, the second option is used, and for ISW2B < 0, the third option. (For ISW2B > 0, the first option is used and RESTMP is not called.)

3.59.1 Second Option

The second option is implemented by the coding down to statement 90. In this option, the input specification of the reservoir conditions is based on the reservoir pressure, p_0, and the total mass flow, $\dot{M}$. In the code, the value of p_0 in atmospheres is available in the location PRESA in unlabelled common, and the sonic mass flux m_* in $g/cm^2 sec$ is given by SMASS in common block /MASSFL/.

The basic problem in the second option is to determine the reservoir temperature, T_0, from these data. The code capabilities available to support this effort are

(1) Entry INTA of subroutine INGAS, which computes all of the conditions in the reservoir from PRESA = p_0 and CTAP = T_0.

(2) Subroutine NRMAX, which computes the sonic mass flux m_* for equilibrium flow, based on given reservoir conditions.

RESTMP determines T_0 by an iteration in which INTA and NRMAX are called, at each step, and T_0 is repeatedly adjusted to make the calculated sonic mass flux nearly equal to SMASS.

RESTMP starts the iteration with a standard guess T_0 = 10000°K. Subroutines INTA and NRMAX are then called to compute the reservoir conditions and sonic mass flux based on the trial value of T_0. The call to INTA must be preceded by a call to subroutine INIT to set a number of species properties which are nondimensionalized using the reservoir temperature.

The sonic mass flux computed by NRMAX is in the nondimensional form

$$SM = -\frac{\rho_* u_*}{\rho_0 u_s} = \frac{m_*}{\rho_0 \sqrt{R_0 T_0 / W_0}} \tag{136}$$

From the perfect gas law,

$$p_0 = \frac{\rho_0 R_0 T_0}{W_0} \tag{137}$$

Elimination of ρ_0 between (136) and (137) gives the relation

$$T_0 = \frac{W_0(T_0)}{R_0} \left[\frac{p_c \cdot SM(T_0)}{m_*}\right]^2 \qquad (138)$$

which is satisfied by the correct value of the reservoir temperature T_0. In (138), W_0 denotes the reservoir molecular weight (CMA), R_0 the universal gas constant in mechanical units (8.31434×10^7 erg/mole ^oK), p_c the reservoir pressure in dyne/cm^2, $SM(T_0)$ the nondimensional sonic mass flux computed by NRMAX, and m_* the sonic mass flux (SMASS, g / cm^{2}sec) based on the input total mass flow $\dot{M}$. Equation (138) may be rewritten

$$T_0 = \frac{W_0(T_0)}{C} \left[SM(T_0)\right]^2 \equiv \frac{F(T_0)}{C} \qquad (139)$$

where the parameter C (denoted by CONST in the program) is a constant of the iteration:

$$C = R_0 \left[\frac{m_*}{p_c}\right]^2 \qquad (140)$$

The second estimate of T_0 is obtained using equation (139), with W_0 and SM values based on the first trial value. The quantity SM is found to vary only weakly with T_0, while the molecular weight is monotonically decreasing function of T_0 at constant pressure. Thus, the error in the second estimate of T_0 is opposite in sign to that of the first estimate; the two values bracket the correct solution. Accordingly, after INIT, INTA, and NRMAX have been called, the third estimate of T_0 is obtained by a linear interpolation technique. The function $F(T_0)$ is approximated as a linear function between $T_0 = T_1$ and $T_0 = T_2$ (the first two estimates). Then

$$\frac{F - F_2}{T_0 - T_2} = \frac{F_2 - F_1}{T_2 - T_1} \qquad (141)$$

Elimination of F between (139) and (141) gives

$$T_0 = -\frac{T_2F_1 - T_1F_2}{T_2 - T_1} \cdot \frac{1}{\frac{F_2 - F_1}{T_2 - T_1} - C} \qquad (142)$$

for the third estimate.

Beginning with the fourth estimate, the dependence of $F(T_0)$ upon T_0 is approximated by a quadratic function based upon the data from the three most recent iterations:

$$F = D_1 + D_2\, T_0 + D_3\, T_0^2 \qquad (143)$$

Substitution of the data from the three points gives a system of linear equations for D_1, D_2, D_3. This system can be written in matrix form

$$A \cdot D = B \qquad (144)$$

where

$$A = \begin{bmatrix} 1. & T_1 & T_1^2 \\ 1. & T_2 & T_2^2 \\ 1. & T_3 & T_3^2 \end{bmatrix} \qquad (145a)$$

$$B = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \qquad (145b)$$

This system of equations is solved using subroutine SIMQ, which places the solution vector D into the locations originally occupied by the vector of constants B. Then elimination of F between (139) and (143) gives

$$T_0 = C\,(D_1 + D_2\, T_0 + D_3\, T_0^2) \qquad (146)$$

The solution of this quadratic equation for T_0 is

$$T_0 = \frac{1}{2D_3}\left[- (D_2 - \frac{1}{C}) \pm\sqrt{(D_2 - \frac{1}{C})^2 - 4\ D_1D_3}\ \right] \qquad (147)$$

The sign before the radical is determined by noting that, for $D_3 \longrightarrow 0$, the solution must approach

$$T_0 = - \frac{D_1}{D_2 - 1/C} \qquad (D_3 = 0) \qquad (148)$$

Thus, the sign represented by $\pm$ must be opposite to that of $-(D_2 - 1/C)$, i.e., the same as that of $(D_2 - 1/C)$.

In the program, the temperature values T_1, T_2, T_3 are stored in the array CTSAVE and the parameter values F_1, F_2, F_3 in the array F. If SIMQ returns an error indicator KS $\neq 0$, which will occur if the matrix (145a) is singular, RESTMP reverts to the linear interpolation technique. The convergence criterion for the iteration is that the relative change in T_0 between two successive iterative steps be smaller than or equal to 10^{-4}.

3.59.2 Third Option

The coding of the third option begins with statement 90 of RESTMP and continues to the end of the subroutine. In this option, the reservoir conditions are specified by input of the total mass flow $\dot{M}$ and the stagnation enthalpy h_0. The latter quantity is given, in calories per gram, by the variable HS in common /RDMAIN/. The third option requires a double iteration to determine both the reservoir temperature T_0 and the reservoir pressure p_0 from the available data.

To start the iteration, an initial estimate of the reservoir pressure is obtained from an approximate sonic flow analysis for a perfect gas. For a gas with constant specific heats,

$$\rho_* = \left[\frac{2}{\gamma + 1}\right]^{1/(\gamma - 1)} \rho_0 \qquad (149a)$$

$$u_* = a_* = \sqrt{\frac{2}{\gamma + 1}}\ a_0 \qquad (149b)$$

$$a_0 = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\frac{\gamma R_0 T_0}{W_0}} = \sqrt{(\gamma - 1)\ h_0} \qquad (149c)$$

These relations are derived, for example, in Liepmann and Roshko (ref. 15, pp. 51-54). Combination of equations (149) and (137) gives

$$p_0 = \frac{\gamma - 1}{\gamma} \sqrt{\frac{\gamma + 1}{2(\gamma - 1)}} \left[\frac{\gamma + 1}{2}\right]^{1/(\gamma - 1)} m_* \sqrt{h_0} \qquad (150)$$

In RESTMP, an initial estimate of p_0 is obtained from equation (150) using a standard estimate of the specific heat ratio, $\gamma = 4/3$. This value gives

$$p_0 = 0.743\ m_*\ \sqrt{h_0} \qquad (151)$$

in which all quantities are assumed to be expressed in absolute cgs units. With p_0 in atmospheres and h_0 in cal/gm, this relation becomes

$$p_0(\text{atm}) = 4.8 \times 10^{-3}\ m_*\ \sqrt{h_0\ (\text{cal/g})} \qquad (152)$$

which is the form used in RESTMP.

The iteration scheme used to solve for the reservoir temperature and pressure is illustrated schematically in figure 25. The starting point for the iteration (labelled "1") lies at the pressure (152) and an initial estimate of temperature, 10,000°K. In the first stage of the double iteration, the temperature is varied at constant pressure to make the reservoir enthalpy equal to the input value HS (denoted by h_s in the figure). The coding of this stage runs from statement 110 down to (but not including) 130. At each step of this iteration on temperature, INIT and INTA are called, and the reservoir enthalpy is computed from CHA using equation I(241). In the first step, the temperature

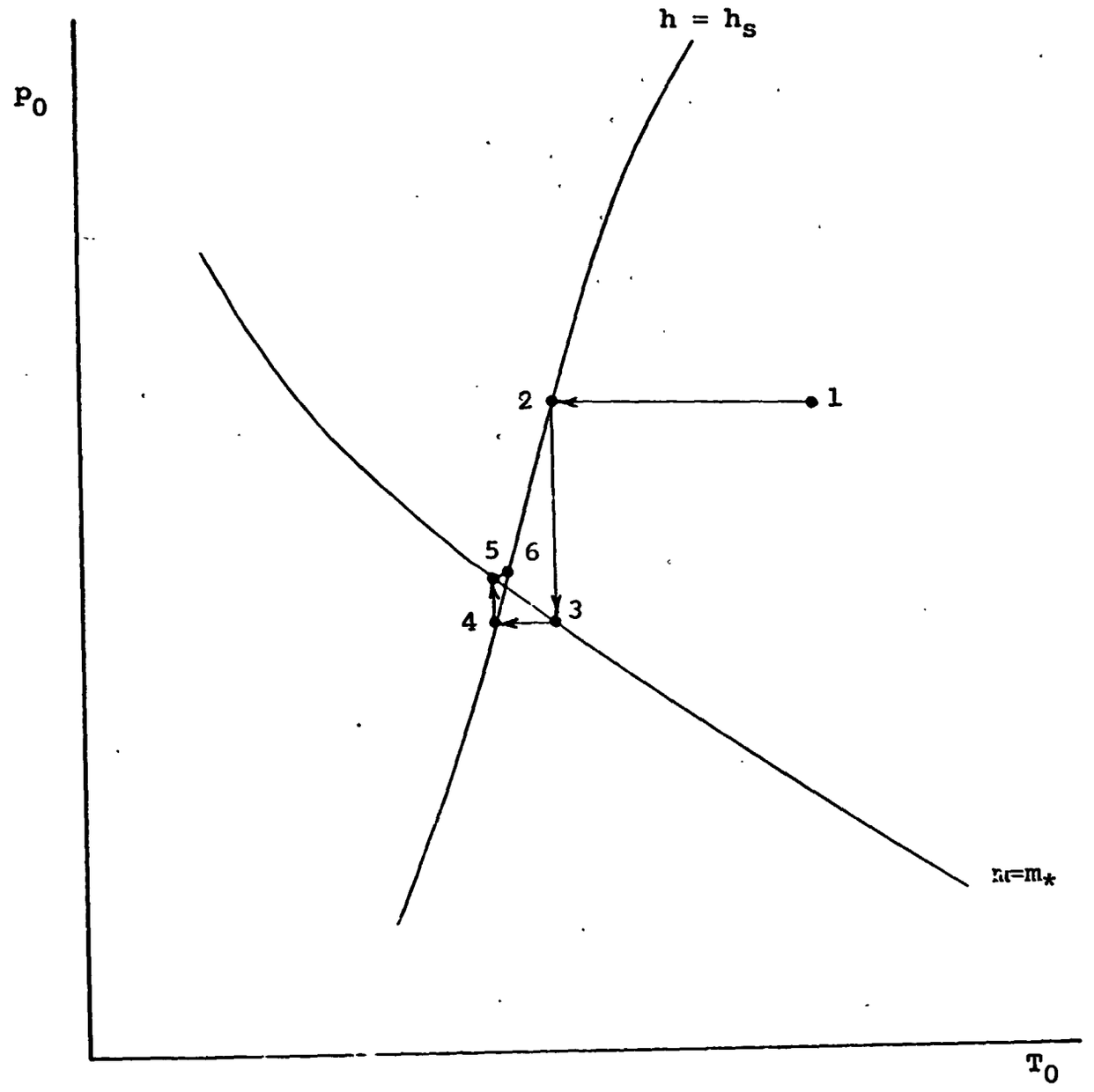


Figure 25. Iteration Scheme of Third Option for Specifying Reservoir Conditions

estimate is improved by assuming that the enthalpy is approximately proportional to the temperature:

$$T_0 = T_1 \cdot \frac{h_s}{h_1} \qquad (153)$$

where T_1 is the initial estimate of T_0 and h_1 is the specific enthalpy corresponding to T_1. In subsequent steps, the relation between T and H is approximated by a linear relation which does not necessarily pass through the origin:

$$T_0 = T + (h_s - h)\ (T_p - T)/\ (h_p - h) \qquad (154)$$

This iteration on temperature is continued until the calculated enthalpy is equal to the desired value to within 1 part in 10^5. The result of this stage is the point "2" in figure 25.

In the second stage, the reservoir pressure is varied at constant reservoir temperature to make the sonic mass flux equal to its desired value, SMASS. This stage is implemented by the coding from statement 130 down to (but not including) 150. In each cycle of this stage of the iteration, subroutine NRMAX is called to compute the non-dimensional sonic mass flux, SM. The sonic mass flux m_* in g/cm^2sec is calculated from this value using equation (136). The estimate of the reservoir pressure is then improved on the assumption that m_* is directly proportional to p_0, and subroutine INTA is called to determine the reservoir conditions based on the new p_0 value. This stage is continued until the calculated sonic mass flux agrees with the desired value to within 1 part in 10^5. The resulting point is "3" in figure 25.

The complete iteration consists of alternate temperature-adjusting and pressure-adjusting stages, and produces a series of points "4", "5", "6", etc., which spiral in towards the desired solution point in $T_0 - p_0$ space (figure 25). After each pair of stages (2,3), (4,5), etc., tests on temperature and pressure are applied to assess the convergence of the entire procedure. The criteria for convergence are that the relative changes in T_0 and p_0 during a stage-pair be less than or equal to 10^{-4}.

If the input control parameter ISW A is nonzero, a dump is written in each cycle of this iteration to allow tracing of the subroutine's operation.

3.60 Subroutine RNKT

RNKT is the routine that computes the changes in the dependent flow variables over an integration step using the Treanor-Runge-Kutta scheme explained in Section 7.5.1 of Volume I. Figure 26 is a flowchart of RNKT.

In the initializations, IFAIL is an indicator for the cause of step failure, used in diagnostic output which is produced when ISW5B is set to a negative value in the input (see glossary of Fortran symbols for common block /TNERK/ in Section 4) TE is a logical indicator. A .TRUE. value means that a chemical nonequilbrium model is in use. If TE = .FALSE., an electronic equilibrium model is being used. LIM denotes the number of dependent variables included in the numerical integration. For a chemical nonequilibrium model, LIM = n + 1 and the variables are γ_j for j = 1 to n and CT = T/T_0. For an electronic nonequilibrium model, LIM = n + 3 and the variables are γ_j, CT, CTE T_e/T_0, and CHA = $h_0 W_0/R_0T_0$. The energy transfer to the electron gas (QDPE) at the start of the integration step is denoted by QDPEB. Finally, DQMRK is the maximum allowable change in QDPE from QDPEB during the integration step.

For ISW5B different from zero, diagnostic dumps are produced at five locations in RNKT. The coding used to generate these dumps is omitted from the flowchart, as it is not essential to the function of the subroutine.

The computations in RNKT closely follow the analysis given in Section 7.5.1 of Volume I. The values y_1, y_2, y_3, y_4 are represented by arrays GJ1(J), GJ2(J), GJ3(J), GJ4(J), where J runs from 1 to LIM. Thus, these arrays contain not only the species concentrations but also the other dependent variables, CT, etc. The derivatives f_1, f_2, f_3 are represented by arrays F1(J), F2(J), F3(J). The parameter $(-P\Delta x)$ in I(398) is represented by an array P(J). The quantities F_1, F_2, F_3 defined by equations I(398) are represented by X1(J), X2(J), X3(J).

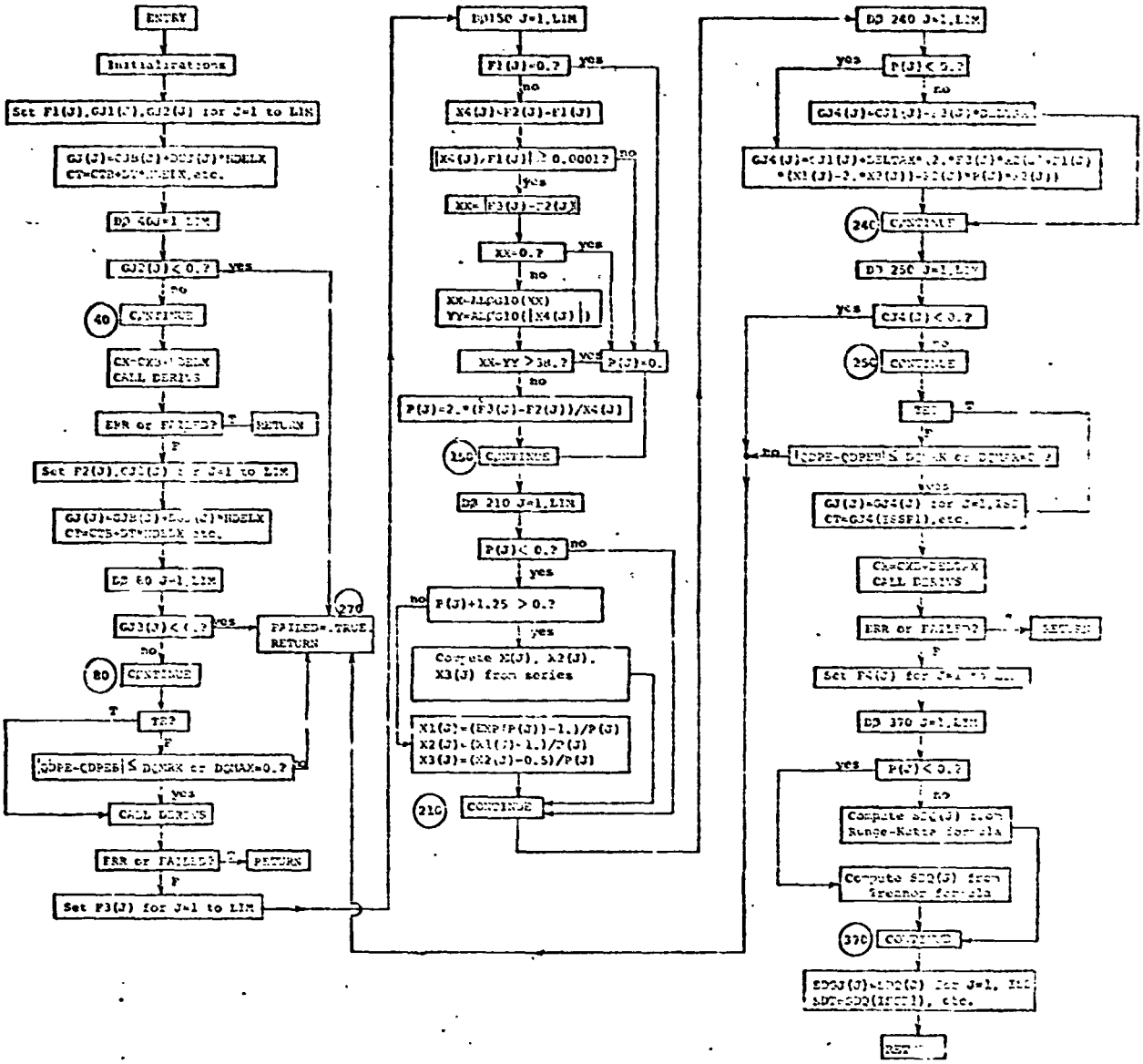


Figure 26. Flowchart of Subroutine RNKT

Following the initializations, RNKT loads data into the F1, GJ1, and GJ2 arrays. If any of the GJ2 values is negative, FAILED is set to .TRUE. and the RETURN is executed. If not, CX is set equal to CXB + HDELX, where HDELX is one-half of the step size DELTAX. Then DERIVS is called to compute the derivatives F2(J), and the GJ3 array is set. The signs of all the GJ3(J) values are tested. In addition, if TE is .FALSE., the new value of the energy transfer QDPE to the electron gas, produced by the call to DERIVS, is tested to determine whether |QDPE-QDPEB| is less than or equal to DQMRK. If not, the error return is executed unless DQMAX = 0 (which means that the current step is the first step in the numerical integration).

If the solution thus far passes these validity checks, DERIVS is called again to determine the derivatives F3(J). Next, the parameters P(J) are calculated. The basic formula for P(J) is I(402). In the Fortran notation, with $P(J) \longrightarrow -P\Delta x$ in I(402), this equation becomes

$$P(J) = 2 \cdot \frac{F3(J) - F2(J)}{F2(J) - F1(J)} \qquad (155)$$

However, under certain conditions, P(J) is set to zero. As explained in Section 7.5.1 of Volume I, this is done if $(f_2-f_1)/f_1$ = X4(J)/F1(J) has a magnitude smaller than 10^{-4}, to avoid excessive loss of accuracy on subtractions. P(J) is also set to zero if F1(J) = 0, to avoid a divide check error in the test just described, or if the evaluation of (155) would lead to a floating point overflow on the indicated division. For $P(J) \geq 0$, the changes of the dependent variables are computed using the Runge-Kutta formula I(392) in place of Treanor's formula I(401).

Once the P(J) have been determined, RNKT proceeds to compute X1(J), X2(J), and X3(J). If $P(J) \leq -1.25$, these quantities are calculated from equations I(398). Since the Fortran P(J) corresponds to $(-P\Delta x)$ in equations I(398), these formulas can be rewritten in the form

$$F_1 \equiv X1(J) = \frac{e^{P(J)} - 1}{P(J)} \qquad (156a)$$

$$F_2 \equiv X2(J) = \frac{X1(J) - 1}{P(J)} \qquad (156b)$$

$$F_3 \equiv X3(J) = \frac{X2(J) - \frac{1}{2}}{P(J)} \qquad (156c)$$

which is used in RNKT. If $P(J) > -1.25$ (but negative), the subtractions in (156) could cause excessive loss of accuracy. In this case, F1, F2, and F3 are evaluated from the power-series expansion

$$F_n = \sum_{k=0}^{\infty} \frac{[P(J)]^k}{(n+k)!} \qquad (157)$$

which can easily be derived by substituting the Maclaurin expansion for e^x into (156). In RNKT, the sums (157) for F1, F2, F3 are accumulated in the locations S1, S2, S3. The general terms in the three expansions are denoted by T1, T2, and T3, respectively. These terms are initialized to their values for k = 0, namely T1 = 1, T2 = 1/2, T3 = 1/6. Then the successive terms in the series are generated recursively:

$$(T_n)_k = (T_n)_{k-1} \cdot \frac{P(J)}{n+k} \qquad (158)$$

where $(T_n)_k$ denotes the k^{th} term in the expansion for F_n. In RNKT, the denominator (n+k) in (158) is represented by (n+Z-3), where Z is initialized to 3 for k = 0 and is incremented by 1 each time around the iterative loop. The

evaluation of (157) proceeds until the addition of T3 to S3 no longer affects the value of S3 to within floating-point accuracy.

Next, the GJ4(J) array is set. For each J, if P(J) is negative, GJ4(J) is computed from equation I(405). For zero or positive P(J), GJ4(J) is calculated from equation I(394c). After the usual validity checks, the flow variables GJ(J), CT, etc., are set to their corresponding values in GJ4(J), CX is set to CXB + DELTAX, and a final call to DERIVS is executed. The resulting values for the derivatives of the flow variable are loaded into F4(J), and the array SDQ(J) of changes in the dependent variables over the whole integration step is computed. For each J, if $P(J) \geq 0$, the Runge-Kutta formula I(392) is used to calculate SDQ(J). If $P(J) < 0$, then Treanor's formula I(401) is used. Finally, the results are transferred from the SDQ(J) array into the storage locations SDGJ(J), SDT, etc., used by other parts of the program.

3.61 Subroutine SHØCK

Subroutine SHØCK* computes the oblique shock angle and the conditions behind the shock for supersonic flow of a perfect gas over a wedge. The first three of the arguments are inputs to the routine. BMF denotes the free-stream Mach number M, DEL the flow deflection angle (i.e., the angle of attack of the wedge surface, α), and GAM the specific heat ratio γ. The outputs are FL and IERR, which are defined in Section 4.128 and by comment cards in the subroutine listing.

The method of solution is based upon the relation between the shock angle σ and the deflection angle α, which is given, for example, in Liepmann and Roshko (ref. 15, p.87). This relation can be expressed as a cubic equation in $\sin^2\sigma$:

---

*This subroutine was programmed by Harvey Buss, Avco Systems Division. Statements for calculating quantities not required in NATA have been converted into "comments" to save storage.

$$\sin^6\sigma + b\sin^4\sigma + c\sin^2\sigma + d = 0 \qquad (159)$$

where

$$b = -\frac{M^2+2}{M^2} - \gamma\sin^2\alpha \qquad (160a)$$

$$c = \frac{2M^2+1}{M^4} + \left[\frac{(\gamma+1)^2}{4} + \frac{\gamma-1}{M^2}\right]\sin^2\alpha \qquad (160b)$$

$$d = -\frac{\cos^2\alpha}{M^4} \qquad (160c)$$

Equation (159) has three roots for $\sin^2\sigma$, in general, of which the lowest corresponds to a decrease in entropy and thus to a physically impossible flow. The lower of the two remaining roots gives the desired weak-shock solution, while the uppermost root gives a strong-shock solution (e.g., a normal shock with $\sigma = \pi/2$ for $\alpha = 0$). Subroutine SHØCK first obtains the lowest of the three roots by solving equation (159) using Newton's method. With $u = \sin^2\sigma$ and

$$F(u) = u^3 + bu^2 + cu + d \qquad (161a)$$

$$F'(u) = 3u^2 + 2bu + c \qquad (161b)$$

an improved estimate u' of the root is obtained, at each step of the iteration, using the formula

$$u' = u - \frac{F(u)}{F'(u)} \qquad (162)$$

The convergence criterion is that u' = u to within 1 part in 10^8. If convergence is not achieved within 50 iterations, an error message is written and the RETURN is executed.

Use of an initial estimate u = 0 ensures convergence to the lowest root. Once this root u_1 has been obtained, the desired weak-shock solution is obtained by factoring the equation F(u) = 0:

$$(u-u_1)\ (u^2 + Au + 3) = C \qquad (163)$$

Equating the coefficients of corresponding powers of u in (161a) and (163) gives

$$A = u_1 + b \qquad (164a)$$

$$B = -d/u_1 \qquad (164b)$$

The two remaining roots of the cubic are then obtained by equating the quadratic factor in (163) to zero:

$$u = \tfrac{1}{2}\left[- A \pm \sqrt{A^2 - 4B}\ \right] \qquad (165)$$

If the flow-deflection angle α is too large, the argument of the square root in (165) becomes negative and equation (159) has no physically valid solution. In this case, the error indicator IERR is set to 0 and the RETURN is executed. When $A^2 - 4B$ is positive, the lower of the two values (165), u_2, is equal to $\sin^2\sigma$ for the weak shock.

After the shock angle σ has thus been determined, subroutine SHØCK calculates the conditions behind the shock using perfect gas relations (ref. 15, p. 86). The only result required for use in NATA is the shock pressure ratio,

$$\frac{p_2}{p_1} = \frac{2\gamma M^2 \sin^2 \sigma - (\gamma - 1)}{\gamma + 1} \qquad (166)$$

3.62 Subroutine SIMQ

SIMQ is a standard subroutine for solving systems of simultaneous linear equations. It was obtained from the IBM Scientific Subroutine Package (ref. 7), and was modified slightly for use in NATA. The modifications consisted of converting it into a double precision routine and the addition of simple coding to allow SIMQ to handle the special case of one equation in one unknown. The only available documentation of SIMQ is that provided by comment cards in the source program.

3.63 Subroutine STUNTS

This routine performs two separate and unrelated operations. When called through the principal entry (STUNTS), it produces tables of species thermal properties as functions of temperature for the species in the specified gas model. When called through the entry STUNT2, it gives an edit of the transport property calculations, and optionally, a table and a punched deck containing averaged species cross section data.

3.63.1 Thermal Property Edit

The thermal property edit is produced by the DØ loop ending at statement 110. For each species J, the properties are computed at a preset series of temperatures TT(I) up to the input value of the reservoir temperature CTAP. These temperatures are as follows:

From 100°K to 1000°K at intervals of 100°K
From 1000°K to 3000°K at intervals of 200°K
From 3000°K to 10000°K at intervals of 500°K
From 10000°K to 20000°K at intervals of 1000°K
From 20000°K to 50000°K at intervals of 2000°K
From 50000°K to 125000°K at intervals of 5000°K

The maximum number of temperatures is 74. (However, the thermo fits for the molecular species in the compiled-in gas models are useful only up to about 20000°K.)

The thermal properties are calculated by calling subroutine THERM for each temperature. THERM is programmed to calculate the properties using two different methods, the physical model and the thermo fit (Section 2.2 of Vol. I, ref. 1). The physical model is used for temperatures more than 500°K below the switchover temperature, CTMXX·CTAP. The thermo fit is used for temperatures more than 500°K above the switchover temperature. In a range 1000°K wide centered at the switchover temperature, a linear combination of the results from the two methods is used, to provide a continuous transition.

To obtain thermal properties based solely on the physical model, rather than on a mixture of the two methods, subroutine STUNTS first sets CTMXX to 1000, and calls THERM for each temperature. With this large value of CTMXX, all of the temperatures are more than 500°K below CTMXX times the reservoir temperature. The results of the thermal property calculations based on the physical model are stored in the array TPRØP(I,K,M) for M = 1. In the case of a nonstandard species for which physical model data have not been provided (ETAJ(J) = 0), the elements of TPRØP for M = 1 are all set to zero and the calculations based on the physical model are skipped.

For each species, if thermo fit data have been provided (IGJ(J) ≠ 0), CTMXX is set to -1000 and THERM is called again for each temperature up to the reservoir temperature, CTAP. With this negative value of CTMXX, all of the temperatures are more than 500°K above CTMXX·CTAP, so that the calculations in THERM are based solely on the thermo fit. The results are stored in TPRØP(I,K,M) for M = 2.

Finally, all of the results for the species are printed out. The properties computed and printed are

$TPRØP(I,1,M) = \frac{\mu_j^o - H_{j0}^o}{R_0 T}$	(chemical potential in excess of the formation enthalpy, divided by R_0T)
$TPRØP(I,2,M) = H_j - H_{j0}^o$	(molar enthalpy in excess of the formation enthalpy, kcal/mole)
$TPRØP(I,3,M) = C_{pj}$	(molar heat capacity, cal/mole^{o}K)
$TPRØP(I,4,M) = S_j^o$	(molar entropy, cal/mole^{o}K)

Examples of the thermal property edit are shown in figures 19 to 46 of Volume II (ref. 2).

3.63.2 Transport Property Edit

The transport property edit consists of three parts, the first two of which are always obtained when STUNT2 is called. These parts are (1) an edit of the steps in the transport cross section calculation as specified in the input or in the precoded data; (2) an edit of the "edited" cross section calculations after unneeded steps have been deleted, and needed but unspecified steps have been added on the basis of default options; (3) an edit of the averaged pair cross sections as functions of temperature.

(1) Specified steps in the cross-section calculation.-- This first part of the transport property edit is implemented by the DØ loop ending at statement 150. The index L runs over the steps of the cross section calculation in the order in which they are to be performed. The sequencing array ISEQ(L) selects the steps from the input or precoded data; i.e., the step which is carried out first is the one indexed I = ISEQ(1), etc. NNQI is the number of species pairs to which the cross section calculated in the L^{th} step will be applied. The indices of the species belonging to such a species pair are IIQJ, JJQJ (referred to the

master list of species).* For each step, STUNT2 prints out L, I, the index KKQ(I) specifying the type of cross section formula used in the step, the parameter values VV(K,I), and the species names for the first pair of species to which the results of the step are applied. The names of species for the remaining species pairs (if any) are then printed on successive lines.

(2) Steps in the edited cross section calculation.-- The DØ loop ending at statement 190 prints out a description of the steps in the "edited" cross section calculation which will actually be used in the transport property calculations for the current case. This "edited" calculation differs from the "input" calculation in two major ways: The steps applying only to species which are absent from the current gas model have been deleted, and steps which are required but which were not specified have been provided on the basis of default options** such as the cross section mixing rule.

The index L in the DØ loop ending at 190 runs over the steps in the edited calculation, in the order in which they are to be performed. KQ(L) is the index of the option (cross section formula) to be used in the L^{th} step. NV(KK) is the number of parameters used by this option. All of the parameters for all of the steps are stored, without gaps, in the singly dimensioned array V(I). The parameters for the L^{th} step occupy the locations from I = MV1 to MV2 in this array.

The indices of the species belonging to the species pairs to which the steps of the cross section calculation apply are stored, without gaps, in two arrays IQ(LQ), JQ(LQ). These indices IQ and JQ are referenced to the species list for the current gas model, rather than to the master list of species. The index LQ for the last species pair to which the results of the L^{th} step apply is NQ(L).

---

*See Section 4.2 of Volume II (ref. 2).

**See Section 4.6 of Volume II (ref. 2).

For each step L, STUNT2 first prints L, the option index KQ(L), and the parameter list V(I) for the option. Then it prints the names, HP(I), HP(J), for the first species pair using a carriage-control character + in column 1 to place these names on the same line. If there are any additional species pairs to which the step is applicable, their names are printed on successive lines.

(3) Averaged transport cross sections.--If ISW1B is negative, STUNT2 prints a table giving the transport cross sections for the various species pairs as functions of temperature. If ISW1B = -1, STUNT2 also punches these same data on cards for use by other computer programs. Some of the cross sections for charged species depend upon the electron partial pressure as well as the temperature. In STUNT2, the mole fractions SAVEC(I) used by the transport property routines are set to their values in the reservoir (for the current case), and are held fixed as the temperature is varied. The pressure is also held constant at its reservoir value. Thus, all of the cross sections printed are based upon the electron pressure in the reservoir.

The calculations and output are provided by the DØ loop ending at statement 230. For each species pair (I,J) with $I \leq J$, the temperature is varied from $1000^{\circ}K$ to the reservoir temperature CTAP by intervals of $1000^{\circ}K$. At each temperature, subroutine TRANSP is called. With ISW1B < 0, TRANSP does not compute the viscosity, Prandtl number, etc., but does call lower-level transport routines to set up the cross section array Q(L,I,J), where I and J are the species indices and L runs from 1 to 3. These data are printed (and punched for ISW1B = -1), together with the temperature, species indices, species names, and a counter ICARD which serializes the punched cards. Their relation to the cross sections used in transport theory is (Section 3 of Vol. I, ref. 1):

$$Q(1,I,J) = \bar{\Omega}_{ij}^{(1,1)}$$

$$Q(2,I,J) = \bar{\Omega}_{ij}^{(2,2)}$$

$$Q(3,I,J) = B_{ij}^{*} \bar{\Omega}_{ij}^{(1,1)}$$

They are expressed in units of 10^{-16} cm^2.

3.64 Subroutine THERM

THERM is the routine which computes the species thermal properties: μ_j^o (chemical potential at standard pressure), H_j (molar enthalpy), S_j^o (molar entropy at standard pressure), and C_{pj} (molar heat capacity at constant pressure). In its internal calculations, NATA uses these properties in nondimensional form; the quantities actually computed in THERM are

$$\text{XMJAT(J)} = \frac{\mu_j^o}{R_0 T} \tag{167}$$

$$\text{SHJ(J)} = \frac{H_j}{R_0 T_0} \tag{168}$$

$$\text{SENT(J)} = \frac{S_j^o}{R_0} \tag{169}$$

$$\text{CCPJ(J)} = \frac{C_{pj}}{R_0} \tag{170}$$

where J = j is the species index in the list of species for the current case. In these formulas, R_0 denotes the universal gas constant and T_0 the reservoir temperature.

When the subroutine is called through the entry THERM1, only the heat capacity is computed.

THERM provides two methods for calculating the species thermal properties: the "physical model" and the thermo fit. These techniques have been explained and discussed in Section 2.2 of Volume I (ref. 1). Data for the physical model are provided for all standard species. In the current version of NATA, thermo-fit data are provided only for the diatomic molecules and molecular ions in air and the planetary atmosphere models. The array IGJ(J) indicates whether thermo fit data are available for each species. If IGJ(J) = 0, there are no thermo fit data for the J^{th} species; in this case the physical model is used

for the species at all temperatures. If ETAJ(J) = 0, there are no physical-model data, and the thermo fit is used at all temperatures.

In the case of species for which both physical-model and thermo fit data have been provided, the physical model is used from low temperatures up to a nominal switchover temperature, CTMXXI, which is preset to 5000°K; and the thermo fit is used at higher temperatures. To prevent possible program failures due to discontinuities in computed species properties resulting from small mismatches between the physical model and thermo fit results, the switchover is spread over a temperature range 1000°K wide, centered at CTMXXI. Equations I(58) are used to mix the results from the two techniques in this temperature range.

The temperature variable used by NATA in its internal computations is CT, a nondimensional temperature defined as T/T_0. The corresponding nondimensional switchover temperature is CTMXX = CTMXXI/T_0. For species and temperatures for which the thermo fit is used, the species properties are computed from the formulas I(33), I(34), I(29), and I(31). Prior to the first call to THERM, the thermo fit coefficients TFA(J) = a, etc., have been nondimensionalized in subroutine INIT so as to allow direct use of the nondimensional temperature CT in the thermo fit formulas.

The physical model calculations are done in two steps. The first step computes the translational, rotational, and vibrational contributions to the species properties, and the second step the contributions due to electronic excitation. The first step is coded separately for monatomic and diatomic species, on the one hand, and for linear triatomic species, on the other. In the case of monatomic and diatomic species, the calculations are done using equations I(50),I(51), I(54), I(56), and I(29). For linear triatomic species, equations I(52), I(55), I(57), and I(29) are used.

If the control parameter INEQV is nonzero, the properties of molecular species are calculated on the assumption that the vibrational degrees of freedom are frozen at

the reservoir temperature. In this case, the formulas given in Section 2.4 of Volume I (ref. 1) are used to calculate the species properties.

3.65 Subroutine THRØAT

The function of THRØAT is to adjust the nozzle geometry in the downstream region so as to make the effective area ratio continuous between the upstream solution by the inverse method and the direct downstream solution. It performs this function by multiplying the rescaling factor RSA, which is used in GEØMAR, and which is initially 1.0, by the factor AG/S1. Here S1 is the geometric area ratio based on the specified nozzle geometry before the adjustment, and AG is the geometric area ratio calculated from the effective area ratio AFNX. If the boundary layer is not included in the solution; AG is equal to AFNX. If the boundary layer is included, AG is computed from AFNX by calling subroutine AGSØLN. Since boundary layer displacement effects upon the inviscid flow are neglected during the nonequilibrium solution by the inverse method, the rescaling also provides continuity of the effective area ratio when the coupling of the inviscid flow with the boundary layer is switched on following the call to THRØAT.

In addition to calculating RSA, THRØAT prints an informative message specifying the conditions at the current flow point, and resets the indicator IUPD from 1 to 0 to signify that the switch from the inverse method to the direct integration has occurred.

3.66 Subroutine TRANSP

Transport property calculations are performed in the NATA code under the general supervision of subroutine TRANSP. This subroutine contains two entries: TRANSX which is called to initialize the routines, and TRANSP(TTAB,P) which carries out the actual transport property calculations for any specified gas conditions.

The entry TRANSX is called prior to the start of actual transport property calculations, in order to inform the

routines of the species to be used in the calculations and to perform certain required initializations. The entry first sets the total number of species N=ISS to be used in the transport calculations and then calls subroutine XSECT which selects cross section data for the species to be used in the calculations from the data available in the code and arranges them in a form convenient for the subsequent computations. Finally, TRANSX calculates a number of quantities depending on the species molecular weights which will be required in the subsequent transport computations, before returning control to the calling program. The quantities calculated here are

$$ZM1(I) = \frac{128 \times 10^{-24}}{9\pi N_0 k^3} W_i = 2.8567\, W_i \frac{cm^2(^oK)^3}{milliwatts^2(\mathring{A})^4} \quad (171a)$$

$$ZM2(I) = \frac{4}{15\, N_0 k} \times 10^7 \quad W_i = 0.032064\, W_i \quad g\ ^oK/joule \quad (171b)$$

$$C(I,J) = \frac{8 \times 10^{-12}}{3k} \sqrt{\frac{2\, W_i W_j}{\pi N_0 k\, (W_i+W_j)}} \quad \text{for } I \leq J \quad (171c)$$

$$C(I,J) = \frac{W_i - W_j}{W_i + W_j} \quad \text{for } I > J \quad (171d)$$

Once the gas species have been defined by calling TRANSX, the transport properties for any given gas condition are computed by calling TRANSP(TTAB,P), where TTAB is the gas temperature and P is the gas pressure in atmospheres. The species mole fractions are obtained from the array SAVEC(J) in common block /TEMPRY/. The first step in the computation is the calculation of the molar specific heat for each species in the mixture. This calculation is carried out by calling the entry THERM1 of subroutine THERM. Input to the calculation is by means of the common variable CT which specifies the ratio between the temperature TTAB for which the specific heats are to be calculated and the reservoir temperature CTAP. The nondimensional molar heat

capacities at constant pressure $W_j c_{pj}/N_0 k$ for each species in the mixture are then returned to subroutine TRANSP in the common variable CCPJ(J). These values are saved by the routine in the array DH for later use in computing the internal thermal conductivity from equation I(95), and are also used to compute the nondimensional frozen specific heat CPTØT = $W\, c_{pf}/N_0 k$ for the mixture, which is required in the calculations of the Prandtl and Lewis numbers, equations I(83) and I(84). In calculating the specific heats, it is necessary to save the previous values of CT and CCPJ in a temporary location and then restore them for later use in other portions of the flow field calculations.

After completing the specific heat calculations, TRANSP next calls subroutine PUTQIN which computes the collision cross sections $\bar{\Omega}_{ij}^{(1,1)}$, $\bar{\Omega}_{ij}^{(2,2)}$, and $B_{ij}^{*}\bar{\Omega}_{ij}^{(1,1)}$ in units of 10^{-16} cm^2 for all pairs of species in the mixture at the specified temperature and pressure. These cross sections are returned to subroutine TRANSP in the upper half ($I \leq J$) of the common arrays Q(1,I,J), Q(2,I,J), and Q(3,I,J) respectively. If an edit of pair cross sections has been requested (ISW1B < 0), a RETURN is now executed, and subroutine STUNTS then prints the cross sections contained in the Q array. However, in a normal NATA run (ISW1B ≥ 0), subroutine TRANSP uses these values of the $\bar{\Omega}_{ij}^{(\ell,s)}$ to compute the matrix elements $A_{ij}^{(\alpha)}$, $a_{ij}^{(\alpha)}$ and $\Delta_{ij}^{(1)}$, equations I(88) to I(90), which are required in the transport property calculations, and stores them in the Q array, replacing the $\bar{\Omega}_{ij}^{(\ell,s)}$ values which were previously stored there. The statements down to 80 compute and store these quantities as follows:

$$Q(1,I,J) = \frac{1}{k}\sqrt{T}\ \Delta_{ij}^{(1)}\ \left(\frac{cm\ (^{o}K)^{3/2}}{milliwatt}\right) \quad \text{for } I \leq J \qquad (172a)$$

$$Q(1,I,J) = \frac{25}{4}\sqrt{T}\ a_{ij}^{(K)}\ \left(\frac{cm\ (^{o}K)^{3/2}}{milliwatt}\right) \quad \text{for } I > J \qquad (172b)$$

$$Q(2,I,J) = \frac{25}{8}\sqrt{T}\ ZM2(I)\ A_{ij}^{(\mu)}\ \left(\frac{cm\ (^{o}K)^{3/2}}{milliwatt}\right)$$

$$= \frac{5}{6}\frac{\sqrt{T}}{k}\ \Delta_{ij}^{(2)} \quad \text{for } I \leq J \qquad (172c)$$

$$Q(2,I,J) = \frac{25}{8}\sqrt{T}\ a_{ij}^{(\mu)}\ \left(\frac{(^{\circ}K)^{1/2}}{\text{millipoise}}\right) \quad \text{for } I > J \qquad (172d)$$

$$Q(3,I,J) = \frac{25}{8}\sqrt{T}\ A_{ij}^{(K)}\ \left(\frac{\text{cm } (^{\circ}K)^{3/2}}{\text{milliwatt}}\right) \qquad (172e)$$

In connection with these computations, it may be noted that the diagonal matrix elements $a_{ii}^{(\alpha)}$ are not required in the transport calculations, as shown by equation I(86), so that it is unnecessary to compute them here. Since the quantity (172e) is not symmetric in i and j, it requires the whole Q(3,I,J) matrix for storage, whereas the other matrix elements are all symmetric, and can be stored in half the array. The quantity $\rho c_{pf} D_{ij}/\sqrt{T}$ in milliwatts/cm $(^{\circ}K)^{3/2}$ is also computed and stored by the code as FLEWIS, where D_{ij} represents the binary coefficient I(99) between the two species ID1 and ID2 which have been specified for the calculation of the Lewis number in the code input.

The calculation of the transport properties from the matrix elements (172) is begun in the DØ loop ending at statement 90, and is then carried forward in a series of three subroutines ELCØND, KINT, and KANDMU. The quantities returned to TRANSP by these subroutines are

SIGMA = σ in mhos/cm		
ZKINT = $K_{int}/\sqrt{T}$	(milliwatts/cm-$(^{\circ}K)^{3/2}$)	
ZK(1) = $K_{tr}/\sqrt{T}$	(milliwatts/cm-$(^{\circ}K)^{3/2}$)	(173)
ZK(2) = $\mu/\sqrt{T}$	(millipoise/$(^{\circ}K)^{1/2}$)	

where σ, K_{int}, K_{tr}, and μ are respectively the electrical conductivity, the internal thermal conductivity, the translational thermal conductivity, and the viscosity of the mixture, as given by equations I(85) to I(87), I(95) and I(98). These quantities are then combined further by subroutine TRANSP according to equations I(83), I(84) and I(94) to obtain the viscosity VISC in poise, and the frozen

Prandtl number PRF and the atom-molecule Lewis number* FLEWIS. These quantities, along with the electrical conductivity SIGMA, are then returned to the calling program in the common block TRPROP. For ISW8B $\neq$ 0, these data, as well as certain other quantities of interest, are also printed out by the final section of subroutine TRANSP before returning control to the calling program.

3.67 Subroutine WEDGE

Subroutine WEDGE calculates the pressure, heat flux, and boundary layer displacement thickness on wedge models. The calculations are done using the modifications I(501), I(503), I(505) of the results of he Cheng-Kemp theory. Optionally, the calculations are also done using the unmodified Cheng-Kemp theory if the input control variable ISW9B is set to a value in the range from -1 to -3 .

WEDGE obtains the free-stream flow properties required in the calculations from the array FVØUT in common block /ØUTPUT/. The data in this array refer to the current flow point, because the free stream conditions are always printed out immediately prior to a call to subroutine MØDEL which, in turn, calls WEDGE. These data are accessed by equivalencing the variable names used in WEDGE to the corresponding FVØUT array elements.

The subroutine logic is straightforward. All quantities which are independent of the angle of attack and the wedge leading-edge radius are computed first. The remaining calculations are performed and the results are printed out in a pair of nested DØ loops over the index IR for leading edge radii and the index IA for angles of attack.

---

*The Lewis number as given by I(84) is multiplied by an integer variable IAMBIP, which is equal to 1 if the two species for which the Lewis number is calculated are both neutral, and equal to 2 if one of these species is an ion. The factor of 2 takes account of the ambipolar nature of the diffusion of ions in a neutral plasma.

For each angle-radius combination, the subroutine performs calculations for all of the specified distances from the leading edge. These distances are specified in the input in two independent ways:

(1) A uniform sequence of NWX points, with an initial value of WX1 and an increment DWX;

(2) A list of arbitrary values in ascending order.

Subroutine WEDGE folds these two sequences together into a single sequence arranged in ascending order, using logic similar to that employed in subroutine NEXTMP.

WEDGE calls subroutine SHØCK to compute the static pressure ratio across the oblique shock, denoted by APR(IA). This quantity is used in calculating the angle of attack parameter Γ (denoted by CAPGAM) from equation I(501) in the modified Cheng-Kemp theory. After Γ and ζ (ZETA) have been computed, WEDGE calls subroutine WESØLN to obtain the approximate solution of the Cheng equation. The arguments of WESØLN are double precision.

The calculations based on the modified Cheng-Kemp theory are performed for ITH = 1. When these calculations have been completed at a given model point, if ISW9B is negative, ITH is set equal to 2 and all of the calculations are done over using the unmodified theory.

The input variable ISW9B is also used to control the output of some results of secondary interest. If IS9=|ISW9B| is equal to 1, the shock ordinate YS is printed; if IS9 is equal to 2, the nondimensional coordinate ZETA is printed; and if IS9 is equal to 3, both YS and ZETA are printed.

3.68 Subroutine WESØLN

The function of subroutine WESØLN is to solve the Cheng equation I(480) for z, zz', and (zz')' when ζ and Γ are given. The solution is based on the approximation I(484). which relates the solution for general Γ to that for $\Gamma = 0$.

The latter is obtained from Cheng's analytical solution I(482), I(483) for $\zeta > 10^{-8}$. For small ζ, there is a great deal of cancellation of terms in I(482), so that many significant figures of accuracy are lost. This problem is kept under control by performing all of the arithmetic in WESØLN in double precision, and by using an analytical series solution for $\zeta \leq 10^{-8}$:

$$z = 1.65096\ \zeta^{2/3} + 0.50869\ \zeta^{5/6} - 0.0249\zeta \qquad (174a)$$

$$zz' = 1.81711\ \zeta^{1/3} + 1.25975\ \zeta^{1/2} + 0.14712\ \zeta^{2/3} \qquad (174b)$$

$$(zz')' = 0.6057\,\zeta^{-2/3} + 0.62988\ \zeta^{-1/2} + 0.09808\ \zeta^{-1/3} \qquad (174c)$$

Equation (174a) was derived by Boger and Aiello (ref. 16). Equations (174b) and (174c) can be obtained from (174a) by differentiation.

For $\zeta > 10^{-8}$, equation I(482b) is solved for λ using Newton's method. The parameter λ is denoted by XL. An initial trial value of λ is obtained from the rough analytical approximation

$$\lambda = zz' = (6\,\zeta)^{1/3} + (3\,\zeta)^{1/2} \qquad (175)$$

which has a maximum error of about 12 percent. At each step in the Newton iteration, ζ is calculated from the current trial value of λ using equation I(482b), and an improved estimate λ' of λ is obtained from

$$\lambda' = \lambda - (\zeta - \zeta_i)\frac{d\lambda}{d\zeta} \qquad (176)$$

in which ζ_i is the input value ZETA and, from I(480) with $\Gamma = 0$ and I(483),

$$\frac{d\lambda}{d\zeta} = \frac{d}{d\zeta}\left(z\frac{dz}{d\zeta}\right) = \frac{1+\sqrt{zz'}}{z} = \frac{1+\sqrt{\lambda}}{z} \qquad (177)$$

where z is computed from I(482a). The convergence criterion of the iteration is that the ζ from I(482b) equal the input value to within 0.01 percent.

Once the correct value of λ for the solution with $\Gamma = 0$ has thus been found, the three outputs of the routine are computed as follows: zz' (denoted by ZZP) is obtained from equation I(491), (zz')' (denoted by ZZPP) from I(434) and I(485), and z (denoted by Z) from I(492).

3.69 Subroutine XSECT

Subroutine XSECT provides NATA with the capability to perform transport property calculations for gas models which do not contain all of the species listed in the precoded cross section data, and in which the species may be listed in a different order. For this purpose the subroutine sets up a new set of data for the cross section computations in common block /TRANS7/ which gives the same values for the cross sections as the original input data in common block /TRANS2/, but which is in a more condensed format, and is referenced to the set of species included in the current gas model rather than to the larger set of species for which cross sections are defined in the precoded and input data. The edited data in common block /TRANS7/ are then used by the code in computing collision cross sections for the gas in the subsequent transport property calculations. The original cross section data are also retained unchanged in common block /TRANS2/, in order to allow the running of stacked cases with different sets of species.

Since several of the options which are used for computing cross sections in NATA (see Section 4.6 of Volume II, ref. 2) give values for the cross sections which depend on the values computed in previous steps of the computation, the edited cross section data in common block /TRANS7/ are set up to carry out the steps of the cross section computations in the same order in which they were specified in the

original data; however, the species references in the data are changed so that the computed cross sections will be indexed in the same way as the species used in the current gas model, rather than being indexed to the original precoded or input data. Further, cross sections for pairs of species not required in the transport calculations are omitted from the edited data, and any missing cross sections are supplied according to the default options described in Section 4.6 of Volume II. The procedures used by subroutine XSECT in preparing these edited data are described in the following paragraphs. It may be noted that several of the cross section options available in the code contain features which require special treatment in this process, so that the coding of the subroutine becomes rather lengthy, although the basic concept is simple. A schematic flowchart of the subroutine showing the major control sections is provided in figure 27.

3.69.1 Initializations

The first executable statement in subroutine XSECT sets the number of species N for which cross sections are to be computed equal to the number of species ISS in the current gas model. The next two statements set IELEC = 1 if electrons are to be considered in the transport calculations and IELEC = 0 otherwise. If electrons are included in the calculations, they must always be the first species.

The statements from "JJ = 50" down through statement 30 establish the correspondence between the species in the current gas model and the master list of species,* which is used in indexing the cross section data supplied in common block /TRANS2/. These statements use the common variable IS(L), which gives the index IS in the master species list of the L^{th} species in the current gas model (for $1 \leq L \leq ISS$), to prepare a new array I(J) which gives the position I in the current model of the J^{th} species in the master species list, for each species included in the current gas model. Species in the master species list which are not included in the current model are filled into the I(J) array in order following the species included in the current model, so that

*See Section 4.2 of Volume II (ref. 2).

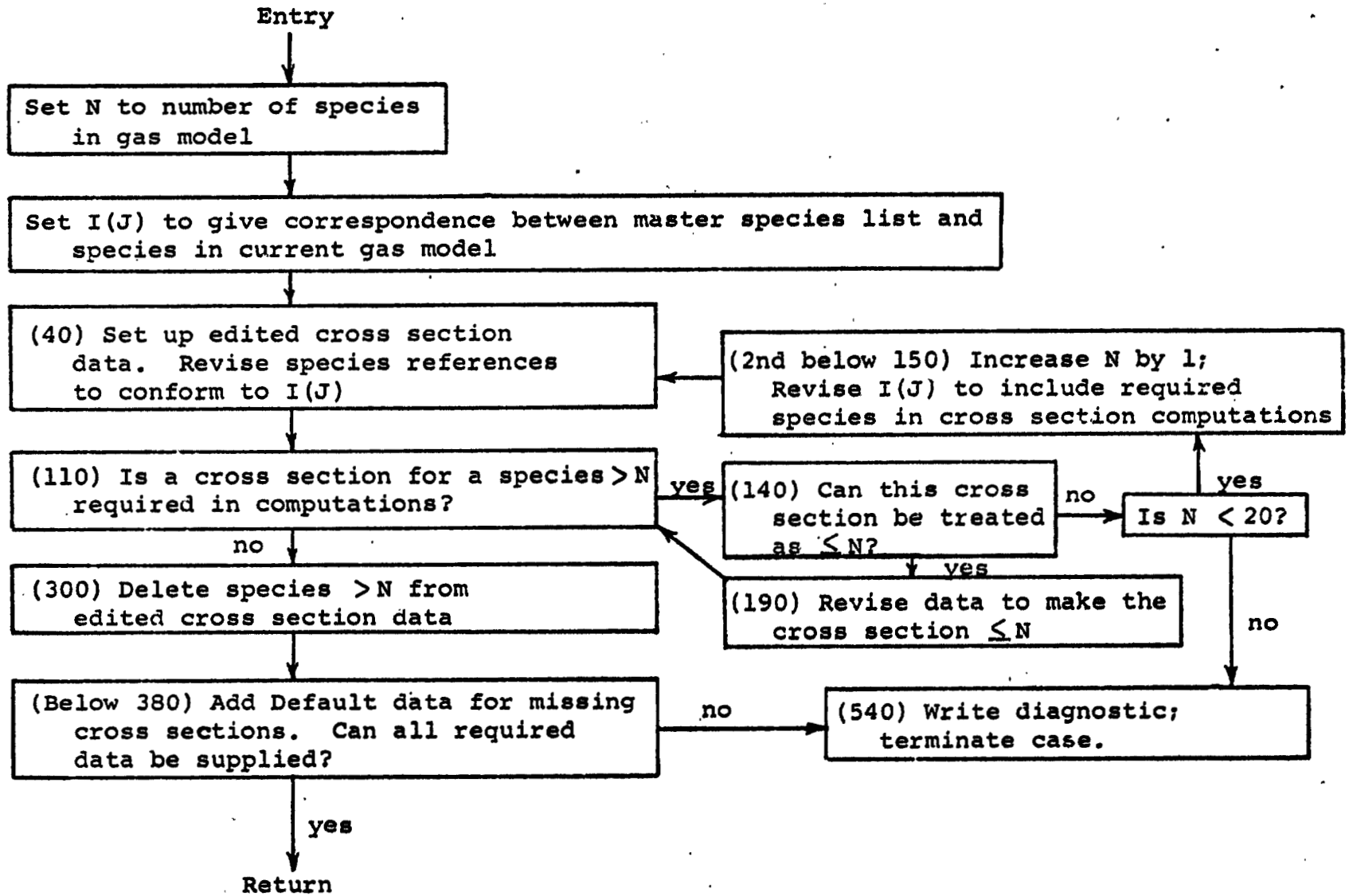


Figure 27. Schematic Flowchart of Subroutine XSECT

I(J) defines a rearrangement of the master species list such that all species J in the original master species list occur at some point I(J) in the rearranged list,* and the first ISS species in the rearranged list are just the species in the current model, in the correct order.

The three statements following statement 30 determine the indices ID1 and ID2 in the rearranged species list of the species to be used in the Lewis number calculations. They are set up so that ID1 < ID2 in all cases.

The common variable Q(1,J,L) is used as a flag in subroutine XSECT to indicate those pairs of species for which cross section data have been supplied. Statements 40 through 50 initialize this variable to 0 for all pairs of species included in the cross section calculations, and the value is then later set to 1 for each species pair for which data are provided. The value Q(1,2,1) = 1. is set initially to indicate that the pair (2,1) is used in the intermediate calculation of the cross sections in subroutine PUTQIN, and is thus not available for use in input. (Note that we are concerned here with ordered pairs (J,L). Only the pairs with $J \leq L$ are used in the actual transport property calculations,** while those with $J > L$ are available for intermediate calculations if desired.)

3.69.2 Resequencing of Steps and Revision of Species References

The coding following statement 50, down to statement 110, revises the species references in the input cross section data to conform to the rearranged species list I(J) described above and sets up a preliminary version of the edited cross section data in common block /TRANS7/ using these revised data. At this stage of the computation, the

---

*It is assumed that the master species list contains no more than 50 species; in the current version of NATA, array dimensions limit the number of species to 30.

**See the discussion of subroutine TRANSP in Section 3.66.

edited data contain the same information as the original cross section data in common block /TRANS2/, but the order of steps in the edited data is rearranged to conform to that specified by the sequencing array ISEQ, and the data on species pairs and cross section parameters for each step are close packed in the IQ, JQ, and V arrays. Further, NQ(L) in the edited data is taken to be the sum from 1 to L of the original NNQ(L) values, and the indices of all species pairs in the edited data are changed to refer to the positions of the species in the rearranged species list, I(J). Since only cross sections Q(J,K) with $J \leq K$ are used in the transport property calculations, the correspondence between species pairs used in setting up the edited cross section data must preserve this property. Accordingly, the pair (J,K) in the original data is replaced by (I(J), I(K)) if J-K and I(J)-I(K) have the same sign and by the pair (I(K), I(J)) if they have different signs. Since this correspondence is 1-1 between ordered pairs of species, it is clear that when every species pair in the original cross section data is replaced by its corresponding pair in the edited data, the calculated cross section values will all be the same as before, but will be rearranged in the order required for the transport property calculations.

The above correspondence between species pairs is carried out in the NATA code by subroutine CXSECT, which returns the revised indices of a species pair in its argument list when it is called with the original indices. In the coding down through statement 80, this correspondence is used to set up the IQ, JQ list of species pairs for the revised cross section computations. This section of coding also sets Q(1,L,J) = 1. for those species pairs which are included in the IQ, JQ list.

For the cross section options KQ = 9 and 12, there are also references to species pairs in the V parameters of the option which must be revised. This revision is carried out by the statements following statement 80 down through the statement before 110. Subroutine BXSECT searches through the steps L of the cross section computation in common block /TRANS7/, starting with the first step after the value of L specified in its argument, and looking for

a step with KQ(L) = 9 or 12. When such a step is found, the subroutine returns the index of the step L, the index MMV + 1 of the first V parameter for the step, and the indices LL and J of the species pair which is referenced in the V parameters of the step. Subroutine XSECT then calls CXSECT to revise the indices for the pair and goes on with the search by calling BXSECT again. This process is continued until the final step L = NKQ of the computations is reached; at this point BXSECT sets L to 0 as a signal that the search has been completed and control passes to the next section of code (statement 110). In addition to revising the indexing of the species pairs, this section of code also sets MMV, for later use, equal to the total number of V parameters used in the cross section computations.

3.69.3 Deletion of Unused Steps

When control reaches statement 110 of subroutine XSECT, the rearrangement of the species in the cross section data to conform to the order in the current gas model has been completed. The following section of coding, from statement 110 through 380, is now concerned with increasing the efficiency of the cross section computations by omitting unnecessary steps in the computation. In this procedure, the statements from 110 to 300 determine which cross sections are to be omitted from the computations, while those from 300 through 380 carry out the actual editing of the cross section data to delete these values.

As noted previously, the number of species N to be included in the cross section computations is set equal to the number of species in the current gas model at the beginning of subroutine XSECT. Thus only cross sections between pairs of species with indices $I \leq N$ in the revised cross section data are required in the transport property calculations. Ordinarily cross sections involving a species with index $> N$ can be omitted from the cross section computations without changing the values of the computed cross sections for the remaining species with indices $\leq N$; however, for the options KQ = 9 and 12 used in the cross section computations, it may happen that the cross section values

for some pair of species with indices ≤ N are to be computed from values obtained previously for some other species pair involving a species with index > N. Statements 110 to 140 check for this possibility, again using subroutine BXSECT to search for those steps in the computation with KQ = 9 or 12. (Note that the coding here assumes that L = 0 when statement 110 is reached. This is always true, since statement 110 can only be reached from the IF statements following statements 100 and 290.) If this search does not reveal any pair of species with an index > N which affects the cross section computations for species with indices ≤ N, then it is possible to omit all species with indices > N from the cross section computations without affecting the values of the calculated transport properties, and control passes at once to statement 300. If, on the other hand, some species pair (LL,J) with an index greater than N is found which affects the computations for indices ≤ N, then the search for further such pairs is temporarily discontinued and the code immediately begins looking for a way to include the pair (LL,J) in the computations. The first approach considered is to treat the cross sections for the pair (LL,J) as an intermediate step in the computation of the cross sections for the species with indices ≤ N, and to fill the computed cross sections for this pair into some unused location (II,JJ) of the cross section array for species ≤ N, where JJ < II ≤ N.* Statements 140 to 150 look for such a location, using the previously set array Q(1,II,JJ) as a flag to indicate which locations are available.

If a suitable location for intermediate storage of the pair (LL,J) is not found in statements 140 through 150, then the following section of code down to statement 190 increases the number N of species to be included in the cross section computations by one, rearranges the list of species I(J) for the edited cross section data so that the species corresponding to the maximum index of the pair (LL,J) will be the new species included in the cross section computations (note that by definition this species was not included

---

*Remember that only species with II ≤ JJ are used in the transport calculations.

in the computations previously), and then returns control to statement 40 to begin the preparation of the edited cross section data again from the beginning, using the new arrangement of species I(J). If this procedure would lead to a value of N > 20, however, so that the dimensions allowed for the cross section computations in the code would be exceeded, it is not carried out and control passes instead to statement 540 where a diagnostic message is printed and the error indicator ERR is set to terminate the case. If more than 50 species were specified in the master species list, the case may also be terminated by reaching the GØ TØ 540 statement after statement 180.

When the statements from 140 through 150 are successful in finding an unused location (II,JJ) which can be used for storing the cross sections of the species pair (LL,J), control passes to statement 190, where the value of Q(1,II,JJ) is immediately set to 1.0 to indicate that the pair (II,JJ) is now being used in the cross section computations. The following statements down to statement 300 then revise the edited cross section data in common block /TRANS7/ to change all references to the pair (LL,J) to (II,JJ). To carry out this revision without changing the values of the computed cross sections, any steps in the computation which use the pair (LL,J) in the empirical mixing rule KQ = 10 must also be converted in the revision process to use the generalized mixing rule KQ = 12. This conversion is carried out by the coding down through the statement NKQ=NSV after statement 270. The code first searches through all steps L of the cross section computations to find any steps using the option KQ = 10 which refer to the pair (LL,J). This may occur in two ways: either the species pair (LL,J) may be a non-diagonal pair (i.e., LL ≠ J) which is computed with the option KQ = 10 at some step of the computations, or it may be a diagonal pair (i.e., LL = J) which is used in the option KQ = 10 to compute the cross sections for some non-diagonal pair involving the species LL. Both of these possibilities are checked for by the IF statement below statement 200, and if either is found the following statements then revise the edited cross section data in common block /TRANS7/ to convert the computation for the given species pair to the option KQ = 12. This is accomplished

by inserting the required data for the KQ = 12 option into the KQ, NQ, IQ, JQ and V arrays just ahead of the data for the step L which originally computed the cross section, and moving the remaining data in these arrays down to make room for the new data. The original option KQ = 10 is retained in the data immediately following the new KQ = 12 option, but is no longer applied to that species pair for which the KQ=12 option is being used. The values of MMV, NKQ and L are also adjusted to agree with the revised data. In these computations, subroutine BXSECT is used to locate the position MM in the V array at which the new data are to be inserted. This is done by replacing NKQ by L in common block /TRANS7/, so as to cause BXSECT to end its search of the KQ array, when step L is reached. The true value of NKQ is saved in the variable NSV throughout this section of the code and is restored to its proper location at the end of the section.

After the required conversions from KQ = 10 to KQ = 12 have been made, control passes to the section of code between card XSE 213 and statement 300, which changes all references to pair (LL,J) in the IQ,JQ, and V arrays to refer to the pair (II,JJ) instead, using the same algorithms as were used previously in rearranging the species in statements 40 through 110. When these changes have been completed, control returns again to statement 110 and the code again searches the edited cross section data for a pair of species with an index > N which affects the cross section computations for species with indices ≤ N. If such a pair is found, it is eliminated from the edited cross section data as described above, and the whole process is repeated until either all species pairs with indices > N which affect the cross sections for species ≤ N have been eliminated from the data and control passes to statement 300, or else the procedures used in the code are unable to eliminate a species pair and cause the case to be terminated as described above.

By the time control reaches statement 300 in subroutine XSECT, the edited cross section data have been put into such a form that all cross section computations involving species with indices greater than N can be omitted

without changing the values of the computed cross sections for the species with indices ≤ N. The revision of the data to omit these computations is carried out by statements 300 to 380. The pairs of species for which cross sections are computed at each step of the computations are given in the IQ and JQ arrays, sothat the steps to be deleted from the computations can be determined simply by searching the IQ, JQ list for pairs in which at least one of the indices IQ or JQ is > N. This search is performed by the preliminary loop ending at statement 310, and the value of IQ for each pair which is to be deleted is set to 0 as a signal for the subsequent calculations.

To determine the required revision of the KQ, NQ, and V arrays in the edited data, it is necessary to establish the correspondence between these arrays and the IQ, JQ arrays. This is done and the arrays are revised in the loop over the steps L of the cross section computation which ends at statement 370. For each step L of the computations, the small loop ending at statement 320 determines the correspondence to the IQ, JQ list and calculates the total number of species pairs NEWLQ which are not deleted up through step L. This number is the new value of NQ for the step.

The second statement below 330 is the test to determine whether the current step L of the computations will be omitted from the revised data. The test compares the current value of NEWLQ with the value for the previous step; if these two values are equal, it indicates that the current step of the computations will not apply to any species pairs in the IQ, JQ list after the deletions are made, so that the step can ordinarily be omitted from the revised data without affecting the computed cross sections. However, for the option, KQ = 10 the step is not omitted, even though it does not apply to any species, since it may be required subsequently in supplying the default data for the cross sections, as described below.

The edited cross section data for those steps which are not to be omitted from the computations are placed in their proper locations in the revised KQ, NQ, and V arrays by the coding from the third statement below 330 down to the

statement before 360. Since the revised arrays in this stage of the calculations are always shorter than, or at most equal to, their initial length, this revision can be carried out entirely within the arrays themselves, without disturbing subsequent steps in the revision. The indices L and MV here indicate the positions of the data in the initial arrays, while NEWL and NEWMV are their positions after the desired omissions have been made. It may be noted further in connection with this calculation that no data need be added to the revised V array for KQ = 10, since this option requires no values for V in its operation. Also the number of V parameters NV(K) required by each option is obtained from preset data in the code. For steps which are omitted from the computations, of course, no data need be added to the KQ, NQ, and V arrays; however, the total number of steps NKQ in the computations is reduced by 1 by statement 360.

Once the revision of the KQ, NQ, and V arrays has been completed, the revision of the IQ and JQ arrays is completed in the loop ending at statement 380 by deleting the unwanted species pairs and moving the remaining pairs up to fill in any resulting gaps. This process completes the editing of the cross section data to remove unnecessary steps in the computations which was begun at statement 110 of the subroutine, and provides a set of edited cross section data which will give the same computed cross sections as the original input data for all species included in the current gas model, but which includes only data for the N species for which cross sections are to be calculated, and omits all data for other species which may have been included in the input or the precoded data.

3.69.4 Default Options

The final section of subroutine XSECT, following statement 380, is for the purpose of providing cross section data according to the default options discussed in Section 4.6 of Volume II (ref. 2) for any pairs of species included in the transport calculations for which data were not supplied in the input. These data are added to the edited cross section data in common block /TRANS7/ to provide a complete set of data for the later cross section computations in subroutine PUTQIN.

The first section of coding below statement 380 supplies data for computing the unspecified neutral-neutral cross sections in the gas mixture from the empirical mixing rule, equation I(102). These data are added to the edited cross section data at the last step in which the option KQ = 10 is used, or, if this option is not used at all in the data, after the last step of the specified data. This procedure has been adopted in the code to permit data on the resonant contributions to the neutral-neutral cross sections to be specified after the last step in the computations using KQ = 10, without affecting the values of the default cross section computed from the mixing rule I(102).

The coding from statement 380 down to statement 400 determines the step LL in the cross section computations at which the unspecified neutral-neutral cross sections are to be added to the data. If no step with KQ(LL) = 10 is found in the data, an additional step is added for this purpose at the end, and the value of NQ(LL) for this step is set to NQ(LL-1), since as yet, the step does not apply to any pairs of species in the IQ, JQ list. The statements from 400 through 420 then determine the charge IZ, in units of the electronic charge, for each of the species in the transport calculations, using data from the common array LPIJ containing the α_{ij} matrix, and test the value of IZ to determine which are the neutral species. When no electrons are present in the gas, (i.e., IELEC = 0), all species are assumed to be neutral and IZ is not set. Since the correspondence I(J) which was previously established between the master species list for the code and the species in the transport property calculations is now no longer needed, the array I is now used to store the indices of the neutral species for later reference. Since no default option has been established in the code for the computation of cross sections for the collisions of a neutral species with itself, the code tests each neutral species to determine whether these data have been specified in the input and if they have not, terminates the case by transferring control to statement 540.

Once the neutral species have been determined, the code calls subroutine AXSECT to add the required data for these species to the edited cross section data in common block /TRANS7/. Subroutine AXSECT then takes the list of neutral species stored in the I array and tests all pairs (J,K) with J K, using the previously set values of the common variable Q(1,J,K), to determine those pairs for which cross section data have not previously been supplied. All such pairs found in this search are then added to the IQ, JQ list at the LLth step of the cross section computations, the values of the Q(1,J,K) for the added pairs are set to 1.0, and the IQ, JQ, and NQ lists are appropriately revised to take account of these additions. In these calculations subroutine AXSECT assumes that the species to be tested are stored in numerical order in the I array, and that NQ(LL) has been properly set prior to entrance into the subroutine.

After completing the specification of the neutral-neutral cross sections, the code next goes on to add data specifying the unspecified ion-ion cross sections to the edited cross section data. The three statments immediately preceding statement 430 establish a location in the edited data for the insertion of these cross sections. If the last step NKQ in the current version of the data is not presently being used in the cross section computations, it is used for the new data, while if it is being used, an additional step is added to the computations at the end of the current data for the insertion of the new data.

The loop from statements 430 to 480 adds data for computing effective Coulomb cross sections I(100) for the unspecified ion-ion collisions to the edited cross section data. The coding of this loop is set up to cause control to pass through the block of statements from 440 to 460 five times. For the case in which electrons are present (NELEC = 1), the values of the indices at each passage

through this block are shown in Table III.* For each passage through the block, the loop ending at statement 450 stores some of the species included in the transport property calculations in the array I, the exact species which are chosen depending on the values of the indices for that passage. Thus, referring to Table III, the array I consists of all singly charged ions in the trarsport calculations for the first passage through the loop, all singly charged ions plus electrons for the second passage, and so forth. The call to XSECT following statement 450 then finds all pairs of the species in the I array for which cross sections have not been previously specified, and revises the edited cross section data to include these pairs in the NKQ^{th} step of the computations. After returning to subroutine XSECT, the code checks to see whether any unspecified pairs were actually found by subroutine AXSECt in this search, and, if they were, supplies values of the parameters KQ and V for the step which are appropriate for the effective Coulomb potential I(100) from the preset array VCØUL, and then adds another new step at the end of the computations to prepare for storage of the next set of cross section data. The number of entries NEWMV in the V array, which was set originally in the loop ending at statement 340, is also revised to conform to the new data. Thus, the first passage through statements 440 to 460 sets the cross sections for all unspecified pairs of singly-charged ions to the effective Coulomb cross sections given for singly-charged ions in equations I(100a). The

---

*Note that for IELEC = 0, all species are assumed to be neutral so that all the unspecified cross sections were previously set in treating the neutral-neutral interactions. Thus, no cross sections remain to be set in the present loop and the edited cross section data are accordingly unaffected by the loop, independent of the values of the indices.

TABLE III

INDICES USED FOR CALCULATION OF EFFECTIVE COULOMB CROSS SECTIONS IN SUBROUTINE XSECT

Passage number through loop	J	KK	JJ	IE	Species in I array*	Species pairs for which cross section data are supplied*
1	1	1	1	1	1	(1,1)
2	1	1	2	0	e,1	(e,e) (e,1)
3	2	2	1	1	2	(2,2)
4	2	1	1	1	1,2	(1,2)
5	2	1	2	0	e,1,2	(e,2)

*The notation e indicates electrons, 1 indicates singly charged ions, and 2 indicates doubly charged ions.

second passage then sets the cross sections for all unspecified pairs involving either singly-charged ions or electrons; but, since the ion-ion cross sections were already set in the previous passage, only electron-electron and electron-ion cross sections are actually set in this passage. From equations I(100a) one sees that the effective Coulomb cross sections for electron-electron and electron-ion collisions are the same (for singly charged ions), so that both types of cross sections can be set in a single step of the computations. Similarly, the last three passages through the loop set the unspecified cross sections involving doubly-charged ions to the values indicated in equations I(100a), as shown in Table III. Thus, when all five passages through the loop have been completed, cross sections will have been specified in the edited cross section data for all pairs of charged species to be included in the transport calculations, up through doubly ionized species. Further, it should be noted that the last step NKQ in the edited data at this stage of the calculations will be an empty position which is not used in specifying the cross sections but is available for the addition of further data.

The preset coding of subroutine XSECT does not provide any default option for specifying the cross sections of ions which are triply charged or higher. Thus the section of code following statement 480 down through the IF statement below statement 490 checks for unspecified ion-ion cross sections for species with higher than double ionization, and, if any are found, terminates the case. Otherwise, the next three statements below the IF statement add all so far unspecified pairs of species in the transport property calculations to the final step NKQ of the cross section computations. Since all neutral-neutral and ion-ion cross sections have previously been specified, the cross sections added at this step are necessarily all neutral-ion cross sections. If any such cross sections are added, parameters are then added to the KQ and V arrays to compute them from an inverse fifth power potential (KQ = 6), using the data on the magnitude of the cross sections stored in the array locations OMEGA1(996), ASTAR (996), and BSTAR(996) in common block /TRANS4/. For the preset data used in the code, this results in the cross sections for the added pairs being set equal to the corresponding $N\text{-}O^+$ cross

sections. On the other hand, if no unspecified neutral cross sections are found, these data are not added and the value of NKQ is reduced by one to eliminate the unused step at the end of the cross section computations.

The above calculations complete the preparation of the edited cross section data in common block /TRANS7/. The number of steps NKQ in the computations is now checked, and if it does not exceed the maximum allowed by the program dimensions (NKQ = 100), control is returned to subroutine TRANSP. Otherwise the case is terminated by a transfer to statement 540.

In their final version, the edited cross section data prepared by subroutine XSECT specify the procedures to be used in computing all of the cross sections required for the transport property calculations in a given case, and are in a form suitable for later use by subroutine PUTQIN and its associated subroutines in the actual computation of these cross sections. These data are in their final form when control returns from subroutine XSECT, and are not altered by the code in the subsequent transport property calculations until XSECT is called again to prepare a new set of edited cross section data for another case.

4. GLOSSARY OF FORTRAN SYMBOLS

This section presents a complete glossary of Fortran symbols for the NATA code. Symbols appearing in common are listed first, followed by the remaining symbols in individual routines. The order is as follows: unlabelled common, labelled common in alphanumeric order of the block names, main program, subroutines and functions in alphanumeric order. The variables in each common block are listed in the order in which they appear in the block. The non-common variables are listed in alphanumeric order for each routine. The mathematical notation is as defined in Volume I (ref. 1). The list for unlabelled common includes some variables which are equivalenced to variables in unlabelled common.

4.1 Unlabelled Common

AA(I,J)	Matrix of coefficients and constants for a system of simultaneous linear equations
AAA(I,J)	Equivalent to AA(I,J)
CDIJ(I,J)	Matrix of coefficients $\bar{\nu}_{i-c,j}$ for expressing the dependent species in terms of the c independent species
CAPX(J)	Mole fraction X_J for the J^{th} species
GJ(J)	Species concentrations in moles per gram of mixture; equivalenced to CAPX(J)
A	α, constant in the density-area relation I(383)
AFNTS	Effective area ratio calculated for equilibrium flow in the nonequilibrium solution by the perturbation method
AFNX	Effective area ratio, A_e
AMACH	Mach number

AR	Indicator (initially 0) which is set to 1 when $d\ell n \tilde{A}_e/dx$ becomes positive downstream of the throat in the nonequilibrium solution by the inverse method.
ARBA	Maximum allowable number of tries at switching from upstream to downstream region in nonequilibrium solution
ARBB	Counter for number of times upstream-downstream switching point is moved downstream
BZERØ	b_0/W^0, where b_0 is the covolume of the molecules in the high-density modification of the equation of state and W^0 is the molecular weight of the undissociated gas.
C	Constant C in density-area curvefit relation I(383)
CARB	Effective area ratio at the beginning of the current integration step
CH	Nondimensional specific enthalpy, hW_0/R_0T_0, where h = specific enthalpy, W_0 = reservoir molecular weight, R_0 = universal gas constant, T_0 = reservoir temperature
CHA	Nondimensional specific enthalpy in the reservoir
CLNT	ℓn T, where T = temperature in ^oK
CM	Molecular weight of gas mixture, W (g/mole)
CMA	Molecular weight in reservoir, W_0 (g /mole)
CRA	R_0, universal gas constant (1.9872 cal/mole ^oK)
CRP	R_0T_0, where T_0 = reservoir temperature (^oK)
CRRB	Intermediate variable in calculation of entropy

CRS	Dummy variable for entropy
CSTA	$0.5\,\ell n\,(W_0/R_0T_0)$, where W_0 is the reservoir molecular weight (g/mole), R_0 the universal gas constant, and T_0 the reservoir temperature (^oK)
CT	Ratio of the temperature to the reservoir temperature, T/T_0
CTAP	Reservoir temperature, T_0 (^oK)
CTB	Dummy variable for CT
CTC	Dummy variable for CT
CTMAX	Ratio of throat temperature to reservoir temperature
CTMXX	Temperature for switching from thermo-fit to physical model for species properties, divided by reservoir temperature
CTP	Temperature in ^oK
CTPL	$\ell n\ T_0$
CTT	CT at the last printed step of the non-equilibrium solution
CX	Streamwise coordinate x(cm), zero at the throat and positive downstream
CXB	CX at the beginning of the current integration step
CXMAX	Maximum allowable value of x(cm)
DATEST	Value of effective area ratio at which nonequilibrium solution is switched from upstream to downstream region
DBTEST	(Not used)

DELT1	Nondimensional temperature decrement used in frozen and equilibrium solutions
DELT2	(Not used)
DELTAX	Increment in x (cm)
DLØGA	$d \ell n A_e/dx$
DLØGR	$d \ell n (\rho/\rho_0)/dx$, logarithmic derivative of nondimensional density
DT	$d(T/T_0)/DX$, gradient of nondimensional temperature
ENT	Intermediate variable used in frozen flow solution
FLUX	Nondimensional mass flux, $(\rho/\rho_0) \cdot u/\sqrt{R_0 T_0/W_0}$
HDELX	DELTAX/2
PCT	$\delta(T/T_0)$, perturbation in nondimensional temperature
PCTEST	Tolerance on C_χ when testing size of $\delta \chi_i$
PRES	Nondimensional pressure, p/p_0
PRESA	Reservoir pressure p_0 (atm)
PRESB	Nondimensional pressure at the preceding flow point; used in Mach number calculation
PRESTH	Nondimensional pressure at throat
PRHØ	Perturbation in nondimensional density, $\delta \rho / \rho_0$
RHAP	Density in reservoir, ρ_0 (g/cm^3)

RHØ	Nondimensional density, ρ/ρ_0
RHØB	RHØ at the preceding flow point; used in Mach number calculation
RHØBAR	Effective density in high-density equation of state divided by effective reservoir density
RHØC	(Not used)
RHØP	Density ρ in gm/cm^3
RHPL	$\ln \rho$
RHTH	Nondimensional density at throat, ρ_*/ρ_0
RØBARA	$\tilde{\rho}_0/\rho_0$
RØBARP	Effective density in reservoir, $\tilde{\rho}_0$ (gm/cm^3)
SCPG	$\sum_j \gamma_j \, c_{pj}/R_0$
SDT	Change in nondimensional temperature in integration interval
SEN	Entropy in cal/g ^oK
SHPG	$\sum_j \gamma_j H_j/R_0T_0$
SC	Factor by which integration interval is increased
SL	Characteristic length, ℓ, now set at 1 cm
SL64	$1/\sqrt{F}$, where F is the conversion factor from calories to ergs
SM	Nondimensional mass flux at the throat, RHTH·(SU)$_*$
SU	Nondimensional velocity, $u\sqrt{W_0/R_0T_0}$

SU2	$(SU)^2$
SUMG	$\sum \gamma_j$
TEST	Convergence criterion value for Newton-Raphson iterations in calculations of thermochemical equilibrium
TESTB	Tolerance allowed in locating temperature at throat for frozen and equilibrium flow
TPRINT	Nondimensional temperature interval at which results of nonequilibrium calculation are to be printed
TSTØP	Minimum value of nondimensional temperature desired in solution
UP	(Not used)
ZP	Natural log of nondimensional pressure
ZPA	Natural log of reservoir pressure in atm
BE(I)	$\beta_i = \sum_j \beta_{ij}$
BET(I)	$-1 + \nu^*_{i-c}$
BLBK(I)	Temporary storage area for boundary layer properties at the switch point from the upstream to downstream solutions
CAI(I)	A_i, constant factor in reaction rate formula
CAPQ(I)	Number of gram atoms of the I^{th} chemical element per mole of the cold gas; equivalenced to BLBK(I)
CAPXTH(J)	X_{j*}, mole fraction of J^{th} species at throat
CCI(J)	Temporary storage used in calculation of species molecular weights; equivalenced to CAPXTH(J)

CCPJ(J)	Molar heat capacity of the J^{th} species divided by R_0
CEACT(I)	Activation energy for i^{th} reaction (cal/mole ^oK)
CGI(I)	Species molecular weights (g/mole)
CGMU(I)	$\ell n\, X_i$, logarithms of mole fractions
CHI(I)	χ_i, variable measuring the departure of the i^{th} reaction from equilibrium, eq. I(289)
CHII(I)	Intermediate variables in computation of equilibrium constant based on mole fractions
CLN1MC(I)	$\ell n\, (1 - \chi_i)$
CLNPI(I)	$\ell n\, P_i$, variables in nonequilibrium solution
CMW(I)	Atomic weights of elements (g/mole)
DGJ(J)	d γ_j/dx; equivalenced to GJA(J)
ETAI(I)	η_i, temperature exponent in rate constant for i^{th} reaction
ETAJ(J)	n_j, number of atoms in a molecule of the J^{th} species; 0 value indicates no physical model data for thermal properties
GJA(J)	γ_{j0}, concentration of j^{th} species in reservoir
GJB(J)	GJ(J) at the beginning of the current integration step
PERTGJ(J)	$\delta\gamma_j$, perturbations in species concentrations
PGJ(J)	Intermediate variables in Newton-Raphson calculation of equilibrium composition.

PI(I)	Variables P_i in nonequilibrium solution, eq. I(288)
PICHI(I)	$P_i \chi_i$
QM(I)	Number of moles q_j of the j^{th} independent species in one mole of mixture
QQ(I)	1 if a third-body list is provided for the I^{th} reaction, 0 if not
SAJ(J)	$b_j + (1.5 + n_j) \ln T_0$, constant in physical-model expression for μ_j^o
SBJ(J)	b_j, eqs. (51), Vol. I; equivalenced to SAJ(J)
SDCHI(I)	$\delta \chi_i$, perturbation in χ_i
SDGJ(J)	Change in γ_j over an integration step
SENT(J)	Nondimensional species entropy, S_j^o/R_0
SHJ(J)	Nondimensional species enthalpy, H_j/R_0T_0
SHJA(J)	Nondimensional formation enthalpy of j^{th} species, H_{j0}^o/R_0T_0
SHJAP(J)	Formation enthalpy of J^{th} species, cal/mole
SKIL(I)	Intermediate variable in calculation of dependent species concentrations from concentrations of independent species
SS(J)	$\ln \gamma_{j0}$, logarithms of reservoir species concentrations
TB(J)	Temporary storage for quantities necessary to restart a step in the nonequilibrium calculation
TFA(J)	a_j, coefficient in thermo-fit expressions for H_j and μ_j^o

TFB(J)	b_j, coefficient in thermo-fit expressions for H_j^o and μ_j^o
TFC(J)	c_j, coefficient in thermo-fit expressions for H_j^o and μ_j^o
TFD(J)	d_j, coefficient in thermo-fit expressions for H_j^o and μ_j^o
TFE(J)	e_j, coefficient in thermo-fit expressions for H_j^o and μ_j^o/T
TFK(J)	k_j, coefficient in thermo-fit expressions for H_j^o and μ_j^o
THEV(J)	Nondimensional characteristic vibrational temperature, θ_{vj}/T_0
THEVP(J)	Characteristic vibrational temperature for J^{th} species, θ_{vj} (^cK)
XMJAT(J)	μ_j^o/R_0T
XNUI(I)	$\nu_i = \sum_j \nu_{ij}$, sum of stoichiometric coefficients on the reactant side of the i^{th} reaction
BETA(I,J)	$\beta_{ij} = \nu'_{ij} - \nu_{ij}$
BTA(I,J)	Equivalenced to BETA(I,J); not used
ELJ(L,J)	Nondimensional energy of L^{th} electronic level of J^{th} species, $E_{\ell j}/R_0T_0$
GELJ(L,J)	Degeneracy of L^{th} electronic level of J^{th} species, $g_{\ell j}$
XNUIJ(I,J)	ν_{ij}, stoichiometric coefficient of J^{th} species on reactant side of i^{th} reaction
XNUIJP(I,J)	ν_{ij}', stoichiometric coefficient of j^{th} species on product side of i^{th} reaction
IC	Number of ions included in chemical model

IM	Index of the first species in the equilibrium flow calculation, incremented when electrons are dropped from calculation
INEQ	Indicator (initially 0), reset to 1 when numerical integration of nonequilibrium solution is begun
INEQV	Indicator for selecting whether equilibrium (0) or frozen (1) vibrational model is to be used
IP	(Not used)
IRUN	Run number for identification
ISC	Number of elements in mixture (including the electron, if ionized species are included)
ISCP1	ISC+1
ISMC	Number of species minus number of elements (n-c)
ISMCNR	Current value of (n-c) in equilibrium flow calculation (decreased by IC when electrons are dropped from the calculation)
ISR	r, number of reactions
ISS	n, number of species
ISSNR	Current value of n in equilibrium flow calculation (decreased by IC when electrons are dropped from calculation)
ISSP1	n+1
ISSP2	n+2
ISSP3	n+3
ISSP4	n+4

ISW1A	Input control variable; 0 suppresses frozen solution
ISW1B	Input control variable; nonzero value gives edits of transport cross section calculations
ISW2A	Input control variable; 0 suppresses non-equilibrium solution
ISW2B	Input control variable; selects types of input data for reservoir calculation
ISW3A	Input control variable; 0 suppresses equilibrium solution
ISW3B	Input control variable; 0 suppresses boundary layer calculations
ISW4A	Input control variable; must be nonzero if another case follows, 0 for last case
ISW4B	Input control variable; nonzero gives dump in boundary layer routine BLAYER
ISW5A	Input control variable; nonzero gives dump tracing the execution of subroutine RESTMP, which computes the reservoir temperature from mass flow and reservoir pressure or stagnation enthalpy
ISW5B	Input control variable, nonzero gives dump in subroutines EXACT, COMM, NONEQ and PRTA for debugging nonequilibrium calculations
ISW6A	Input control variable; if positive, only the reservoir equilibrium calculation is done; if negative, edit of species thermal properties is produced
ISW6B	Input control variable; 0 suppresses the output of species mole fractions in the free-stream and model-point output

IUPD	Indicator (initially 1), reset to 0 when switch from inverse method to direct integration is made in nonequilibrium solution
IZERØ	0 (set in READ)
JJK	Indicator (initially 0), reset to 1 in NEWRAP when electrons are dropped from the equilibrium calculation
LC	(Not used)
M1	Index equal to c+1
NFIT	Indicator. 0 if thermo fit data are not used for any species
NIT	Counter for number of iterations in Newton-Raphson procedure
NNN	Integration step counter
NNS	Number of times Δx has been increased by current value of SC
NQS	Number of successful integration steps before Δx is increased
NQT	(Not used)
NTEST	Number of iterations allowed in Newton-Raphson procedure
IGJ(J)	Indicator; 1 if thermo-fit data are used for J^{th} species, 0 if not
IGM(J)	Number of electronic levels for the J^{th} species
ITB(I)	Temporary storage for indices necessary to restart step in nonequilibrium calculation

KUR(I,J)	U_{ij}, third body matrix
LPIJ(I,J)	α_{ij}, number of atoms of the J^{th} chemical element in a molecule of the I^{th} species
ACØM(I)	Hollerith description for case
ELMENT(I)	Chemical symbols for elements (Hollerith)
HP(I)	Symbols for species (Hollerith)

4.2 Common /AEGEØM/

SQRTA	Square root of the geometric area ratio
S1	Geometric area ratio (reset to the effective area ratio in AESOLN)
S2	Derivative of geometric area ratio (reset to derivative of the effective area ratio in AESOLN)

4.3 Common /AREA/

ATPI(J,I)	Downstream boundary of the J^{th} section in the I^{th} profile (cm from the throat, positive downstream)*
PARAM(L,J,I)	For L = 1 to 3, the parameter values for the J^{th} section in the I^{th} profile. See Sec. 4.3 of Volume I
RTHCM(I)	Throat radius (cm) for the I^{th} profile
NSECT(I)	Number of sections in the I^{th} profile
NSECTU(I)	Number of upstream sections in the I^{th} profile
ISHAPE(J,I)	Shape index for J^{th} section in the I^{th} profile

---

*"ATPI" is an acronym for "area transfer point input"

NPRØFL(I) Index of I^{th} profile in the precoded data; NPROFL(1) is equivalent to NOZZLE.

NPRFLS Number of profiles (1 for a nozzle, 2 for a channel)

NBL Index (1 or 2) of the profile which diverges from the channel axis least rapidly downstream of the throat

4.4 Common /AVG/

WSAVE Parameter controlling the averaging distance for the boundary layer correlation parameter, n

4.5 Common /BL/

DELBL(L) Displacement thickness of the boundary layer on the L^{th} profile divided by R0. For two-dimensional and axisymmetric nozzles, only DELBL(1) is used.

BLINT(L) Integral I for the boundary layer on the L^{th} profile, eq. I(172)

XZERØ Position at which boundary layer is assumed to begin (negative value, cm from the throat)

TWALL Nozzle wall temperature (^{o}K)

CPWALL Specific heat of the gas at the wall temperature (cal/g ^{o}K)

VISRØT Ratio of viscosity in the reservoir to the reservoir temperature (poise/^{o}K)

DIAM(L) Throat diameter for the L^{th} profile (inches)

SW $S_w = (h_w/h_0)-1$, where h_w is the gas enthalpy at the wall and h_0 the free-stream stagnation enthalpy

RO	Characteristic length used in boundary layer calculations (cm); equal to the throat radius for two-dimensional and axisymmetric nozzles; equal to $\sqrt{A'_*/\pi}$ for channels, where A'_* is the geometric cross sectional area of the channel at the throat
JDIM	Dimensionality index, 0 for two-dimensional nozzles, 1 for axisymmetric nozzles
IPØINT	Index of flow solution points at which boundary layer calculations are done

4.6 Common /BLNE/

XI(I)	For I = 1 and 2, the streamwise coordinate along the nozzle or channel surface for the I^{th} profile (cm)
ØRDINP(I)	The quantity Φ for the boundary layer on the I^{th} profile at the preceding flow point, eq. I(171)
DELBLP(I)	δ^*/RO for the boundary layer on the I^{th} profile at the preceding flow point
DDBLP(I)	$(d\,\delta^*/dx)/RO$ for the boundary layer on the I^{th} profile at the preceding flow point
AMPP	Mach number at the point preceding the previous flow point
AMP	Mach number at the preceding flow point
CØUPLD	Logical variable, set to .TRUE. when the coupling of the boundary layer to the inviscid flow is switched on
ISMD	The quantity (30-i) in eq. I(418)

4.7 Common /BLOUT/

REPF — Reynolds number per foot (ft^{-1})

THETA(L) — Momentum thickness for the boundary layer on the L^{th} profile, divided by R0

SN(L) — Correlation parameter n, eqs. I(158) and I(174), for the boundary layer on the L^{th} profile

XSN(L) — Averaged correlation parameter $\bar{n}$, eq. I(217), for the boundary layer on the L^{th} profile

PRREF — Prandtl number at the reference temperature

HR — Recovery enthalpy (cal/g)

QWDØT(L) — Heat flux to the wall surface with the L^{th} profile (Btu/ft^2sec)

TAUW(L) — Shear stress on the wall surface with the L^{th} profile ($lb\text{-}f/ft^2$)

4.8 Common /BLRAD/

YOZO — Product of the two throat radii for a channel (cm^2)

4.9 Common /CERRØR/

FAILED — Logical variable, set to .TRUE. when a convergence test on an integration step is failed in subroutine DERIVS or RNKT

4.10 Common /CHAN/

CP1(I) — Specifications for standard channel no. 1:

I = 1 Index of first channel profile in compiled-in list of profiles

I = 2 Index of second channel profile in compiled-in list of profiles

I = 3 Hollerith name for channel

I = 4 Index (1 or 2) of the profile which diverges from the channel axis least rapidly downstream of the throat

I = 5 Hollerith name for facility

CP2(I) Specifications for standard channel no. 2

CP3(I) Specifications for standard channel no. 3

CP4(I) Specifications for standard channel no. 4

CP5(I) Specifications for standard channel no. 5

4.11 Common /CØLDSP/

CGMW Molecular weight of the cold gas mixture (g/mole)

QPJ(J) Mole fraction of the J^{th} species in the cold gas mixture

IJCS(J) Index of the J^{th} cold species in the list of species for the current problem

NCS Number of cold species

4.12 Common /CONVRT/

CF(1) Conversion factor from nondimensional temperature CT to temperature in ^{o}K

CF(2) Conversion factor from nondimensional pressure PRES to pressure in atm

CF(3) Conversion factor from nondimensional density RHØ to density in lbm/ft^3

CF(4)	Conversion factor from nondimensional velocity SU to velocity in ft/sec
CF(5-7)	Conversion factors equal to 1
CF(8)	Conversion factor from viscosity in poise to viscosity in lbm/ft sec
CF(9)	Conversion factor equal to 1
CF(10)	Conversion factor from length in cm to length in inches
CF(11)	Conversion factor from nondimensional enthalpy CH to enthalpy in Btu/lbm
CF(12)	Conversion factor equal to 1
CF(13-16)	Conversion factors from nondimensional boundary layer thicknesses DELBL and THETA to thicknesses in inches
CF(17-26)	Conversion factors equal to 1

4.13 Common /DRV/

SHJDGJ	$(R_0T_0)^{-1} \cdot \Sigma H_j \, d\gamma_j/dx$, where H_j is the molar enthalpy of the jth species and γ_j is the concentration of the jth species in moles per gram of mixture (set in subroutine EXACT)

4.14 Common /EELEM/

EEP1(J) . . .	Data for first input-defined element J = 1 Atomic number J = 2 Atomic weight (g/mole)
EEP10(J)	Data for tenth input-defined element J = 1 Atomic number J = 2 Atomic weight (g/mole)

4.15 Common /ELEM/

EPl(I) Specifications for standard chemical element no. 1:

EPl(1) Hollerith name
EPl(2) Atomic weight (g/mole)

EP2(I) Specifications for standard chemical element no. 2

. .
. .
. .

EPl0(I) Specifications for standard chemical element no. 10

4.16 Common /EQC/

This block is used by subroutine EQCALC to communicate the results of its thermochemical equilibrium calculations to the calling routine.

ZCAP(J) Mole fraction X_j of Jth species

ZSEN Entropy (cal/g ^{o}K)

ZCH Nondimensional molar enthalpy, $(R_0T_0)^{-1}\sum X_jH_j$

ZCM Mean molecular weight

ZRBP Effective density ($\tilde{\rho}_0$) in reservoir for imperfect-gas model

ZRHØ Nondimensional density, ρ/ρ_0

4.17 Common /EQC2/

ZPZ $\ell n\ (p/p_0)$, logarithm of the nondimensional pressure

ZGMU(I) $\ell n\ X_i$, logarithm of the mole fraction for the Ith chemical species

4.18 Common /ERROR/

ERR — Logical flag set to "true" in DUMP routine; if "true", control is passed back to Main program, DUMPEX is called to print a dump of common data, and case is terminated

4.19 Common /GLIM2/

NØREAC(I) — Logical control variables; when NØREAC(I) is set to .TRUE., the I^{th} reaction is suppressed

4.20 Common /INGNE/

AIN(K,I) — The inverse A_{ki} of the square submatrix $\bar{\alpha}_{ij}$ of the matrix α_{ij} specifying the number of atoms of the j^{th} element per molecule of the i^{th} species; see eq. I(3)

4.21 Common /LN/

ISATØM — Index of atom used in Lewis number calculation, in master list of species

ISMØL — Index of molecule used in Lewis number calculation, in master list of species

JATØM — Index of atom used in Lewis number calculation, in list of species for current problem

JMØL — Index of molecule used in Lewis number calculation, in list of species for current problem

4.22 Common /MASSFL/

SMASS — Sonic mass flux based on total mass flow and geometric throat area ($g/cm^2 sec$)

CTMXXI — Temperature (^oK) above which species thermal properties are computed from the thermo fit for those species for which thermo fits are supplied

TSTØPI — Free-stream temperature at which the flow solutions will be terminated (^{o}K)

IS(J) — Index of the Jth species in the current problem, in the master list of species

4.23 Common /MIXT/

GPl(I) — Specifications of standard gas mixture no. 1:

I = 1	Hollerith name of mixture
I = 2	Number of chemical elements in mixture, including e^- if the gas model contains ion species
I = 3	Number of chemical species in mixture, including e^- if model contains ion species
I = 4	Number of reactions included in gas model
I = 5	Number of ions in gas model, excluding e^-
I = 6-15	Indices of elements present in mixture, in master list of elements; if electrons are present, they are the first element
I = 16-25	Mole fractions (QPJ) of the cold species
I = 26-45	Indices of the species included in the gas model, in the master list of species
I = 46-109	Indices of the reactions included in the model, in the master list of reactions

	I = 110-119 Indices of the cold species, in the master list of species
	I = 120 Number of cold species
	I = 121 Atom index for Lewis number calculations, in the master list of species
	I = 122 Molecule index for Lewis number calculations, in the master list of species
	I = 123 INT. If INT = 0, electron temperature equals gas temperature. If INT > 0, the model includes electronic nonequilibrium, and INT is the index of the extra reaction properties required in TNEP(I,INT)
	I = 124 Indicator for inclusion (1) or exclusion (2) of Fay-Riddell Lewis number factor in the stagnation-point heat flux
GP2(I)	Specifications of standard gas mixture no. 2
.	.
.	.
.	.
GP6(I)	Specifications of standard gas mixture no. 6

4.24 Common /MODPAR/

XMP1	Initial distance from throat at which model condition calculations are to be done (cm)
DXMP	(Not used)
FSTAG	Control variable; 0. value suppresses frozen shock calculations at model points, negative value suppresses equilibrium shock

CATFAC — Catalytic efficiency parameter Φ for stagnation point heat flux calculations; 0 for noncatalytic wall

TMØDEL — Model wall temperature for stagnation point heat flux calculations (^{o}K)

XMØDP1 — Initial distance from the throat at which model condition calculations are to be done (inches)

DXMØDP — (Not used)

TPLATE — Wall temperature for calculations of heat flux to a flat plate at zero angle of attack (^{o}K)

KDIM — Control variable for stagnation point heat flux calculations; 0 for two-dimensional model geometry, 1 for axisymmetric model

4.25 Common /MØDPT/

TSDIAM(I) — Test-section diameters at which model condition calculations are to be done (inches); in channel flow problems, specified channel widths at which flow calculations are to be done

TSAR(I) — Geometric area ratios at the nozzle stations defined by TSDIAM(I)

NTS — Number of specified test section diameters

MBL — In channel flow problems, index of the profile defining the channel width

4.26 Common /MØDP2/

MØDLPT — Logical indicator, .TRUE. when the current step in the nonequilibrium solution is at a model point

4.27 Common /NEQ/

ØMDST(L) — 1 - δ^*/R0 for the Lth profile at the sonic point

DDELBL(L) — (d δ^*/dx)/R0, derivative of nondimensional displacement thickness for the boundary layer on the Lth profile

4.28 Common /NEWMP/

FACMP — Factor by which distance from throat is increased in a geometric sequence of model points

NMØDPT — Number of model points in geometric sequence

4.29 Common /NØZZ/

ZP1(I) — Specifications of standard nozzle profile no. 1:

I = 1 — Throat radius in cm

I = 2 — Distance upstream of the throat of the point at which the boundary layer is assumed to originate (negative value, cm)

I = 3 — Number of profile sections upstream of the throat

I = 4 — Number of profile sections downstream of the throat

I = 5-16 — Shape indices (ISHAPE) of the profile sections (in order from upstream to downstream):

ISHAPE = 1 — Straight section

ISHAPE = 2 — Circular section convex toward axis

ISHAPE = 3 Circular section concave toward axis

I = 17-27 Downstream boundaries of profile sections (distances from the throat, positive downstream, cm)

I = 28-63 Parameters defining the profile sections (lengths in cm).* There are three parameter values P_1, P_2, P_3 for each profile. For ISHAPE = 1, the equation of a straight profile is

$$r(x) = P_1 + P_2 x$$

For ISHAPE = 2 or 3

P_1 = distance of circle center from axis

P_2 = x-coordinate of circle center

P_3 = circle radius

I = 64 Hollerith facility name

ZP2(I) Specifications of standard profile no. 2

.
.
.

ZP20(I) Specifications of standard profile no. 20

4.30 Common /ØUTMØD/

TB7 $N_0 \rho_0$, where N_0 is Avogadro's number (6.0225 x 10^{23} particles per mole) and ρ_0 is the density in the reservoir in g/cm^3.

---

*See Section 4.7 of Volume II

4.31 Common /ØUTPUT/

FVØUT(I) Array used for output of flow variables; set in subroutine ØUT1 (entry ØUT2)

I = 1 Axial coordinate, x (inches)

I = 2 Temperature (^{o}K)

I = 3 Static enthalpy (Btu/lb)

I = 4 Pressure (atm)

I = 5 Density (lb/ft^3)

I = 6 Nozzle diameter (inches); geometric area ratio in channel solutions

I = 7 Velocity (ft/sec)

I = 8 Mach number

I = 9 Entropy ($Btu/lb^{o}R$)

I = 10 Frozen specific heat ratio

I = 11 Effective area ratio

I = 12 Reynolds number (per ft)

I = 13 Molecular weight (g/mole)

I = 14 Viscosity (lb/ft sec)

I = 15 Electrical conductivity (mho/cm)

I = 16 Geometric area ratio; width of face with first profile in channel solutions

I = 17 Displacement thickness (inches)

I = 18 Momentum thickness (inches)

I = 19 Heat flux to wall (Btu/ft^{2}sec)

I = 20 Shear stress on wall (lbf/ft^2)

I = 21 Recovery enthalpy (Btu/lb)

I = 22 Prandtl number at the reference temperature

I = 23 Stanton number

I = 24 Reynolds number based on θ

I = 25 Re_θ for boundary layer transition

I = 26 Width of face with second profile (channel solutions only)

I = 27 Displacement thickness on second profile (inches)

I = 28 Momentum thickness on second profile (inches)

I = 29 Heat flux to wall on second profile (Btu/ft^{2}sec)

I = 30 Shear stress on wall on second profile (lbf/ft^2)

I = 31 Recovery enthalpy (Btu/lb)

I = 32 Prandtl number at the reference temperature

I = 33 Stanton number on second profile

I = 34 Re_θ on second profile

I = 35 Re_θ for transition

GJMF(J) (Not used)

4.32 Common /PØLYAT/

THEVE(K,J)	For K = 1-4, the four characteristic vibrational temperatures of the Jth species in the current gas model, if this species is a linear triatomic molecule. In subroutine READ, the entries K = 2-4 are set to values in ^{o}K. In subroutine INIT, these values are nondimensionalized by dividing them by the reservoir temperature, and the value K = 1 is set

4.33 Common /RDLIST/

IGAS	Index of standard gas mixture
IGASØ	IGAS value in preceding case (if any)
NØZZØ	Nozzle profile index for preceding case
ICHANØ	Channel index for preceding case

4.34 Common /RDMAIN/

HS	Input value of stagnation enthalpy (cal/g)
SUPGØ	Logical control for suppressing output of the gas model data when the model is the same as in the preceding case
MFITER	Input control for the iteration to take the displacement thickness into account in the reservoir condition calculations; zero value suppresses the iteration
NØ^{m}RAN	Input logical control for suppressing all transport property calculations

4.35 Common /RDMØD/

LEWIS	Index controlling use of the Fay-Riddell Lewis number factor in stagnation-point heat flux calculations. For LEWIS = 1, the factor is used; for LEWIS = 2, it is omitted
IAMBIP	Index specifying whether the diffusion process upon which the Lewis number is based is ambipolar (IAMBIP = 2) or not (IAMBIP = 1)

4.36 Common /RDØUT/

FLØW	Total mass flow in lbm/sec
FACNAM	Hollerith facility name
CHANAM	Hollerith channel name
LIMØUT	Index, nonzero if the boundary layer parameters N and XSN are to be printed in the output, equal to 0 if not

4.37 Common /RDTR/

ISW8B	Index controlling diagnostic dumps in the transport property routines. If ISW8B = 0, these dumps are omitted. If ISW8B > 0, the PUTQIN dump is produced once every ISW8B times the subroutine PUTQIN is called. If ISW8B < 0, the PUTQIN dump is suppressed.

4.38 Common /RDWEDG/

ANGLE(I)	Angles of attack (degrees)
RADLE(J)	Radii of leading edge (inches)
WX1	Initial distance from leading edge of wedge (inches)
DWX	Increment in distance from leading edge (inches)

WXI(K) Specified distances from leading edge (inches)

TWEDGE Surface temperature of wedge model (^{o}K)

WK Leading-edge drag coefficient, k

NWX Number of distances from leading edge (at increment of DWX) at which surface conditions on wedge model are to be calculated

NANGLE Number of angles of attack

NRADLE Number of leading-edge radii

WEDGEM Logical control variable; if "false", conditions on wedge are not calculated

AXISYM Logical control variable; if "false", stagnation point heat fluxes are not calculated

ISW9B Control variable, normally 0, with the following possible optional values:

$\pm$ 1 Print shock ordinate Y_s

$\pm$ 2 Print nondimensional coordinate ζ

$\pm$ 3 Print both Y_s and ζ

If ISW9B is negative, calculations are done using the unmodified Cheng-Kemp theory as well as the modified theory

4.39 Common /REAC/

RP1(I) Specifications of standard reaction no. 1:

I = 1 Coefficient A in formula for forward reaction rate constant (cm^3/mole sec or cm^6/$mole^2$ sec)

I = 2 Exponent η in formula for rate constant

	I = 3	Activation energy E_a in formula for rate constant (cal/mole)
	I = 4	Equal to 1.0 if a third-body list is provided for the reaction, equal to 0.0 if not
	I = 5	Number of reactant species (≤ 3)
	I = 6	Number of product species (≤ 3)
	I = 7-9	Indices of reactant species in master list of species
	I = 10-12	Indices of product species in master list of species
	I = 13-15	Numbers of molecules of reactants
	I = 16-18	Numbers of molecules of products
	I = 19	Number of third bodies
	I = 20-29	Indices of third-body species in master list of species
RP2(I)	Specifications of standard reaction no. 2	
.	.	
.	.	
.	.	
RP92(I)	Specifications of standard reaction no. 92	

4.40 Common /READAT/

PRESAI	Reservoir pressure input (atm)
DELT1I	Input value of nondimensional temperature decrement used in the frozen and equilibrium flow calculations and in starting the non-equilibrium solution

TPRNTI	The free-stream nonequilibrium solution is printed out at temperature intervals greater than or equal to TPRNTI times the reservoir temperature. For TPRNTI = 0., every step is printed
DELTXI	Input value of initial step size in x for nonequilibrium integration (cm)
CXMAXI	Distance beyond the throat at which the flow solutions are stopped (inches)
CTAPI	Reservoir temperature input (^{o}K)
XZERØI	Nozzle or channel inlet position at which the boundary layer is assumed to originate (negative value, measured in inches upstream from the throat)
PARAMI(L,M,N)	Input array of parameter values for nozzle or channel profiles (see definitions of code inputs, Group 4, in Section 2.3 of Volume II)
BZERØI	Input of constant in imperfect gas correction
HSTAG	Input stagnation enthalpy (Btu/lb)
READXS	Input control for reading cross section data for transport property calculations
READG	Input control for reading data on elements, species, and/or reactions
AAMS	Input control for automatic air model selection
AXIMØD	Input control for suppressing stagnation point model condition calculations if only wedge calculations are desired
ICASE	Case number in the current job
NEELS	Number of elements being defined in the input

ISW7B	Input control for suppressing the output of the boundary layer parameters N and XSN
INT	Indicator for electronic nonequilibrium in a gas model
ICHAN	Index of standard channels
NQSI	Number of successful integration steps required before the step size in the non-equilibrium calculation is increased
INEQVI	Input control for equilibrium or frozen molecular vibration
NRECO	Input of number of records already on data tape at beginning of run
JCS(I)	Indices of cold species in master list of species
ISCI	Input of the number of chemical elements in the gas model
ISSI	Input of the number of chemical species in the gas model
ISRI	Input of the number of reactions in the gas model
ICI	Input of the number of ions in the gas model
IE(I)	Input indices of elements present in the gas model, in the master list of elements
IR(I)	Input indices of reactions in the gas model, in the master list of reactions
NSECTS(I,J)	For I = 1, number of upstream sections in the curvefit for the Jth profile; for I = 2, number of downstream sections
IEEP(I)	Indices assigned to the input-defined elements in the master list of elements

4.41 Common /RENE/

GAMIN	Concentration (moles/g) below which a species will be frozen if it decreases so rapidly that it controls the integration step size.
HTEST	Maximum relative change in total enthalpy in an integration step.
TETEST	Maximum relative change in electron temperature in an integration step
QTEST	Criterion value for maximum allowable change in energy transfer to electron gas in an integration step
DCHLL	Parameter used in selecting initial integration step size; the initial Δx is not allowed to exceed 0.01 $\lvert\delta\chi_i\rvert_{min}$/DCHLL.
DCHRAT	Parameter controlling the artificial increase in reaction rates in the perturbation solution to avoid premature startup of the numerical integration (Section 7.3.7 of Volume I)
CCHI	Parameter C_χ in criterion for switch from perturbation solution to numerical integration
TTEST	Maximum relative change in the gas temperature in an integration step
GTEST	Maximum relative change in a species concentration in an integration step

4.42 Common /RESPRP/

VISCR	Viscosity in the reservoir (poise)
PRR	Prandtl number in the reservoir
SIGR	Electrical conductivity in the reservoir (mho/cm)

FLEWR — Lewis number in the reservoir

4.43 Common /SPEC/

SP1(I) — Specifications for standard species no. 1:

I = 1	Hollerith name for species
I = 2	Number of elements in species (≤ 3)
I = 3-5	Indices of elements in the master list of elements
I = 6-8	Numbers of atoms of elements
I = 9-14	Thermo-fit parameters a, b, c, d, e, k
I = 15	Formation enthalpy (cal/mole)
I = 16	Number of atoms in a molecule of the species
I = 17	Chemical constant, b
I = 18	Characteristic vibrational temperature (^{o}K)
I = 19	Number of electronic levels (≤ 10)
I = 20	1 if thermo-fit is used, 0 if not
I = 21-30	Degeneracies of electronic levels
I = 31-40	Energies of electronic levels (cal/mole above ground state)
I = 41-43	Second, third, and fourth characteristic vibrational temperatures for linear triatomic species (^{o}K)

4.44 Common /SS/

CAS	Square of the sound speed divided by the non-dimensional temperature T/T_0 (cm^2/sec^2)
US	Standard velocity used for nondimensionalizing the flow velocity, $\sqrt{R_0 T_0/W_0}$, where T_0 is the reservoir temperature, W_0 the reservoir molecular weight, and $R_0 = 8.31436 \times 10^7$ erg/mole$^\circ$K

4.45 Common /STAG/

T2E	Initial estimate of temperature behind equilibrium normal shock ($^\circ$K)
EPSLØN	Initial estimate of density ratio across equilibrium normal shock
T2F	Initial estimate of temperature behind frozen normal shock ($^\circ$K)
EPSF	Initial estimate of density ratio across frozen normal shock

4.46 Common /STEPS/

NSTEPS	Number of successful integration steps since the last printout of conditions at a flow point

4.47 Common /SWITCH/

XPB	The corrected x-coordinate of the flow point preceding the point at which BLAYER is first called with x corrected for the displacement thickness in the frozen and equilibrium solutions (cm)
DXPB	The difference in corrected x values between the point at XPB and the preceding point

W	Constant parameter w determining the characteristic distance over which the boundary layer correlation parameter n is averaged; see eqs. I(218) and I(217)

4.48 Common /TAPØUT/

XXX(I)	Block of storage used to set up data which are to be saved on binary tape for subsequent processing by other programs
ITPØUT	File number of binary output tape
NRCØUT	Number of records written on binary tape during currert case of job
IFLØW	(Not used)
ITYPER	Type of record written on binary tape: 1 "Case" record with overall definition of flow solution being computed 2 "Point" record containing free-stream flow variables 3 "Model" record containing data for stagnation point model
IMP	Index of model points written onto binary tape in current solution
DATAPE	Logical control variable; if "false", no data are written onto binary tape

4.49 Common /TEMPRY/

SAVEC(J)	Species mole fractions for which transport property calculations are done

4.50 Common /THRT/

RSA	Factor by which cross sectional area of flow has been rescaled in nonequilibrium solution

4.51 Common /TNCE/

SUMGH — $\sum_{j=2}^{n} \gamma_j$, sum of the molar concentrations for all species except the electron (mole/g)

SCPGH — $\sum_{j=2}^{n} C_{pj} \gamma_j / R_0$, sum of the products of the molar heat capacities and molar concentrations for all species except the electron, divided by the universal gas constant

QDPR — Power radiated from the gas, cal/cm^3-sec

QDPE — Net power transferred to the electron gas (cal/cm^3-sec)

4.52 Common /TNDC/

XMJATD(J,L) — Chemical potential at standard pressure (μ_j^o, for the J^{th} species, evaluated at the L^{th} temperature (L = 1, heavy particle temperature; L = 2, electron temperature), divided by R_0 times the L^{th} temperature

CLNTD(L) — Natural logarithm of the Lth temperature (L = 1, heavy particle temperature; L = 2, electron temperature)

CTD(2) — Nondimensional temperatures (L = 1, T/T_0; L = 2, T_e/T_0)

4.53 Common /TNE/

TN1(I) — Specifications of data for standard electronic nonequilibrium model no. 1 (helium):

I = 1-25 — KTF(I), indicator for type of formula for forward rate constant of the Ith reaction; see Sec. 4.4 of Vol. II

	I = 26-50 KTR indicators for the reverse reactions in the gas model. KTR = 0 if the backward rate constant k_r is zero; KTR = 1 if $k_r = k_r(T)$; and KTR = 2 if $k_r = k_r(T_e)$
	I = 51-75 ITR values for the reactions in the gas model. ITR is an indicator of the rule for partitioning the reaction energy between the electrons, the heavy particles, and radiative losses. The significance of its values is discussed in Sec. 4.4 of Vol. II
	I = 76-100 ϵ_0 parameters for reactions
	I = 101-125 Values of the parameter "a" for reactions with ITR = 1
	I = 126-155 Temperature values for table of elastic collision cross section, $\bar{Q}^{(1,1)}$
	I = 156-185 $\bar{Q}^{(1,1)}$ values for table
	I = 186 Parameter b for reactions with KTF = 4
TN2(I)	Specifications of data for electronic non-equilibrium model no. 2 (argon)

4.54 Common /TNEQ/

TLIST(1)	For I = 1-30, temperatures for elastic collision cross section table for current gas model (^{o}K)
PØM(I)	For I = 1-30, cross section values for elastic collision cross section table for current gas model

4.55 Common /TNERK/

SDTE $\Delta T_e/T_0$, change in electron temperature in integration step

CTEB T_e/T_0 at the start of the integration step

DCHA $(W_0/R_0T_0)(dh_0/dx)$, rate of change of non-dimensional total enthalpy

CHB W_0h_0/R_0T_0, nondimensional total enthalpy at start of integration step

SDCHA $\Delta(W_0h_0/R_0T_0)$, change in nondimensional total enthalpy in integration step

DQMAX D_{qm}, criterion value for convergence test on changes in energy transfer to the electron gas

IFAIL Indicator for cause of integration step failure:

IFAIL	Meaning
0	Failure in COMM or DERIVS (called from RNKT)
1	Test on T in NØNEQ
2	Test on T_e in NØNEQ
3	Test on h_0 in NØNEQ
4	Test on DLØGA in NØNEQ
5	Test on QDPE in NØNEQ
11-30	Test on GJ(IFAIL-10) from RNKT in NØNEQ
31-50	Test on GJ(IFAIL-30) from element conservation in NØNEQ

IFAIL	Meaning
60	Failure in DERIVS or CØMM (called from NØNEQ)
-1 to -ISS	Test on sign of GJ(-IFAIL) in RNKT (GJ2 or GJ3)
-(ISS+1)	Test on sign of T in RNKT (GJ2 or GJ3)
-(ISS+2)	Test on sign of T_e in RNKT (GJ2 or GJ3)
-(ISS+3)	Test on sign of h_0 in RNKT (GJ2 or GJ3)
-(ISS+4)	Test on QDPE in RNKT (GJ2 or GJ3)
-31 to (-30+ISS)	Test on sign of GJ(-IFAIL-30) in RNKT (GJ4, Runge-Kutta)
-(31+ISS)	Test on sign of T in RNKT (GJ4, Runge-Kutta)
-(32+ISS)	Test on sign of T_e in RNKT (GJ4, Runge-Kutta)
-(33+ISS)	Test on sign of h_0 in RNKT (GJ4, Runge-Kutta)
-(34+ISS)	Test on QDPE in RNKT (GJ4, Runge-Kutta)
-61 to -(60+ISS)	Test on sign of GJ(-IFAIL-60) in RNKT (GJ4, Treanor)
-(61+ISS)	Test on sign of T in RNKT (GJ4, Treanor)
-(62+ISS)	Test on sign of T_e in RNKT (GJ4, Treanor)

IFAIL	Meaning
-(63+ISS)	Test on sign of h_0 in RNKT (GJ4, Treanor)
-(64+ISS)	Test on QDPE in RNKT (GJ4, Treanor)

4.56 Common /TNØNEQ/

CTE	Electron temperature divided by gas temperature in reservoir
DTE	Derivative of CTE with respect to x, i.e., $d(T_e/T_0)/dx$
BPAR	Parameter b for reactions with KTF = 4
EPAR(I,JR)	EPAR(1,JR) = parameter ϵ_0 for the JRth reaction in cal per N_0 reactions; EPAR(2,JR) = parameter a for the JRth reaction if ITR(JR) = 1
NT	Indicator for electronic nonequilibrium in the gas model; for NT =1, the gas and electron temperatures are assumed to be equal; for NT = 2, they are not
ITR(JR)	Indicator of the rule for partitioning the reaction energy between the electrons, the heavy particles, and radiative losses in the JRth reaction of the current gas model. See discussion in Section 4.4 of Vol. II
KTF(JR)	Indicator for temperature dependence of the forward rate constant k_f for the JRth reaction in the current gas model; see Sec. 4.4 of Vol. II

KTR(JR) Indicator for temperature dependence of the backward rate constant k_r for the JRth reaction in the current gas model:

KTR = 0	$k_r = 0$
KTR = 1	$k_r = k_r(T)$
KTR = 2	$k_r = k_r(T_e)$

ICH(J) Indicator for charged species; equal to 1 if the Jth species is neutral, equal to 2 if the species is an ion

IPA(I) Species index for the product atom in the Ith reaction; if KTF(I) is not equal to 4, IPA(I) is set to zero

4.57 Common /TRANI/

IIi(K) For K = 1 to a maximum of 5, the index of the first species in the Kth pair of species to which the cross section calculated in the i^{th} step of the transport property calculation is applied. The common block provides storage for i = 1 to 100

4.58 Common /TRANJ/

JJi(K) For K = 1 to a maximum of 5, the index of the second species in the Kth pair of species to which the cross section calculated in the i^{th} step of the transport property calculation is applied. The common block provides storage for i = 1 to 100. Only pairs with IIi(K) $\leq$ JJi(K) are used

4.59 Common /TRANS1/

T Temperature in ^{o}K for transport calculations

Q(K,I,J) — Q(K,I,J) is used in the cross section computations to store the averaged cross sections for the species pair (I,J) as follows:

$$Q(1,I,J) = \bar{\Omega}_{ij}^{(1,1)},$$

$$Q(2,I,J) = \bar{\Omega}_{ij}^{(2,2)},$$

$$Q(3,I,J) = B^{*}_{ij}\ \bar{\Omega}_{ij}^{(1,1)}.$$

At other stages in the calculations, Q(K,I,J) is also used in subroutine TRANSP to store the matrix elements for the transport calculations, and in subroutine XSECT to indicate the species pairs for which cross section data have been supplied

ZM2(I) — Quantity proportional to molecular weight of species I, eqs. (171)

4.60 Common /TRANS2/

KKQ(L) — Index of option to be used for step L of cross section computations in input or precoded data; see Section 4.4 of Vol. II

NNQ(L) — Number of species pairs in IIQ, JJQ arrays for which option KKQ(L) is specified in input data

ISEQ(K) — Sequencing array specifying the order in which the defined steps of the transport cross section calculation are to be carried out

NNKQ — Number of steps specified for cross section computations

I(J) — Temporary array of species indices used by subroutine XSECT in preparing edited cross section data

4.61 Common /TRANS3/

B(I,L)	Intermediate variable in transport calculations; see KINT and KANDMU, Sections 3.27 and 3.28
BR(I,L)	Reciprocal of B(I,L)
A(L)	Intermediate variable in transport calculations; see KANDMU, Section 3.27
X(I)	Mole fraction of species I in gas mixture
DH(I)	Nondimensional internal specific heat $(W_i C_{pi}/N_0 k - 5/2)$ for species I in gas mixture

4.62 Common /TRANS4/

TL(I)	Independent variable (temperature) for tabulated cross section data
ØMEGA1(I)	Value of first parameter determining cross sections at TL(I) for tabulated cross section data
ASTAR(I)	Value of second parameter determining cross sections at TL(I) for tabulated cross section data
BSTAR(I)	Value of third parameter determining cross sections at TL(I) for tabulated cross section data

4.63 Common /TRANS5/

N	Number of species included in transport calculations
IELEC	IELEC=0 indicates that electrons are not included in the calculations
ID1	Index in the current gas model of first species used in Lewis number computation

ID2	Index in the current gas model of second species used in Lewis number computation

4.64 Common /TRANS7/

V(M)	List of parameters for cross section computations in edited data
KQ(L)	Option to be used for step L of cross section computations in edited data
NQ(L)	Index of last species pair in IQ,JQ arrays to which step L of the edited cross section computations is to be applied
IQ(N)	List of first indices for species pairs to which edited cross section data are to be applied
JQ(N)	List of second indices for species pairs to which edited cross section data are to be applied
NKQ	Number of steps in cross section computations for edited cross section data

4.65 Common /TRANS8/

NV(K)	Number of parameters V required for option K of the cross section computations
N	Number of species included in cross section computations

4.66 Common /TRANV/

VVi(K)	For K = 1 to a maximum of 5, the input parameters for the i^{th} step of the cross section calculations. The common block provides storage for i = 1 to 100

4.67 Common /TRPRØP/

VISC	Viscosity of mixture in poise
PRF	Frozen Prandtl number for mixture (dimensionless)
SIGMA	Electrical conductivity of mixture in mhos/cm
FLEWIS	Atom-molecule Lewis number of mixture (dimensionless)

4.68 Common /WMØD/

TSF	Frozen stagnation temperature (^{o}K)

4.69 Main Program

ATEST	Previous value of A in the Newton-Raphson iteration to determine the parameter α in the density-area relation I(383)
ATH	Effective cross-sectional area of the flow at the throat, with allowance for the boundary layer displacement thickness (cm^2)
CHTH	Nondimensional enthalpy hW_0/R_0T_0 at the throat
CMTH	Molecular weight of gas mixture at the throat
D	$(\rho_*/\rho_0)^A$ (A+2) -2, where ρ_* is the density at the sonic point in the equilibrium flow solution and A is the current estimate of the parameter α in the density-area relation
DELT	Initial decrement of the nondimensional temperature for starting the perturbation calculation of the nonequilibrium flow
D1	$(\rho_*/\rho_0)^A \left[(A+2) \ln (\rho_*/\rho_0) + 1\right]$
ELTMIN	Case execution time in minutes

ET	Elapsed time from beginning of execution (seconds)
ETO	Elapsed time from beginning of execution at the start of a case (seconds)
GJ(I)	Species concentrations in moles/g
GJA(I)	Species concentrations in the upstream reservoir, moles/g
GMF(J)	Mole fraction of the J^{th} species under equilibrium sonic conditions
HOP	Reservoir enthalpy (cal/g)
I	DØ index; also, index, in the list of species for the current problem, of the K^{th} cold species
IRCØ	Indicator; value 1 indicates that call to TRANSX was skipped in the previous case because tables of species thermal properties were generated
ISØLN	Indicator for type of flow solution: ISØLN = 1 Frozen ISØLN = 2 Equilibrium ISØLN = 3 Nonequilibrium
IS3	Storage for input value of ISW3B; used to restore value after a case in which ISW3B was set to zero because NØTRAN = .TRUE.
K	Cold species index
N	Iteration counter in Newton-Raphson solution for parameter α in area-density relation
NCASE	Counter for cases in a run
NCØMPL	Number of cases successfully completed in a run

NERRØR	Number of failed cases in a run
NØZZLE	Nozzle profile index
NRCTØT	Total number of records written on tape 8 during a run
ØNE	1.0
PIV	3.14159
SUM1	$\ln\ (\rho_*/\rho_0)$
SUM2	A+2
SUM3	$(\rho_*/\rho_0)^A$
UTH	Nondimensional flow velocity (SU) at throat
VISCC	Viscosity in the reservoir (lbm/ft-sec)

4.70 Subroutine AESØLN

DELP(L)	Displacement thickness in cm, on Lth profile
DELSTP(L)	Displacement thickness at the throat in cm, on Lth profile
DENØM	One-fourth of the effective flow area at the throat for a channel (cm^2)
DYDX	Slope dy/dx for the first profile in a channel
DZDX	Slope dz/dx for the second profile in a channel
FAC1	Effective ordinate of first profile in a channel (cm)
FAC2	Effective ordinate of second profile in a channel (cm)

SQRTAE	Square root of the effective area ratio, $\sqrt{A_e}$
X	Axial coordinate in nozzle (cm)
Y	Geometric ordinate of first profile in a channel (cm)
Z	Geometric ordinate of second profile in a channel (cm)

4.71 Subroutine AGSØLN

A	Trial value of geometric area ratio
AE	Effective area ratio
AEC	Constant term in equation relating the geometric and effective area ratios for a channel
AG	Final solution for the geometric area ratio
DA	Dummy argument in call to GEØMAR; represents X-derivative of geometric area ratio (not used)
DEL(L)	Displacement thickness divided by R0, for boundary layer on Lth profile
DELP(L)	Displacement thickness (cm) for boundary layer on Lth profile
DSTP(L)	Displacement thickness at the throat (cm) for boundary layer on Lth profile
F	Function which is 0 when the correct geometric area ratio has been found
FØ	F value in previous step of iteration
ICØUNT	Iteration counter
L	DØ index over channel profiles

RNAME	Routine name for call to DUMP routine
SQRTA	Square root of geometric area ratio
UPDØWN	Indicator; 1.0 if downstream solution is desired by calling routine, -1.0 if upstream solution is desired
X	Axial coordinate in nozzle (cm)
XL	Temporary storage for previous value of X
XØ	Value of X in previous iteration
Y	Geometric ordinate of first profile in a channel (cm)
Y0Z0	$y_0 z_0$, product of profile ordinates at the throat; one-fourth of the geometric throat area in a channel (cm^2)
Z	Geometric ordinate of second profile in a channel (cm)

4.72 Subroutine AXFIT

AG	Geometric area ratio
UPDØWN	Indicator; -1.0 upstream of throat, 1.0 downstream

4.73 Subroutine AXSECT

I(J)	List of species for which cross section data are wanted (common array)
II	Index of first species in species pair
IV	Location of species II in the I array
JJ	Index of second species in species pair
JV	Location of species JJ in the I array

L	Running index for DØ loops
LL	Step in cross section computations where data are to be added
LQ	Position of species pair in IQ,JQ list
LQ1	Position in IQ,JQ list where additional species pairs are to be added
N	Number of species pairs added to IQ,JQ list
NN	Number of species from I array to be considered in adding data

4.74 Subroutine BLAYER

AE	Effective area ratio
AG	Geometric area ratio
AGJ	Factor proportional to r^{2j} in numerical integration from x_0 to the first flow point
AG1	Geometric area ratio in numerical integration from x_0 to the first flow point
AJ	Factor proportional to r^{2j}
ALPR	Natural logarithm of the Prandtl number at the reference temperature
AM	Mach number
AM1	Mach number in numerical integration from x_0 to the first flow point
AS	Sound speed (cm/sec)
AVCON	W/R0 parameter used in averaging the correlation parameter

AX	Intermediate quality, eq. I(194d) in curvefit to shear parameter
B	Coefficient of n in linear approximation to $N(n,S_w)$, eq. I(164)
BLINTI	Value of integral I, eq. I(172), at the beginning of a step in the boundary layer solution
BM1	B-1
BX	Intermediate quantity, eq. I(194e), in curvefit to shear parameter
CC	MA_g in low Mach number region of integration from x_0 to first flow point
CØSB(L)	Cosine of angle between tangent to Lth profile and the nozzle axis
C1	Intermediate quantity in quadratic solution for dM/dx
C2	Intermediate quantity in quadratic solution for dM/dx
DADX	dA_g/dx, derivative of geometric area ratio
DLM	$d\ell n\, M/dx$, logarithmic derivative of Mach number
DMDX	dM/dx, derivative of Mach number
DUDX	Derivative of nondimensional velocity
DX	Axial distance from the previous flow point to the current one
DXI	Change in ξ in the current step

DXØ	Size of last previous "final" step, apart from steps at model points
DXP	Distance from the point before the previous flow point to the previous flow point
DXSQ	$(DX)^2$
DX2	DX + DXP
DYZDX(L)	Derivative of the radius of the Lth profile with respect to the axial coordinate
EK1	Coefficient A in eq. I(164)
ENPRIM	n', eq. I(17?)
FINAL	Logical flag; "false" when BLAYER is called during an intermediate calculation of the non-equilibrium integration; "true" for the final calculation of conditions at each flow point
F0	Intermediate quantity in curvefit to shear parameter, eq. I(194)
F2	Intermediate quantity in curvefit to shear parameter
F3	Intermediate quantity in curvefit to shear parameter
F4	Intermediate quantity in curvefit to shear parameter
GAMMAE	Effective specific heat ratio, $\gamma_e = a^2W/RT$, used in calculation of effective hypersonic parameter
GJ(I)	Species concentrations (moles/g)
H	Nondimensional enthalpy, hW_0/R_0T_0

HCØN	R_0T_0/W_0, factor for converting nondimensional enthalpy into cal/g
HE	Free-stream specific enthalpy (cal/g)
HF	Form factor, c^*/θ
HFP1	HF + 1
HREF	Reference enthalpy
HTR	Incompressible form factor, H_{inc}
HTRP1	HTR + 1
HW	Enthalpy of the gas at the wall temperature
H0	Specific stagnation enthalpy (cal/g)
I	DØ index
ITYPE	Index for type of nozzle geometry ITYPE = 1 Two-dimensional nozzle ITYPE = 2 Axisymmetric nozzle ITYPE = 3 Rectangular channel
J	DØ index
K	DØ index over the cold species
L	DØ index running over the nozzle or channel profiles
LPR	Index, equal to LPRIME(L)
LPRIME(L)	Equal to 2 for L = 1, equal to 1 for L = 2
ØLDMF	Logical flag; "true" if no mole fraction has changed by 0.01 since the last flow point at which transport properties were calculated

ØMEGA	ω , viscosity-temperature exponent in eq. I (191)
ØM1	Max (1- ω ,0)
ØNE	1.
ØRDIN(L)	Φ for the boundary layer on the Lth profile, eq. I(171)
ØRD1	A value of Φ used in the numerical integration from x_0 to the first flow point
ØRD2	See ØRD1
P	Nondimensional pressure, p/p_0
PRPWR	$(PRREF)^{0.56}$
PRW	Prandtl number at the wall temperature
R	Ratio of local nozzle radius to throat radius for axisymmetric nozzles
RA	Intermediate quantity in curvefit to Reynolds analogy factor
RAF	Reynolds analogy factor R_A in eq. I(195)
RB	Intermediate quantity in curvefit to Reynolds analogy factor
RER	Reynolds number based on R0
RH	Nondimensional density
RMR	$\rho\mu$ ratio, $\rho_w \mu_w / \rho_e \mu_e$
RWMW	ρ_w ;
R1	Radius of a profile in the numerical integration from x_0 to the first flow point

R2	See R1
SIGH	Hypersonic parameter σ
SQT	Intermediate quantity in transverse curvature correction, eq. I(187)
SWFAC	$(S_w + 1)\left[7 + 3.4\ (S_w + 1)\right]$
SWFAC2	$1 + 4.5\ (S_w + 1)^{0.9}$
SWP1	$S_w + 1$
T	Nondimensional temperature, T/T_0
TE	Free stream temperature, T
TEP	Free stream temperature at which transport property calculations were last done
THETA1	Momentum thickness based on eq. I(185), divided by R0
TR	T^*/T_w, ratio of reference temperature to wall temperature
TRCALC	Logical flag, .TRUE. if transport properties were calculated at the current flow point
TREF	Reference temperature (^oK)
TREFP	Reference temperature at which transport properties were last calculated
TRP(I)	Array equivalenced to common block /TRPRØP/, to permit processing these data in a DØ loop
TRPSV(I)	Array used to save the transport properties at the free stream temperature
U	Nondimensional velocity, u/u_s (see US in common block /SS/)

UPRIME	Free stream velocity, u (cm/sec)
VISCR	Viscosity at the reference temperature (poise)
VISCW	Viscosity at the wall temperature (poise)
VR	VISCR/VISCW
WD	Weight factor w in calculation of d δ^*/dx; eq. I(419)
WW	$e^{-\Delta x/a}$ in eq. I(217); weighting factor for $\bar{n}_p$ in the calculation of the averaged correlation parameter, $\bar{n}$
X	Axial coordinate (cm)
XII	Previous value of XI
XL	Shear parameter ℓ, eqs. I(194)
XP	Previous value of x (cm)
XPXP	X + XP
XSAVE(I)	Temporary storage for species mole fractions SAVEC(I)
XSNV	min ($\bar{n}$,0); used in place of $\bar{n}$ in calculation of HTR to ensure the use of a nonpositive value
XTEST(J)	Mole fraction of the Jth species used in the last previous calculation of transport properties
XX	Axial coordinate variable in numerical integration from x_0
YZ(L)	Radius of the Lth profile (cm)
ZERO	0.

4.75 Subroutine BLCALL

FINAL	Logical subroutine argument; .FALSE. when BLAYER is called during an intermediate calculation of the nonequilibrium integration, .TRUE. for the final calculation of conditions at the end of each successful integration step
GJ(J)	Species concentrations in moles/g
J	DØ index over the species

4.76 Subroutine BXSECT

I	First index of species pair in V array
J	Second index of species pair in V array
K	Value of KQ for step L of cross section computations
L	Index of step in cross section computations which is currently being considered
M	Number of sets of constant factors to be used in computations, for option KQ = 14 only
MADD	Index used in locating species pairs in V array
MM	MM + 1 and MM +2 are the locations of the current species pair in the V array
MV	Index of last parameter in V array for step L-1 of the computations

4.77 Subroutine CØMM

ALGJ(J)	$\ln \gamma_j$, natural logarithm of the Jth species concentration

CLKF	$\ell n\ k_f$, natural logarithm of the forward rate constant
CLKR	$\ell n\ k_r$, natural logarithm of the reverse rate constant
CLTBF	$\ell n\ (\Sigma \rho \gamma_j)$, where the sum is over the third-body species in a reaction
CLZ2	ℓn (Z2)
CLZ3	ℓn (Z3)
EF	ϵ_{fi} in eq. I(321c)
ER	ϵ_{ri} in eq. I(321c)
GJ(J)	Concentration (γ_j) of the Jth species (mole/g)
I	DØ index over the reactions
ICHG	Equal to 1 for a neutral species, equal to 2 for a charged (ionic) species
ICØUNT	Counter for number of times CØMM has been called in a run
ICYCLE	Interval in ICØUNT at which ISW5B is set equal to 1 to provide diagnostic dumps in subroutine CØMM, EXACT, RNKT, and PRTA
IPAI	Index of atomic species produced in current reaction for KTF(I) = 4
J	DØ index over the species
KF	Indicator for temperature dependence of forward reaction rate constant; for KF = 1, k_f is a function of T; for KF = 2, it is a function of T_e. For other values of KF, see discussion of KTF input in Sec. 2.3 of Vol. II.

KR	Indicator for temperature dependence of reverse reaction rate constant; for KR = 1, k_r is a function of T; for KR = 2, it is a function of T_e
LØGC	Logical indicator for method of calculation of PICHI(I); for LØGC = .FALSE., PICHI(I) is computed as $P_i \chi_i$; for LØGC = .TRUE., it is calculated as $-\exp[\ln P_i + \ln (1-\chi_i)]$ to avoid possible underflow of the exponential function in $P_i = \exp(\ln P_i)$
ØNE	1.
QELAS	Energy transfer to the electron gas as a result of elastic collisions (cal/cm^3 sec)
QF	$\dot{q}_{fi}$ in eq. I(321b)
QR	$\dot{q}_{ri}$ in eq. I(321b)
R	Effective nozzle radius for loss of resonance radiation
RCALC	Logical indicator; if .TRUE., R has already been computed for current flow point
RKF	k_f, forward rate constant
RKR	k_r, reverse rate constant
SUMF	$\sum_{j=1}^{n} \nu_{ij} (\ln \rho + \ln \gamma_j)$
SUMR	$\sum_{j=1}^{n} \nu'_{ij} (\ln \rho + \ln \gamma_j)$
TAU	Optical thickness based on R for resonance radiation
TEP	Electron temperature, T_e (^oK)

TEQ	T_e^3
T6	$\sum_{j=1}^{n} \nu_{ij} \ln \gamma_j$
T7	$\sum \gamma_j$ over the third-body species for a reaction
VEJ	Collision frequency factor ν_{ei} in the calculation of energy transfer to the electron gas by elastic collisions
VIC	Electron-ion elastic collision frequency (ν_{ei}) divided by the ionic species concentration γ_j
VIC1	$(8/3)\sqrt{\pi/m_e}\ e^4 N_0/(2k)^{3/2}$, where m_e is the electron mass (g), e is the electronic charge (esu), N_0 is Avogadro's constant, and k is Boltzmann's constant
VIC2	$k^3/(\pi e^6)$
VNC	$(4/3)\sqrt{8k/(\pi m_e)}\ N_0$
VNCC	Electron-neutral collision frequency (ν_{ej}) divided by the neutral species concentration (γ_j)
Y	Radius of first channel profile, cm
Z	Radius of second channel profile, cm
Z1	$\ln$ u, natural logarithm of flow velocity in cm/sec
Z2	R_0T', where T' = T for KR = 1 and T' = T_e for KR = 2 (cm^3 atm/mole)
Z2F	R_0T_0 (cm^3 atm/mole)
Z3	$\rho R_0T'$ (see Z2)

4.78 Subroutine CXSECT

I(J)	New index of species for which old index is J
J	Second index of species pair
L	First index of species pair
M	Temporary variable

4.79 Subroutine DERIVS

CPJE	Temporary storage for specific heat of the electrons
CTSAVE	Temporary storage for the nondimensional gas temperature T/T_0
DELI1(L)	δ_{i1}^* for the Lth profile (eq. I(424))
DELI2(L)	δ_{i2}^* for the Lth profile (eq. I(424))
DELØ1(L)	δ_{o1}^* for the Lth profile (eq. I(424))
DELØ2(L)	δ_{o2}^* for the Lth profile (eq. I(424))
DEN	Denominator in eq. I(424)
HJE	Temporary storage for nondimensional enthalpy of the electron gas (H_e/R_0T_0)
I	DØ index over species
ITRY	Counter for iteration to obtain self-consistent solution for the boundary layer displacement thickness (Section 7.6 of Vol. I)
K	DØ index over profiles
SENTE	Temporary storage for nondimensional entropy of the electron gas (S_e^o/R_0)

4.80 Subroutine DSMSØL

A — Matrix initially containing the coefficients and constants for a system of linear equations to be solved:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1i}x_i = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2i}x_i = b_2$$

$$\vdots$$

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ii}x_i = b_i$$

The coefficients a_{jk} are stored in the locations A(j,k), for j=1 to i and k=1 to i. The constant terms b_j are stored in A(j,i+1), for j=1 to i. The solution values x_k are written into the locations A(k,i+1) for k=1 to i, that is, the locations originally occupied by the b_j.

I — Number of equations in the system to be solved

J — DØ index running from 1 to I

K — DØ index running from 1 to I

KS — Error indicator; nonzero value indicates singular matrix of coefficients

L — Counter for matrix elements during rewriting of the matrix in close-packed form

LM — First index for close-packed form of matrix

LN — Number of columns already filled in close-packed matrix

LN1 — Column index (second index) in close-packed matrix

M1 — I + 1

N	First dimension of A in the dimension statement in the calling routine
N1	N + 1
RNAME	Routine name for calling DUMP routine

4.81 Subroutine DUMP

RNAME	Name of routine from which DUMP was called

4.82 Subroutine DUMPEX

No local variables.

4.83 Subroutine ELCØND

I	Species index
SIGMA	Electrical conductivity in mhos/cm
SSIG	$(e/k)^2/\sqrt{T}$

4.84 Subroutine ELTIME

CALLED	Logical indicator, initially .FALSE., set to .TRUE. when ELTIME is first called
DET	Elapsed time since last printed time (seconds)
ET	Elapsed time from beginning of execution (seconds)
ETM	Elapsed time from beginning of execution (minutes)
ETØ	ET argument value passed from calling routine
ETP	Previous value of ETØ
I	Elapsed time from start of execution in milliseconds

IP	Control for printing the time; zero value suppressed the print

4.85 Subroutine EPART

EF	Energy gained by the electron gas in the forward direction of a reaction (cal/mole)
ER	Negative of the energy gained by the electron gas in the reverse reaction (cal/mole)
E0	Reaction energy to be partitioned between radiative losses and heating of the electron gas, ϵ_0 (cal/mole)
IR	Reaction index in the list of reactions for the current gas model
IT	Indicator (ITR) of the rule for partitioning the reaction energy in the current reaction
QF	Energy lost by radiation in the forward reaction (cal/mole)
QR	Negative of energy lost by radiation in the reverse reaction (cal/mole)
R0	Universal gas constant, cal/mole-^{o}K
TE	Electron temperature (^{o}K)
THR	$3R_0/2$, where R_0 is the universal gas constant in cal/mole-^{o}K

4.86 Subroutine EQCALC

CTSAVE	Variable used to save and restore the initial value of the nondimensional temperature CT
I	DØ index
IL	I + ISC

J	DØ index
K	DØ index
L	I + ISC
N	Iteration counter for Newton-Raphson solution of equations I(233)
P	Pressure (atm)
RNAME	Routine name for calling DUMP routine
T	Temperature (^{o}K)
ZA	Newton-Raphson correction factor ($1 + h_n^r$) to mole fractions, eq. I(235)

4.87 Subroutine EXACT

AAA(L,M)	Array of coefficients and constants for system of linear equations
DGJ(J)	$d\gamma_j/dx$
ECØEF	$1/(\rho u R_0 T_0)$
GJ(J)	γ_j, species concentrations, moles/g
I	DØ index
J	DØ index over species
K	I - ISC
NE	Number of equations to be solved for rates of change of dependent variables during the nonequilibrium integration
SUMTDG	$\sum_{j=1}^{n} \frac{T}{T_0} \frac{d\gamma_j}{dx}$ for chemical nonequilibrium models; $\frac{T_e}{T_0} \frac{d\gamma_e}{dx} + \sum_{j=2}^{n} \frac{T}{T_0} \frac{d\gamma_j}{dx}$ for electronic nonequilibrium models

SJ	$-1 + \nu^{*}_{i-c} + \frac{H_i}{R_0 \bar{T}} - \sum_{j=1}^{c} \bar{\nu}_{i-c,j} \frac{H_j}{R_0 \bar{T}}$
U	$(u/u_s)^2/W_0$

4.88 Function EXP

EXP	Value of function
X	Subroutine argument

4.89 Subroutine FINDX

A	Geometric area ratio (from calling routine)
AR	Geometric area ratio based on a trial value of X
DADX	Derivative of geometric area ratio with respect to X (cm^{-1})
DZDX(I)	Slope of Ith profile in channel
ERA	Error in geometric area ratio AR at current value of X
ERRL	Smallest negative error in AR for any previous trial value of X
ERRU	Smallest positive error in AR for any previout trial value of X
IENTRY	Indicator of entry point; 1 for FINDX, 2 for FINDXC
MBL	Index (1 or 2) of the profile whose ordinate varies most rapidly downstream of the throat in a channel
N	Iteration counter

RNAME	Routine name for call to DUMP routine
UPDØWN	Indicator; -1.0 if upstream solution is desired, 1.0 if downstream solution is desired
UPPER	Logical indicator; .TRUE. if a positive ERA has been found at some trial value of X
V	Trial value of |X| (cm)
VL	Trial value of |X| corresponding to error ERRL
VØ	Trial value of |X| in previous iteration step
VU	Trial value of |X| corresponding to error ERRU
X	Final solution for X (cm)
Z(I)	Ordinate of the Ith profile in a channel

4.90 Subroutine FRØZEQ

AR	Common variable, used here for the geometric area ratio
AS	Sound speed (cm/sec)
BLVAR(I)	Array equivalenced to the contents of common block /BLOUT/
BLVARØ(I)	BLVAR values at preceding flow point
BLVARS(I)	BLVAR values at the current regular flow point; used in interpolating at a model point
CLNMA	Logarithm of the molecular weight in the reservoir
CTSAVE	Nondimensional temperature at the regular flow point following a model point

CTSTAR	The nondimensional temperature (CT) value at the flow point preceding the throat, in solutions including the boundary layer
CT1	Second most recent trial value of CT in the iteration to find the conditions at a model point
CT2	Most recent trial value of CT in iteration to find the conditions at a model point
CXSAVE	X coordinate of the regular flow point following a model point
DELBLØ(I)	For I = 1, 2, the nondimensional boundary layer displacement thickness for the Ith profile at the preceding flow point
DELBLP(I)	For I = 1, 2, the nondimensional boundary layer displacement thickness for the Ith profile at the point before the preceding flow point
DELT	Decrement in nondimensional temperature CT in calculation of frozen throat conditions
DELTSV	Variable used to save the input value of DELT1
DELTV	Current decrement in nondimensional temperature CT
ERRX	Position error in iteration to determine the conditions at a model point
FLAG	Indicator for upstream and downstream regions, for use by the geometry subroutines. Negative value indicates an upstream position, positive value a downstream position

F1,F2,F3 — Values of nondimensional mass flux in iteration to determine the frozen throat conditions

GJ(J) — Species concentrations, mole/g

I — DØ index

IPASS — Indicator for iteration to take the boundary layer displacement thickness into account in the determination of reservoir conditions. IPASS = 1 causes a return to the main program after the throat has been reached

IRSTRT — Counter for number of times solution has been restarted with a lower value of W to suppress instability due to coupling of the boundary layer and the inviscid solution

ISØLN — Indicator for type of flow solution:

ISØLN = 1	Frozen solution
ISØLN = 2	Equilibrium solution

ISØNIC — Indicator for region of flow solution:

ISØNIC = -1	Upstream region
ISØNIC = 0	Sonic point
ISØNIC = 1	Downstream region

ISWTCH — Counter for number of flow points beyond the throat

ITS — Index of the next specified test section diameter (TSDIAM)

J — DØ index over gas species

K — DØ index

LASTPT — Indicator for last point in flow solution (at CX = CXMAX)

M — DØ index over channel profiles

MØDLPT — Logical indicator; if .TRUE., a model point is being calculated

N — DØ index

NMPI	Counter for number of iterations to determine the conditions at a model point
ØNE	1.0
SKIP	Logical variable used to indicate that conditions at a model point have just been determined
SØLN(I,J)	Hollerith array
SWTCH	Logical indicator; if .TRUE., FRØZEQ has begun taking the displacement thickness into account in calculations of the flow-point position
S1	$-\frac{1}{W_0} \sum_{j=1}^{n} X_{j0} \ln X_{j0}$
S2	$-(\ln p_0)/W_0$
TSTEPC	Logical indicator for special control of temperature decrement far downstream, to prevent distance of successive flow points from becoming too large
T1,T2,T3	Nondimensional temperature values in iteration to determine frozen throat conditions
XFRAC	Distance from preceding flow point to model point, divided by distance from preceding flow point to current point
XLAST	Position of preceding flow point (cm)
XMØDEL	Model-point position (cm)
XMØD1	Position of the next model point in the geometric sequence of model points (cm)
XØ	Position of the preceding flow point (cm)
XP	Position of the point before the preceding flow point (cm)

X1	Position of the flow point corresponding to the second most recent trial value of CT in the iteration to find the conditions at a model point (cm)
X2	Position of the flow point corresponding to the most recent trial value of CT in the iteration to find the conditions at the model point (cm)
Z	F1 - F2 + F3 - F2

4.91 Subroutine GEØM

AN	Effective area ratio corresponding to the current trial value of ρ, based on the area ratio-density relation, eq. I(383)
A2	Square of the effective area ratio
ARG	$2C(A_e-1)/\alpha$
DADR	$\rho_0 \, dA_e/d\rho$, derivative of the effective area ratio with respect to nondimensional density
DD	Correction in nondimensional density
DELRHØ	Absolute value of the difference between ρ/ρ_0 and the nondimensional density at the throat, based on approximate solution near throat
DELSTP	Displacement thickness at the throat (cm)
D2LA	$[d^2(\ell n\, A_e)/dx^2]^*$, the second derivative of the logarithm of the effective area ratio at the throat
J	DØ index over the profiles in a channel
N	Iteration counter
NSUJ	Number of upstream sections in the curvefit to a profile

ØMZ	1. - Z
ØPHA	$1 + \alpha/2$, where α is the exponent in the density-area ratio relation I(383)
RNAME	Routine name for call to DUMP
SGN	-1 if dA_e/dx is positive, +1 if dA_e/dx is negative
Y	Consistency check on x and ρ values; if positive, x and ρ data are inconsistent
Z	$(\rho/\rho_0)^{\alpha}$

4.92 Subroutine GEØMAR

ARATIØ	Geometric area ratio
DAB	Difference between X and the x-coordinate of the center of a circular profile section
DERIVA	dA_g/dx, derivative of geometric area ratio (cm^{-1})
DRV(I)	Slope (derivative) of ordinate for Ith profile
DYDX	Slope of ordinate for first profile, after rescaling area
DZDX	Slope of ordinate for second profile, after rescaling area
I	Index of two profiles
IENTRY	Index of entry point: 1 - GEØMAR; 2 - GMAR; 3 - GMAR2; 4 - GMAR3
ISJ	Shape index for a profile section

J	Index for the sections in a profile
K	Index for the profile section in which X lies
NSM1	Number of sections in a profile minus 1
Q	Difference in ordinate between a point on a circular profile section and the circle center (cm)
RAD(I)	Ordinate of the Ith profile (cm)
RAT	Ratio of a profile ordinate to the ordinate at the throat
SN(I)	-1.0 for I = 1, + 1.0 for I = 2
SQTRSA	Square root of area rescaling factor RSA
S1(I)	Unrescaled geometric area ratio; or ratio of ordinate to ordinate at the throat for Ith profile
S2(I)	Derivative of geometric area ratio; or ratio of derivative of ordinate to ordinate at the throat for Ith profile (cm^{-1})
X	Axial coordinate in nozzle (cm)
Y	Ordinate of first profile (cm)
Z	Ordinate of second profile (cm

4.93 Subroutine INGAS

CAPQ(K)	Q_k, number of gram-atoms of the kth chemical element per mole of the cold gas
CCI(I)	Intermediate variable in the calculation of the molecular weights of species from atomic weights of elements

I	Species index
IL	Index running from c + 1 to n
J	Element index
K	Index running from 1 to c
L	Index running from c + 1 to n
SUM	Dummy variable used for accumulating summations

4.94 Subroutine INIT

I	Species index in the current gas model
II	Species index in the master list of species
J	Index of additional vibrational modes in triatomic molecules
K	Number of electronic levels for a species
L	Index of the electronic levels for a species
SPRP(M,II)	Array containing species property data in common /SPEC/
T1	Reservoir temperature T_0(°K)
T2	T_0^2
T3	T_0^3
T4	T_0^4

4.95 Subroutine KANDMU

I	Species index
I1	I-1

J	Species index
L	Index of property being calculated: L = 1 for translational thermal conductivity, L = 2 for viscosity
R	Weighting factor used in computing $\bar{a}^{(\alpha)}$ from eq. I(86)
U	Variable used to accumulate numerator in eq. I(86)
V	Temporary variable used in computing denominator in eq. I(86)
W	Temporary variable used in computing numerator in eq. I(86)
Y	Temporary variable, used for accumulating denominator in eq. I(86) and numerator in eq. I(85)
ZK(L)	$ZK(1) \cdot \sqrt{T}$ is the translational thermal conductivity in milliwatts/cm-^{o}K, $ZK(2) \cdot \sqrt{T}$ is the viscosity in millipoise

4.96 Subroutine KINT

I	Species index
I1	I-1
J	Summation index for sum over species pairs with J < I
L	Index of property being computed (see discussion of KINT in Sec. 3.28)
NI	Species index
NJ	Summation index for sum over species pairs with NJ > NI

N1	N + 1
ZKINT	ZKINT $\sqrt{T}$ is the internal thermal conductivity in milliwatt/cm-^{o}K

4.97 Subroutine LIST

CAPQ(I)	Q_k, number of gram-atoms of the kth chemical element per mole of the cold gas
CØM1(I,J)	Hollerith array for output (see DATA statement)
CØM2(I,J)	Hollerith array for output (see DATA statement)
CXMAXI	Distance beyond throat at which case will be terminated (inches)
I	DØ index
J	DØ index
K	DØ index
KDQ1	KDIM + 1
NWXI	Number of specified distances WXI from the leading edge of a wedge model
SBJ(J)	b_j, eqs. I(51)
SHJAP(J)	Formation enthalpy of the Jth species, cal/mole
THEVP(J)	Characteristic vibrational temperature for the Jth species, ^{o}K
WØRD(I,J)	Hollerith array for output (see DATA statement)

4.98 Subroutine MATINV

A	Original matrix; replaced by inverse
AA	Temporary storage used in interchanging columns
B	Temporary storage used in interchanging rows
BB	Temporary storage used in reduction of off-diagonal elements
BIG	Pivotal element in a column (largest absolute magnitude)
I	DØ index
IV(K)	Index array for recording interchanges of rows
J	Temporary storage used in interchanging IV elements; also DØ index
JA	Index of current row; reset to index of next row
JB	Index of preceding row
JJ	Index of current row
K	First index of pivotal element in a column
K1	K + 1
L	DØ index
M	Equal to MM
MM	Size of matrix to be inverted
M1	M - 1
NN	First dimension of A in dimension statement of calling routine

RNAME	Routine name for calling DUMP routine

4.99 Subroutine MØDEL

ALPHA	Ratio of the current estimates of the pressure p_2 and the density ρ_2 behind the shock
CATM	Conversion factor from atm to dyne/cm^2
CKS1	Parameter characterizing low density effects on the stagnation-point heat flux, ξK^2 (per foot)
CM1	Mean molecular weight of the gas ahead of the shock, g/mole
CTSAVE	Temporary storage for nondimensional free stream temperature CT
DET	Computer time used in model calculations, seconds
DISC	Discriminant in quadratic solution for the pressure behind the equilibrium normal shock
ELN(I)	Exponent of Lewis number in Fay-Riddell heat transfer correlation (equilibrium boundary layer. I = 1, frozen boundary layer I = 2)
EMF	Temporary storage for electron mole fraction at stagnation point, to permit output of electron concentration
EPS	Density ratio $\epsilon = \rho_1/\rho_2$ across normal shock
EPSØLD	Previous estimate of ϵ
EQUALS	Hollerith equality sign used in output
ERRHI	Most recent positive error in the nondimensional enthalpy in iterative solutions for conditions behind the normal shock and at the inviscid stagnation point

ERRLØ	Most recent negative error in the nondimensional enthalpy
ERZ	Current error in the nondimensional enthalpy
ET	Elapsed time since the beginning of the run, seconds
ETO	Elapsed time since the beginning of the run when MØDEL is entered, seconds
GJ(I)	Species concentrations, moles/g
HCF	Enthalpy of formation of the gas mixture at the stagnation point
HCØN	Conversion factor (RT_0/W_0) from nondimensional enthalpy to enthalpy in cal/g
HD	Nondimensional kinetic energy per unit mass of the gas ahead of the shock, $(u/u_s)^2/2$
HE	Free-stream static enthalpy (cal/g) ; used in flat-plate calculation
HI	Logical variable; if .TRUE., a positive enthalpy error has already been obtained in the iteration
HR	Recovery enthalpy (cal/g) in flat-plate calculation
HRATIØ	Ratio of the dissociation enthalpy to (h_e-h_w)
HSTAG	Enthalpy at the inviscid stagnation point, cal/g
HSTAR	Reference enthalpy 'cal/g) in flat-plate calculation
HW	Gas enthalpy at the wall temperature of the stagnation point model, cal/g

H2	Nondimensional enthalpy behind the normal shock
I	DØ index
ICØUNT	Iteration counter
ISØLN	Index specifying the two types of normal-shock solution (equilibrium ISØLN = 1, frozen ISØLN = 2)
ISS5P1	Number of lines of output required for the mole fractions at the inviscid stagnation point
J	Index of chemical elements
K	DØ index
L	Model-shape index
LIMSCØ(I)	For I = 1, 2, stored values of LMSC for the equilibrium and frozen normal shock solutions
LIM1	Lower limit of implied DØ loop for output of mole fractions
LIM2	Upper limit of implied DØ loop for output of mole fractions
LMSC	Number of data items from the SCØUT array to be printed for the current normal shock solution
LØ	Logical indicator; if .TRUE., a negative enthalpy error has already been obtained in the iteration
N	DØ index
NEL	Number of chemical elements in a species

PC	$4\ \rho_1^2\ u_1^2$, atm-g/cm^3
PRW	Prandtl number at the model wall temperature
PS	Stagnation pressure, atm
PT	$p_1 + \rho_1 u_1^2$, atm
P1	Pressure ahead of the shock, atm
P2	Pressure behind the normal shock, atm
QFRE(L)	Stagnation point heat flux times the square root of the nose radius for an equilibrium boundary layer (hemispherical model L = 1, flat-faced model L = 2), Btu/ft$^{3/2}$-sec
QFRF(L)	Stagnation point heat flux times the square root of the nose radius for a frozen boundary layer (hemispherical model L = 1, flat-faced model L = 2), Btu/ft$^{3/2}$-sec
QP	$p_1 T_2 / T_1$
QS	Factor used in calculating QFRE and QFRF
QSRI	Stagnation point heat flux times the square root of the nose radius based on the SRI approximation, Btu/ft$^{3/2}$-sec
QUANT(I)	Hollerith names of output quantities in SCØUT array
RHØSTR	Density at reference temperature in flat-plate calculation
R1	Density ahead of the shock, g/cm^3
R1U12	$\rho_1 u_1^2$, atm
R2	Density behind the shock, g/cm^3
SCOUT(I)	Array used for output of model conditions

SHV	Array equivalenced to common block /STAG/ to permit setting these data in the DØ loop over ISØLN
SØD(L)	Shock standoff distance divided by model radius (hemispherical L = 1, flat-faced L = 2)
SQTE	Square root of the shock density ratio, $\sqrt{\epsilon}$
SQ2F(K)	Ratio of the stagnation-point heat flux for the current model geometry to that for an axisymmetric model (two-dimensional model K = 1, axisymmetric K = 2)
THI	T_2 value corresponding to enthalpy error ERRHI, ^oK
TITLE2	Hollerith data used in output
TLØ	T_2 value corresponding to enthalpy error ERRLØ, ^oK
TSTAR	Reference temperature in flat-plate calculation, ^oK
T1	Temperature ahead of the shock, ^oK
T2	Temperature behind the shock, ^oK
U1	Velocity ahead of the shock, cm/sec
VGP(L)	Velocity gradient parameter $(R_b/u_1)(du_e/dx)_s$ (hemispherical model L = 1, flat-faced L = 2)

4.100 Subroutine NEWRAP

GJ(I)	Species concentrations, mole/g
I	DØ index; also J + ISC
IL	I + ISC

J	DØ index
K	DØ index
L	I + ISC
LL	ISSNR + 1
M2	ISC + 2
NØDEPS	Logical indicator; .TRUE. when the gas model contains no dependent species
RNAME	Routine name for calling DUMP routine
ZB	Correction factors $1+h_n^r$ to the mole fractions, eq. I(260)
ZC	Correction factor $1+h_{c+1}^r$ to the pressure

4.101 Subroutine NEXTMP

ITS	Index of the next specified test section diameter
ØNE	1.0
XMØD1	Coordinate of the next model point in the geometric sequence of model points, cm
XMØD2	Coordinate of the model point at the next specified test section diameter, cm
XMØDEL	Coordinate of the next model point, cm

4.102 Subroutine NØNEQ

AB	Temporary storage location used in interchanging rows of β_{ij} matrix during determination of matrix rank

ADCH	Absolute value of a $\delta\chi_i$ value
ALØGT	ℓn (10000)
BLNEAR(M)	Array equivalenced to the first 10 locations of common block /BLNE/
CA	BETA(M,I)/BETA(IX,I)
CAISV(I)	Storage for the correct values of CAI(I) to allow resetting CAI(I) when the artificial increase of reaction rate constants has been used to prevent premature switching from the perturbation technique to the numerical integration
CI(K)	Array specifying elemental composition of the gas mixture in the reservoir; CI(K) is the number of gram-atoms of the Kth element per gram of mixture
CJ	Number of gram-atoms of an element per gram of mixture at the end of a successful integration step
DCA(K)	$\frac{1}{(\gamma_k')^2} \sum_{j=1}^{c} \delta c_j A_{jk}$ in eq. I(346c)
DCHMAX	Largest $\lvert\delta\chi_i\rvert$ for any reaction
DCHMIN	Smallest $\lvert\delta\chi_i\rvert$ for any reaction
DCI(K)	δc_k, correction in concentration of Kth chemical element required at the end of an integration step (eq. I(344), g-atom/g
DELBLB(L)	Nondimensional boundary layer displacement thickness for the Lth profile, $\delta_L^*/R0$, at the start of an integration step

DELT	Nondimensional temperature interval in the perturbation solution, $\lvert\Delta T\rvert / T_0$
DELTSV	Input value for DELT1
DQEM	Maximum allowable change in the energy transfer to the electron gas in one integration step
DXM	Largest allowable step size for the first integration step after the switch from the perturbation solution
DXØLD	Step size Δx in the preceding integration step (cm)
DXSAVE	Temporary storage for saving and restoring the integration step size when a reduced step is used to calculate the conditions at a model point (cm)
GJ(J)	Concentration of the Jth species, γ_j, mole/g
GSQ(J)	Square of GJ(J)
HDXSAV	One-half of DXSAVE, cm
I	DØ index
ICØUNT	Counter for number of times a step has been restarted with reduced step size
IERR	Error exit indicator (see subroutine listing)
IF(N)	Array for storing and printing out IFAIL values for successive restarts of a step (see glossary for common block /TNERK/)
II	I + ISC
IMAX	Index of the reaction with the largest $\lvert\delta\chi_i\rvert$

IRA	Counter for number of times the artificial increase in rate constants has been used to prevent premature switching from the perturbation technique to numerical integration
ISMCP1	n - c + 1
ISTEPS	Counter for number of integration steps
ISUBD	Counter for number of times the temperature interval ΔT has been cut in half in an attempt to find a point in the perturbation solution satisfying the second inequality of eq. I(381)
ISUPR(I)	Array of indices of the reactions which are suppressed when the decrease of a minor species is found to control the integration step
IS5	Input value of ISW5B
ITAPE	Number of records written on the binary output tape since subroutine THRØAT was called
ITS	Index of the next model point defined by a specified test section diameter
IX	Rank of the β_{ij} matrix
I2	Index used in calculating the rank of β_{ij}
J	Index used for various purposes
K	Index used for various purposes
IA	Index used in calculating the rank of β_{ij}
L	DØ index
M	Index used in calculating the rank of β_{ij}

N	Index used in calculating the rank of β_{ij}
NSUPR	Number of reactions suppressed when the decrease of a minor species is found to control the integration step
ØNE	1.0
QEØLD	Energy transfer to the electron gas at the preceding flow point, cal/cm^3 sec
RATIØ	Ratio of the smallest and largest $\|\delta\chi_i\|$ values
RATIØ2	Factor by which the forward rate constant is increased in the artificial increase of a reaction rate to prevent premature switching from the perturbation technique to the numerical integration
RNAME	Subroutine name for call to DUMP routine
SCD	Factor by which the step size Δx is divided when an integration step fails and has to be restarted
SDGJ(J)	Change in the concentration γ_j of the Jth species in an integration step
TENTH	10000.
TEP	Electron temperature (^{o}K)
TP	Gas temperature (^{o}K)
TSZ	Temporary storage for TPRINT
XMØDEL	Position of next model point, cm
XMØD1	Coordinate of the next model point in the geometric sequence of model points, cm

XMSET	Logical indicator, set to .TRUE. when NEXTMP is first called to determine XMØDEL
ZZZ	Temporary storage

4.103 Subroutine NRMAX

DELT	Decrement of nondimensional temperature
F1,F2,F3	Mass flux values in iteration to determine equilibrium sonic flow conditions
GJ(J)	Species concentrations, mole/g
I	DØ index over species
J	DØ index over species
T1,T2,T3	Nondimensional temperature values in iteration to determine the equilibrium sonic flow conditions
Z	F1 - F2 + F3 - F2

4.104 Subroutine ØUT1

ANAM	Hollerith word "ARAT"
ARAT	Geometric area ratio
ASØLN(I,J)	Hollerith array containing names for types of solution (see DATA statement)
ASTRSK	Hollerith word filled with asterisks
CHDIMS(L)	Transverse dimensions of a channel at the current flow point, inches
CP	Molar heat capacity at constant pressure, divided by R_0

DARAT	Derivative of the geometric area ratio
DEL	Electron density, electrons/cm^3
DELSTP(L)	Displacement thickness on the Lth profile, cm
DGJ(I)	$d\gamma_i/dx$, derivatives of species concentrations, mole/ g-cm
DLGDX(I)	$d\ln\gamma_i/dx$ for the Ith species
DNAM	Hollerith work "DIAM"
EMF	Mole fraction of the electrons
FAREA	Effective cross-sectional area of the flow at the throat, cm^2
FLX	Mass flux, lbm/ft^2-sec
FVØ2(I,J)	Array used for output of SN and XSN
GAMMA	Ratio of specific heats, $\gamma = c_p/c_v$
GJ(I)	Species concentrations, mole/g
HØUT	Enthalpy, Btu/lb
HW	Enthalpy at the nozzle wall, Btu/lb
HO	Stagnation enthalpy, Btu/lb
I	DØ index
IBL	Index such that FVØUT(IBL) is the boundary layer displacement thickness on the nozzle wall or on the broad face of a channel
IFLAG	Indicator for entry points to subroutine IFLAG = 1 Entry ØUT1 IFLAG = 2 Entry ØUT2(ISØLN)

II	Index 7 + 10*I
INEQP1	INEQ + 1
ISØLN	Indicator for type of solution ISØLN = 1 Frozen ISØLN = 2 Equilibrium ISØLN = 3 Nonequilibrium
ISS5P1	(ISS-1)/5+1, number of lines of output required for printing the species mole fractions
IS6	|ISW6B|
IX	Counter of records written on tape 8 for a given flow solution
I1	Index of the first FVØUT(I) element to be printed on a line
I2	Index of the last FVØUT(I) element to be printed on a line
I6	Counter; number of flow points printed since last output of reaction rate data
J	DØ index
K	DØ index
L	DØ index
LIM1	Index of first species mole fraction to be printed on a line
LIM2	Index of last species mole fraction to be printed on a line
PØUT	Pressure, atm
PZ	3.1415927

RØUT	Density, lb/ft^3
SØUT	Entropy, $Btu/lbm-^oR$
STEP(I)	Hollerith array (see DATA statement)
STFAC	$1/\ [\rho u(h_0-h_w)]$, $(Btu/ft^2-sec)^{-1}$
TFLØW	Total mass flow, lb/sec
TNNAM(I)	Hollerith array of variable names (see DATA statement)
TNØUT(I)	Array used for output of quantities for electronic nonequilibrium model
TØUT	Temperature, ^oK
TYPSLN(I,J)	Hollerith array used to indicate whether nonequilibrium solution is based on perturbation technique (INEQ = 0) or numerical integration (INEQ = 1)
UØUT	Velocity, ft/sec
VARNAM(I)	Array of Hollerith names for output variables (see DATA statement)
VARN2(I)	Array of Hollerith names for N, XSN
WNAM	Hollerith word "WIDTH"
Y	Distance from nozzle axis to nozzle surface (or to first profile in the case of a channel)
Z	Distance from axis to second profile of a channel

4.105 Subroutine PERT

I	DØ index
J	DØ index

K	DØ index
Z	Temporary storage used in computing sums
Z1	Temporary storage used in computing sums

4.106 Function PIØMEG

I	Index of entries in cross section table
PIØMEG	Cross section value returned to the calling routine
TE	Electron temperature, ^{o}K

4.107 Subroutine PRØP

J	DØ index over species
RNAME	Routine name for call to DUMP
S4	$\frac{1}{R_0} \sum_{j=1}^{n} \gamma_{j0} s_j^{o}$

4.108 Subroutine PRTA

AS	Speed of sound, cm/sec

4.109 Subroutine PUTQIN

I	First index of species pair
IDBG	Count to determine when debug output is produced
J	Second index of species pair
K	Option KQ to be used in step L of the computations
KK	Type of cross section; first index of Q(KK,I,J)

L	Index of current step in cross section computations
IQ	Index of species pair in IQ,JQ array
IQ1	Index of first species pair in IQ, JQ array to be used in step L of computations
IQ2	Index of last species qair in IQ, JQ array to b used in step L of computations
M	Index of first parameter in V array to be used in step L of computations
ØM(KK)	Variable for temporary storage of cross sections computed in current step. ØM(1)= $\bar{\Omega}^{(1,1)}$, ØM(2)= $\bar{\Omega}^{(2,2)}$, ØM(3) = B* $\bar{\Omega}^{(1,1)}$
WRØTE	WRØTE = .TRUE. indicates that debug output has been produced in the current call to PUTQIN
X	Electron partial pressure in atmospheres

4.110 Subroutine QCØUL

C	Control variable. For C > 0, ℓn (f Λ) is computed from eqs. I(100); for C $\leq$ 0, ℓn(f Λ) is set to 1
ØM1	$0.8Q_c$ (see eq. I(100b))
X	Electron partial pressure in atmospheres
Y	20.91Y = (f Λ)^2 (see eqs. I(100c) and I(100d))

4.111 Subroutine QEX

A	Coefficient A in eq. I(104) for the exchange cross section

B	Coefficient B in eq. I(104) for the exchange cross section
ØM(K)	Contribution to cross sections from option; ØM(1) = $\bar{\Omega}^{(1,1)}$ ØM(2) = $\bar{\Omega}^{(2,2)}$ ØM(3) = B*$\bar{\Omega}^{(1,1)}$
V(M)	Input parameters VV for option
Y	Quantity A - ½ B ln (T/M)
ZM	Molecular weight

4.112 Subroutine QEXP

A	Coefficient A/k in °K for the exponential potential
ALPHA	Quantity α
N1	Position of first entry in tabulated collision integrals for the exponential potential
ØM(K)	Computed cross section values (see subroutine QEX)
RHO	Characteristic length ρ for the exponential potential in Å
V(M)	Input parameters VV for option

4.113 Subroutine QINTRP

A	Value of independent variable to be used in interpolation
B(J)	Interpolated value of Jth dependent variable
I	Interpolation is performed between entries I and I + 1 of the table
J	Index of dependent variable
N	Number of entries in table
N1	Location of first entry in table
N2	N2 + 1 is the location of the last entry in table

4.114 Subroutine QLJ

EPSLN	Well depth parameter ϵ/k for the Lennard-Jones potential, in ^{o}K
N1	Position of first entry in tabulated cross sections for the Lennard-Jones potential
ØM(K)	Computed cross section values (see subroutine QEX)

SIGMA	Collision diameter σ for the Lennard-Jones potential, in Å
TSTAR	Nondimensional temperature, kT/ϵ
V(M)	Input parameters VV for option

4.115 Subroutine QMIX

I	First index of species pair
J	Second index of species pair
K	Type of cross section, i.e., first index of Q(K,I,J)
LQ	Index of species pair in IQ, JQ array
LQ1	Index in IQ, JQ array of first species pair to be used in computation
LQ2	Index in IQ, JQ array of last species pair to be used in computation
SQT(K,I)	$\frac{1}{2}\sqrt{Q(K,I,I)}$

4.116 Subroutine QREPP

ITL	Location in ØMEGA1 array where data are stored
ØM(K)	Computed cross section values (see subroutine QEX)
VV(M)	Input parameters VV for option

4.117 Subroutine QSAME

C	Constant by which previously computed cross sections are multiplied
I	First index of species pair for previously computed cross sections

J	Second index of species pair for previously computed cross sections
ØM(K)	Computed cross section values (see subroutine QEX)
VV(M)	Input parameters VV for option

4.118 Subroutine QTAB

A	Constant multiplying factor for tabulated cross sections
N1	Location of first entry in tabulated cross section data
NL	Number of entries in tabulated cross section data
ØM(K)	Computed cross section values (see subroutine QEX)
V(M)	Input parameters VV for option

4.119 Subroutine Q11

LQ1	Location in IQ, JQ list of the first species pair to which the ramp function is to be applied
LQ2	Location in IQ, JQ list of the last species pair to which the ramp function is to be applied
T0	Temperature in ^{o}K beyond which cross sections are set to zero
T1	Temperature in ^{o}K beyond which cross sections are unchanged
VI(M)	Input parameters VV for option

VV(M)	Input parameters for subroutine Q14

4.120 Subroutine Q12

I	First index of first previously computed species pair
J	Second index of first previously computed species pair
K	First index of second previously computed species pair
L	Second index of second previously computed species pair
M	Index indicating cross section type
ØM(M)	Computed cross section values (see subroutine QEX)
VV(N)	Input parameters VV for option

4.121 Subroutine Q13

C	Constant by which cross section is multiplied
I	First index of species pair
J	Second index of species pair
K1	Type of cross section to be computed
K2	Type of cross section to be used in computation
IQ	Location of species pair in IQ, JQ list
IQ1	Location in IQ, JQ list of first species pair to be used

LQ2	Location in IQ, JQ list of last species pair to be used
VV(M)	Input parameters VV for option

4.122 Subroutine Q14

C2	Constant applied to $\bar{\Omega}^{(2,2)}$
C3	Constant applied to $B^* \bar{\Omega}^{(1,1)}$
I	First index of species pair
J	Second index of species pair
LQ	Location of species pair in IQ, JQ list
LQ1	Location in IQ, JQ list of the first species pair to be used with the set of multiplying factors
LQ2	Location in IQ, JQ list of last species pair to be used with the set of multiplying factors
VV(M)	Input parameters VV for option

4.123 Subroutine RADIUS

AG	Geometric area ratio
AGJ	Factor proportional to r^{2j} in eq. I(171)
ITYPE	Indicator for type of nozzle geometry ITYPE = 1 Two-dimensional nozzle ITYPE = 2 Axisymmetric nozzle ITYPE = 3 Channel
L	Index of profiles (1 for a nozzle, 1 or 2 for a channel)

M	NPR(L)
NPR(L)	2 for L = 1; 1 for L = 2
R	Distance from axis to profile
RATIØ	R/R0
X	Axial coordinate in nozzle or channel, cm
YZ(L)	Distance from axis to Lth profile in a channel

4.124 Subroutine READ

AL	Number of atoms of the Kth element per molecule of the Ith species, as obtained from the SPRP array
ASYM(IZ)	Hollerith chemical symbol for the element with atomic number IZ
CAPQ(I)	Number of gram atoms of the Ith chemical element per molecule of the cold gas
CP(I,J)	Array for addressing the channel data in common block /CHAN/
CSMW	Molecular weight of one of the cold species
DAL	Error in rounding A to an integer LP
DUM(I)	Array for zeroing common block /TNCE/
ELNAME(K)	Hollerith name for the Kth element in one of the cold species
EPRP(I,J)	Array for addressing the element data in common block /ELEM/

EXPMP	Exponent used in computing the factor (FACMP) used to generate a geometric sequence of distances x at which model calculations will be done
FNMP	Number of model points in geometric sequence, minus 1
GPRP(I,J)	Array for addressing the gas model data in common block /MIXT/
HWINØV	Estimate of stagnation enthalpy based on Winovich equation, Btu/lb
I	DØ index
IES	Index of an element in a species, in the master list of elements
IGMV	Number of electronic levels in a species
IGS	Input value of IGAS
II	Index used in place of subscripted integer variables
IPRFL	Index used in place of subscripted integer variable, NPRØFL(J)
ISH	A value of ISHAPE(K,J)
ISKUR	Index of a third-body species in the master list of species
IUPDI	Initial value of IUPD
IZ	Atomic number of an element defined in the input using EEPRP
J	DØ index
JDP1	JDIM + 1

JJ	Index used in place of a subscripted integer variable, JCS(I)
JPS	Index of a species on the product side of a reaction, in the master list of species
JRS	Index of a species on the reactant side of a reaction, in the master list of species
K	DØ index
KK	Index of an element in a cold species, in the master list of elements
L	DØ index
LL	Index used in obtaining the PARAM array from the data stored in the ZPRP array
LP	Integer obtained by rounding AL
NATØM	Number of atoms of an element in a molecule of a species
NELS	Number of elements in a species
NK	Number of third body species for a reaction
NØZZLE	Index of the nozzle profile in the precoded geometry fits
NPSS	Number of product species for a reaction
NRSS	Number of reactant species for a reaction
NSECTD(J)	Number of downstream sections in the profile curvefit for the Jth profile of a channel or nozzle
NSECTJ	Index used in place of dimensioned integer variable, NSECT(J)

NSM1	NSECT(J) - 1
ØNE	1.0
PLUS	Hollerith + sign
RPRP(I,J)	Array for addressing reaction data in common block /REAC/
SBJ(J)	b_j; eqs. I(51)
SETIGS	Logical indicator; .TRUE. if automatic air model selection option is in effect
SHAPD	Array of Hollerith data for describing the shapes of nozzle profile sections
SHJAP(I)	Formation enthalpy of the Jth species, cal/mole
SPNAME(I)	Array of nominal species names for species defined in the input
SPRP(I,J)	Array for addressing species data in common block /SPEC/
SY1(K)	Array of Hollerith left parentheses
SY2(K)	Array of Hollerith right parentheses
THEVP(I)	Vibrational temperature for the Ith species, ^{o}K
TNEP(I,J)	Array for addressing thermal nonequilibrium data in common block /TNE/
WØRD(I,J)	Hollerith data for output of the words "two-dimensional" and "axisymmetric"
WØRD2(I)	Hollerith data for output of the words "gap" and "diam"

ZPRP(I,J) Array for addressing the geometry data in common block /NØZZ/

4.125 Subroutine RESET

I Elapsed time since the beginning of the run in milliseconds

ICPU CPU time remaining before TIME.GØ, milliseconds

IN Entry point indicator: IN = 1, RESET; IN = 2, TIME

IZERØ CPU time remaining before TIME.GØ at call to RESET, milliseconds

4.126 Subroutine RESTMP

AM(I,J) Matrix of coefficients for system of equations in quadratic interpolation for the reservoir temperature

BM(I) Constant terms for system of equations in quadratic interpolation for the reservoir temperature

CØNST $R_0(m*/p_0)^2$, in which R_0 is the universal gas constant, $m*$ the known sonic mass flux, and p_0 the reservoir pressure, all expressed in cgs units

CTAPP Previous estimate of reservoir temperature, ^oK

CTSAVE(I) For I = 1 to 3, successive estimates of the reservoir temperature, the value for I = 1 being the most recent, ^oK

CTSV Temporary storage for the reservoir temperature estimate used in the current iteration, ^oK

DCT The difference between the two most recent estimates of the reservoir temperature, ^oK

DCTAP	Absolute value of the difference between the current and improved estimates of the reservoir temperature, ^{o}K
DCTAPØ	DCTAP in the previous iteration step, ^{o}K
DMC	Intermediate quantity in quadratic interpolation for the reservoir temperature, ^{o}K
D1,D2	Intermediate quantities in linear interpolation for the reservoir temperature, ^{o}K
F(I)	For I = 1 to 3, successive estimates of the quantity $W_0 \cdot (SM)^2$, the value for I = 1 being the most recent; W_0 is the molecular weight in the reservoir and SM the nondimensional sonic mass flux
H	Calculated value of reservoir enthalpy, cal/g
HERR	Allowable error in calculated stagnation enthalpy cal/g
HØ	Calculated stagnation enthalpy in previous iteration step, cal/g
I	DØ index
J	Index for shifting data in CTSAVE and F arrays
KS	Indicator for singularity of AM matrix
N	Iteration counter for main iteration
N2	Iteration counter for temperature calculation in enthalpy option
N3	Iteration counter for pressure calculation in enthalpy option

PCGS	Reservoir pressure, dyne/cm^2
PERR	Allowable relative error in reservoir pressure
PLAST	Previous estimate of reservoir pressure, atm
RGAS	Universal gas constant in cgs units
RNAME	Hollerith name for calling DUMP routine
SMERR	Allowable error in sonic mass flux, g/cm^2 sec
SMP	Calculated sonic mass flux, g/cm^2 sec
TERR	Allowable relative error in reservoir temperature
TLAST	Previous estimate of reservoir temperature, ^{o}K

4.127 Subroutine RNKT

DGJ(J)	d γ_j/dx, derivatives of species concentrations, mole/g-cm
DQMRK	Twice DQMAX; criterion value for allowable changes in energy transfer to the electrons, QDPE, cal/cm^3 sec
FK(J,K)	Array equivalenced to F1(J), F2(J), F3(J), F4(J)
F1(J)	f_1, derivative of the Jth dependent variable at the start of the integration step
F2(J)	f_2, derivative of the Jth dependent variable at x_2, y_2 (eq. I(394a))
F3(J)	f_3, derivative of the Jth dependent variable at x_3, y_3 (eq. I(394b))

F4(J)	f_4, derivative of the Jth dependent variable at x_4, y_4 where y_4 is given by I(394c) or I(405)
GJK(J,K)	Array equivalenced to GJ1(J), GJ2(J), GJ3(J), GJ4(J)
GJ1(J)	y_1, the Jth dependent variable at the start of the integration step
GJ2(J)	y_2, the Jth dependent variable at x_2 (eq. I(394a))
GJ3(J)	y_3, the Jth dependent variable at x_3 (eq. I(394b))
GJ4(J)	y_4, the Jth dependent variable at x_4 (eq. I(394c) or I(405))
IFØR	Indicator for type of formula used to compute GJ4(J); IFØR=1 for I(394c), IFØR=2 for eq. I(405)
ISTMNT	Indicator for locations in subroutine at which diagnostic dumps are written (see listing)
J	Index of dependent variables: J = 1 to n $\quad \gamma_j$ J = n + 1 $\quad T$ J = n + 2 $\quad T_e$ J = n + 3 $\quad h_0$
K	DØ index
LIM	Number of dependent integration variables
P(J)	Parameter ($-P \Delta x$) for the Jth dependent variable, eq. I(402)
QDPEB	Electron energy transfer (QDPE) at the start of the integration step, cal/cm^3 sec
SDGJ(J)	Change in γ_j over an integration step, mole/g

SDQ(J)	Change in Jth dependent variable over the integration step
S1	Temporary storage in computation of I(398a) for small $P\Delta x$
S2	Temporary storage in computation of I(398b) for small $P\Delta x$
S3	Temporary storage in computation of I(398c) for small $P\Delta x$
TE	Logical indicator, .TRUE. for chemical nonequilibrium models, .FALSE. for electronic nonequilibrium models
T1	Temporary storage in computation of I(398a) for small $P\Delta x$
T2	Temporary storage in computation of I(398b) for small $P\Delta x$
T3	Temporary storage in computation of I(398c) for small $P\Delta x$
XX	$\log_{10}(\|f_3-f_2\|)$
X1	F_1 (eq. I(398a))
X2	F_2 (eq. I(398b))
X3	F_3 (eq. I(398c))
X4	f_2-f_1
Y(K)	Array equivalenced to Y1, Y2, Y3, Y4
YY	$\log_{10} (\|f_2-f_1\|)$
Y1	$f_1 + Py_1$ in I(401)

Y2	$f_2 + Py_2$ in I(401)
Y3	$f_3 + Py_3$ in I(401)
Y4	$f_4 + Py_4$ in I(401)

4.128 Subroutine SHØCK

ALP	The quantity $\sin^2\sigma$ - B, where σ is the shock angle
B	The quantity $-\left[\frac{M^2+2}{M^2} + \gamma \sin^2\alpha\right]$, where M is the free-stream Mach number, γ the specific heat ratio, and α the body angle
BETA	The quantity ALP · $\sin^2\sigma$ + C
BMF	Free-stream Mach number, M
BM2	M^2, square of the Mach number
BM4	M^4
C	The quantity $\frac{2M^2+1}{M^4} + \left[\frac{1}{4}(\gamma+1)^2 + \frac{\gamma-1}{M^2}\right]\sin^2\alpha$
CD	Cosine of the body angle, $\cos\alpha$
C1	$\gamma+1$
C2	$\gamma-1$
C3	2B
D	$-\cos^2\alpha / M^4$
DEL	The body angle α ; inclination of the wedge surface to the direction of free-stream motion (degrees)
DELR	α in radians

DIS	$\sqrt{(ALP)^2 - 4 \cdot BETA}$
FL(I)	Array used for communicating results to the calling routine FL(1) shock angle FL(2) Mach number behind shock FL(3) Density ratio, ρ_2/ρ_1 FL(4) Static pressure ratio, p_2/p_1 FL(5) Static temperature ratio, T_2/T_1 FL(6) Total pressure ratio, p_{t2}/p_{t1} FL(7) Ratio, T_{stag}/T_1
FPU	Derivative dF/du of the function F(u)
FU	$F(u) = u^3 + Bu^2 + Cu + D$
GAM	Specific heat ratio γ
IERR	Argument whose value is set to 2 when Newton-Raphson iteration to find the shock angle does not converge
KIT	Iteration counter
RT	$\sin^2 \sigma$, where σ is the shock angle
SD	$\sin \alpha$, where α is the body angle
UG	Previous trial value of u in solution of cubic equation F(u) = 0
UN	Current trial value of u in solution of cubic equation F(u) = 0
U1	First solution for $\sin^2 \sigma$
U2	Second solution for $\sin^2 \sigma$
V1	$M^2 \sin^2 \sigma$

4.129 Subroutine SIMQ

A(I)	Matrix of coefficients for the system of equations to be solved, stored columnwise with no gaps
B(I)	Constant terms for the system of equations to be solved; these data are replaced by the solution values
BIGA	Pivotal (largest) element in a column
I	Row index
IA	Index of matrix A in back solution
IB	Index of recomputed array B in back solution, containing solution values
IC	Index of array B in back solution
IJ	Index in search for pivotal element in column
IMAX	Index of pivotal element in column
IQS	Index in elimination of variables
IT	Index used for several purposes
IX	DØ index in elimination of variables
IXJ	Index in elimination of variables
IXJX	Index in elimination of variables
I1,I2	Indices in interchange of rows
J	Column index
JJ	Index of the first element in a column of the reduced matrix

JJX	Index in elimination of variables
JX	DØ index in elimination of variables
JY	J + 1
K	DØ index in interchange of rows
KS	Error indicator; set to 1 if matrix of coefficients is singular
N	Number of equations to be solved
NY	N - 1
SAVE	Temporary storage location used in interchanging rows and eliminating variables
TØL	Criterion for value of pivotal element in test for singular matrix

4.130 Subroutine STUNTS

ASTRSK	Hollerith word filled with asterisks
DELTT(I)	Array of preset temperature intervals for table of species thermal properties, ^{o}K
HOO	Heat of formation of a species in kcal/mole
I	DØ index; also value of IQ(LQ); also value of ISEQ(L)
ICARD	Counter of punched cards containing cross section data
II	Index of preset temperature intervals for thermal property table
IIQJ	Index of the first species of a pair of species whose cross section is computed in the Ith step of the transport property calculation

IT	Counter for lines in the thermal property table with the same value of the temperature increment
IZ	Number of lines in thermal property table
J	DØ index;also JQ(LQ)
JJQJ	Index of the second species of a pair of species whose cross section is computed in the Ith step of the transport property calculation
K	DØ index
KK	KQ(L)
L	Index of steps in the cross section calculation
LIM	1 for species with no thermo fit; 2 for species with a thermo fit
LIM1	1 for species with physical model data; 2 for species without such data
LQ	DØ index
LQ1	Index of the first species pair (in the IQ,JQ arrays) to which the Lth step of the edited cross section computation is applied
LQ2	Index of the last species pair (in the IQ,JQ arrays) to which the Lth step of the edited cross section computation is applied
M	DØ index
MV1	Index of the first parameter (in the V array) used in the Lth step of the edited cross section computation
MV2	Index of the last parameter (in the V array) used in the Lth step of the edited cross section computation

NDT(I)	Number of lines in the thermal property table for which the Ith preset temperature increment is used
NNQI	Number of species pairs to which the cross section calculated in the Ith step of the calculation is applied
TPRØP(I,K,M)	Computed thermal properties for the Ith line of the table; M = 1 contains the results of the physical model, M = 2 the thermo-fit results (if any) K = 1 Chemical potential K = 2 Enthalpy K = 3 Specific heat K = 4 Entropy
TT(I)	Temperature value for the Ith line of the thermal property table, ^{o}K
TV	Temperature value, ^{o}K

4.131 Subroutine THERM

CCPTF	Nondimensional specific heat based on the thermo fit
CLGT	Natural logarithm of the nondimensional temperature, $\ell n(T/T_0)$
E1	Logical variable, .TRUE. if the routine was entered through the entry point THERM1, otherwise .FALSE.
J	Species index
K	Index of vibrational states in a linear triatomic species
M	Index of the electronic states in a species

MIX	Logical variable, set to .TRUE. if the species thermal properties are to be calculated by "mixing" results from the physical model and the thermo fit
N	Number of electronic states in a species
SHTF	Nondimensional enthalpy based on the thermo fit
STF	Nondimensional entropy based on the thermo fit
SZ1	Temporary storage for a value required in the entropy calculation for a linear triatomic species
S1	See eq. I(56b)
S2	See eq. I(56c)
S3	See eq. I(56d)
WTHØ	Weight of the physical model results in the "mixed" calculation of species properties
WTTF	Weight of the thermo fit results in the "mixed" calculation of species properties
WW	$(S2)^2$
XMTF	Chemical potential based on the thermo fit results
XX	Dummy variable used in calculations of electronic state contributions to species properties
ZT	Difference between the temperature and the switchover temperature CTMXXI from the physical model to the thermo fit, ^{o}K
Z1-Z9	Storage for intermediate values in calculations of the vibrational contributions to species properties

4.132 Subroutine THRØAT

AG	Geometric area ratio
I	DØ index
ØNE	1.0
XX	Dummy argument for call to AGSØLN

4.133 Subroutine TRANSP

C(I,J)	Molecular weight dependent factor used in transport calculations (see eqs. 171)
CMSAVE	Mean molecular weight W for the mixture
CPTØT	Nondimensional frozen specific heat at constant pressure for the mixture, W c_{pf}/R_0
CSAVE(I)	Array for saving previously computed values of the nondimensional specific heats CCPJ(I)
CTSAVE	Variable for saving previous value of the temperature ratio CT
I	Species index
J	Species index
J1	J - 1
P	Gas pressure in atmospheres
S	$\sqrt{\text{TTAB}}$
SSIG	Factor used in computation of electrical conductivity, $(e/k)^2/\sqrt{T}$
TTAB	Temperature in ^{o}K for which transport calculations are to be made

ZK(L)	See eq. (173)
ZKINT	See eq. (173)
ZM1(I)	Factor proportional to the molecular weight of species I (see eq. (171))

4.134 Subroutine WEDGE

A	$(\gamma + 1)/2$, where γ = specific heat ratio
AE	Free-molecule accommodation coefficient for heat transfer
ALMIN	Minimum angle of attack for calculation of free-molecule heat flux (0.01 radian)
ALPHA	Angle of attack in radians
AM	Free-stream Mach number M
AMD	Double-precision Mach number for call to SHØCK routine
AM2	M^2
AND	Input angle of attack in degrees or ANMIN, whichever is larger
ANMIN	Minimum angle of attack for oblique shock calculation (10^{-4} degree)
APR(IA)	Static pressure ratio, p_2/p_1, based on the oblique shock calculation for the IAth angle
AST(J)	Hollerith array for marking heat fluxes which are at the free-molecule limit with an asterisk
ASTQ(IT)	Hollerith blank or asterisk for ITth column of output table
A2	A^2

A4	A^4
A5	A^5
A7	A^7
C	$[\mu(T^*)/\ \mu(T_1)] \cdot [T_1/T^*]$; see Sec. 8.2.2, Vol. I
CAPGAM	Angle-of-attack parameter Γ (eq. I(481c) or I(501))
CH	Heat transfer coefficients, C_H
CHD	Coefficient $0.332\ A\ \epsilon\ \tau\ N^2 t$ for calculating C_H from δ^*
CHDFAC	Coefficient $0.332\ A\ \epsilon$
CPGM	Single-precision value equal to Γ
DELSTW(IT)	Boundary layer displacement thickness at the ITth point along the wedge
DFAC	Coefficient $\gamma\ k^3/8\ A^5\ \epsilon$
DFAK	Quantity DFAC $\cdot\ t/(\tau N)^4$
EN	Parameter $N = M\sqrt{C/Re_t}$
EPS	Quantity $\epsilon = (\gamma-1)/(\gamma+1)$
EPS2	ϵ^2
ERFV	Quantity 1 + erf(S sin α) used in calculating free-molecule limit to heat flux
EXPV	Quantity $e^{-(S \sin\alpha)^2}$ in free-molecule heat flux calculations

FINAL	Logical variable, set to .TRUE. when calculations have been done for all of the specified distances from the leading edge for given angle of attack and nose radius
FL(K)	Array used by subroutine SHØCK to communicate its results to WEDGE. See glossary of symbols in SHØCK for definitions of the elements of FL
GAMMA	Specific-heat ratio, γ
GAMD	Double-precision value of GAMMA, for call to SHØCK routine
GFAC	Factor used in calculating Γ ; equal to $k/2A^2\epsilon M$ for the modified Cheng-Kemp theory, and to $\gamma k/2 A^2 \epsilon$ for the unmodified theory
GFAK	GFAC/$(\tau N)^2$
GFC	Coefficient $k/2A^2\epsilon$ in GFAC
GMW	Mean molecular weight of gas, g/mole
G2	γ^2
G3	γ^3
HRATIØ	Ratio h_w/h_0 of enthalpy at the wall to the free-stream stagnation enthalpy
HW	Gas enthalpy at the wall temperature, cal/g
H0	Free-stream stagnation enthalpy, cal/g
IA	Index of angles of attack
IAMAX	Index of angle of attack at which classical oblique shock solution gives a detached shock

IERR	Error indicator for classical oblique shock solution, set in SHØCK routine: 0 Detached shock 1 Attached shock - normal case 2 Convergence failure
IFM	Control variable; reset from 0 to 1 if any heat flux value is reduced to the free-molecule limit
IR	Index of leading edge radii
IS9	Absolute value of ISW9B
IT	Index of columns in the output table
ITAB	Index of blocks in the output table
ITH	Index equal to 1 for the modified Cheng-Kemp theory, equal to 2 for the unmodified theory
ITHS	Value of ITH before resetting because of IERR = 0
J	Implied DØ index in WRITE statements
NW1	Index of points in the uniform sequence of distances from the leading edge
NW2	Index of specified distances from the leading edge
ØMEGA	Parameter Ω , eq. I(502)
PFAC	Quantity $4 A^4 \epsilon^2 M^2/ \gamma k^2$
PFAK	Coefficient PFAC $\cdot$ $(\tau N)^4$ in formula for pressure on wedge
PI	Value 3.1415927
PINF	Free-stream pressure, atm

PRATIØ	Ratio of pressure on wedge to free-stream pressure, p_w/p_1
PW:(IT)	Pressure on the wedge at the ITth distance from the leading edge, atm
QFM(IA)	Free-molecule limit to the heat flux for the IAth angle of attack, Btu/ft^2-sec
QFMC1	Coefficient 0.0886 $p_1 a_e \sqrt{RT_1/2\pi W}$ in calculation of free-molecule heat flux, eq. I(512)
QFMC2	Coefficient $S^2 + [\gamma - \frac{1}{2}(\gamma+1) T_w/T_1]/(\gamma-1)$ in free-molecule heat flux calculation, eq. I(512)
QWW(IT)	Heat flux to the wedge at the ITth distance from the leading edge, Btu/ft^2-sec
REPF	Reynolds number per foot
RGC	Gas constant (8.314 x 10^7 erg/mole $^\circ$K)
RHØINF	Free-stream density, ρ_1
RNAME	Hollerith routine name for call to subroutine DUMP
RU2	Dynamic pressure $\rho_1 u_1^2$
S	Free-molecule flow parameter $M\sqrt{\gamma/2}$
SQT	Quantity $\sqrt{zz' + \Omega\zeta}$ in modified Cheng-Kemp calculation; $\sqrt{zz'}$ in unmodified calculation
SQTPI	$\sqrt{\pi}$
SSA	$S \sin\alpha$
SSASQ	$(S \sin\alpha)^2$

TAU	Parameter $\tau = 0.664 + 1.72\ h_w/h_0$
TAUN	τN
TAUN2	$(\tau N)^2$
TAUN3	$(\tau N)^3$
TAUN4	$(\tau N)^4$
TAUN6	$(\tau N)^6$
TINF	Free-stream temperature, T_1
TN	Diameter of leading edge, inch
TRATIØ	Ratio of wedge surface temperature to free-stream temperature
TREF	Reference temperature, $(1 + 3\ h_w/h_0)\ T_0/6$, where T_0 is the free-stream stagnation temperature behind a frozen normal shock
UNIF	Free-stream velocity, u_1
UN(J)	Hollerith array for labelling the output as being based on the modified or unmodified Cheng-Kemp theory
VMUINF	Free-stream viscosity, lb/ft-sec
WK2	Square of leading-edge drag coefficient, k^2
WK3	k^3
WK4	k^4
WXV1	Next distance from the leading edge in the specified uniform sequence, inch

WXV2	Next separately specified distance from the leading edge, inch
XC	Coordinate x parallel to the free-stream flow for a point on the surface of the wedge at a distance XW(IT) from the leading edge; XC = XW(IT)·cos α, inch
XM	Distance from the leading edge up to which merging effects are important (inch); see eq. I(510a)
XS	Distance from the leading edge beyond which the strong-interaction approximation breaks down (inch); see eq. I(510b)
XW(IT)	The ITth specified distance from the leading edge, inch
YS(IT)	Shock ordinate at the ITth distance from the leading edge, inch
YSFAC	Coefficient $\gamma^2 k^3/8\ A^4 \epsilon$ in calculation of shock ordinate
YSFAK	Coefficient YSFAC·t/$(\tau N)^4$ in calculation of shock ordinate
Z	Nondimensional shock ordinate z; eq. I(481a)
ZERØ	0.0
ZETA	Nondimensional coordinate ζ parallel to free-stream flow; see eq. I(481b)
ZFAC	Coefficient $16\ A^7 \epsilon\ /\ \gamma^3 k^4$
ZFAK	Coefficient ZFAC·$(\tau N)^6/t$
ZTA(IT)	ζ at the ITth distance from the leading edge
ZZP	zz'
ZZPP	(zz')'

ZZPS	z..' as a single-precision value

4.135 Subroutine WESØLN

B0	Coefficient $\sqrt[3]{6}$ in initial approximation to λ
B1	Coefficient $\sqrt{3}$ in initial approximation to λ
CALLED	Logical variable used to bypass resetting of the "C" coefficients after the first call to WESØLN
CAPGAM	Angle of attack parameter, Γ
CGSQ	Γ^2
C1	1. The "C" symbols are coefficients in the Cheng-Kemp analytical solution of the Cheng equation for $\Gamma = 0$
C2	2.
C3	1/3
C4	4.
C9	9.
C32	1.5
C103	10/3
C229	22/9
C463	46/3
C769	76/9
DLDZT	$d\lambda/d\zeta$ for Newton-Raphson iteration

N	Iteration counter
PJ	$\zeta^{1/3}$
P2	$\zeta^{1/2}$
P12	$\zeta^{2/3}$
Q1	$1 + \sqrt{\lambda}$
Q2	$(1 + \sqrt{\lambda})^2$
Q3	$(1 + \sqrt{\lambda})^3$
Q4	$(1 + \sqrt{\lambda})^4$
Q5	$\ln (1 + \sqrt{\lambda})$
SQTL	$\sqrt{\lambda}$
XL	λ
Z	Nondimensional shock ordinate, z
ZETA	Nondimensional coordinate ζ parallel to the direction of free-stream flow
ZT	ζ calculated using trial value of λ
ZZP	zz'
ZZPP	(zz')'

4.136 Subroutine XSECT

Because a number of symbols are used for different purposes in different sections of subroutine XSECT, the symbols used in each section of the subroutine are defined separately below.

Statements to Statement 40

I(J)	Revised index for species J in master species list (common variable)
J	Index of species in master species list
JJ	Revised index for species not included in transport calculations
K	Count of species
L	Revised index for species included in transport calculations

Statements 40 to 110

J	Species index
K	Index of steps in the cross section calculation in the order in which they are carried out
KIJ	Index in packed lists of species pairs to which edited cross section data are applied
KM	Index of parameters for a single step of the cross section calculation
KV	Index in packed list of parameter values for all steps of the cross section calculation
L	Species index; step of cross section computations
LL	First index of species pair
LQ	NQ(L)
M	Index of the Kth step in the cross section calculation, equal to ISEQ(K)

MMV	Total number of parameters in V array
NK	Index of the option to be used in the Kth step; also, number of parameters used in step

Statements 110 through 190

I(J)	Revised index for species J in master list (common variable)
II	First index of unused species pair
III	II-1
J	Second index of species pair referenced in V array; species index in master species list
JJ	Index of species added to cross section computation
K	Index in master species list of species added to cross section computations
L	Index of step in cross section computations
LL	First index of species pair referenced in V array
LQ	Location of species pair in IQ, JQ array
LQ1	Location in IQ, JQ array of first species pair for step
LQ2	Location in IQ, JQ array of last species pair for step
MV	Dummy variable (not used)

Statement 190 to Card XSE 213

IV	First index of species pair computed with option KQ = 10
J	Second index of species pair to be moved
JV	Second index of species pair computed with option KQ = 10
K	Count of array elements; dummy species index
L	Step in cross section computations; dummy species index
LL	First index of species pair to be moved
LQ	Location of species pair IV, JV in IQ, JQ array
LQ1	Location in IQ, JQ array of last species pair for step L-1 of computations
LQ2	Location in IQ, JQ array of last species pair for step L of computations
LV	Index of array elements to be moved
MM	Location in V array of last parameter for step L-1 of computations
MMV	Total number of parameters in V array
MV	Index for parameters in V array
NSV	Total number of steps in edited cross section computations

Card XSE 213 to Statement 300

II	First index of unused storage location for species pair
IV	First index of a species pair
J	Second index of species pair to be moved
JJ	Second index of unused storage location for species pair
JV	Second index of a species pair
L	Step of cross section computations
LL	First index of species pair to be moved
LQ	Index of species pair in IQ, JQ array
LQ2	Total number of species pairs in IQ, JQ array
MV	Dummy argument, value not required

Statement 380 to end of subroutine

I(K)	Species for which cross section data are to be supplied (common variable)
IE	For IE = 1, cross section data are not supplied for electrons

II	Absolute value of IZ for a species
IZ(L)	Charge of species L in units of electron charge (Note that IZ is negative for positively charged species)
J	Index indicating charge for species for which cross sections are to be supplied (see Table III)
JJ	Index indicating whether cross sections are to be supplied for electrons (see Table III)
KK	Index indicating charge of species for which cross sections are to be supplied (see Table III)
L	Step in cross section computations; species index
LL	Computational step at which additional cross section data are supplied for neutral-neutral interactions
NEWMV	Total number of parameters in V array
NN	Number of species for which cross section data are to be supplied

VCOUL(M,KK,J,JJ) Parameters V(M) for effective Coulomb cross sections between two species with the electronic charges KK and J, from eqs. I(100a). The parameters for JJ = 1 are for ion-ion collisions, and for JJ = 2 for ion-electron collisions

5. REFERENCES TO COMMON VARIABLES

This section presents a computer-generated analysis of references to common variables and common-equivalent variables in the NATA code. The variables are listed in alphanumeric order. For each variable, the name of the common block containing it is given. The block name / / refers to unlabelled common. An asterisk following the block name indicates that the variable does not appear in common, but is equivalenced to a common variable in the designated block in one or more routines. The routines in which each variable is referenced are indicated by numbers, based on the list of routine names given on the first page. Routine number 2 is the block data routine containing species, reaction, geometry, and other data; number 3 is the transport cross section block data routine. The routines EXP and RESET are missing from the list because the analysis was carried out for the UNIVAC version of NATA, which does not contain these routines. Neither EXP nor RESET contains any common blocks.

Lines containing a blank variable name are continuations of the preceding lines, in cases in which the number of referencing routines is too large to be printed on a single line.

Appearance of a variable in a common or equivalence statement is not considered to constitute a reference. A routine is listed as referencing a given variable if the variable appears in an arithmetic or DATA statement, a READ or WRITE statement or a NAMELIST statement, or is used as an index or a subroutine or function argument.

In the case of a common-equivalent variable, the program used to prepare the list checks for the presence of the associated common block in the routine, but does not verify the presence of the equivalence statement in the routine. For this reason, some of the listed references to common-equivalent variables are spurious. For example, in subroutine BLAYER, the variable X is equivalenced to the

variable CX in unlabelled common. Subroutines GEØMAR, ØUT1, and TRANSP are also listed as referencing X. All three of these references are spurious because X is not equivalenced to a common variable in these three routines.

In some cases, different names are used for the same common variable in different routines. All such names are included in the list below, each with its own list of referencing routines.

INDICES OF ROUTINES

1	MAIN	36	NRMAX
2	BLOCKD	37	OUT
3	BLOCKD	38	OUT1
4	AESOLN	39	PERT
5	AGSOLN	40	PIOMEG
6	AXFIT	41	PROP
7	AXSECT	42	PRTA
8	BLAYER	43	PUTQIN
9	BLCALL	44	QCOUL
10	BXSECT	45	QFX
11	COMM	46	QEXP
12	CXSECT	47	QINTRP
13	DERIVS	48	QLJ
14	DSMSOL	49	QMIX
15	DUMP	50	QREPP
16	DUMPEX	51	QSAME
17	ELCOND	52	QTAB
18	ELTIME	53	Q11
19	EPART	54	Q12
20	FQCALC	55	Q13
21	EXACT	56	Q14
22	FINDX	57	RADIUS
23	FROZFQ	58	READ
24	GEOM	59	RESTMP
25	GEOMAR	60	RNKT
26	INGAS	61	SHOCK
27	INIT	62	SIMQ
28	KANDMU	63	STUNTS
29	KINT	64	THERM
30	LIST	65	THROAT
31	MATINV	66	TRANSP
32	MODEL	67	WEDGE
33	NEWRAP	68	WFSOLN
34	NEXTMP	69	XSECT
35	NONEQ		

VARIABLE	COMMON	REFERENCING ROUTINES
A	/ /	1, 16, 24,
AA	/ /	20, 33, 35, 39,
AAA	/ /*	21,
AAMS	/READAT/	2, 58,
ACOM	/ /	58,
AE	/ /*	8,
AENTS	/ /	1, 6, 16, 35, 42,
AFNX	/ /	11, 16, 23, 24, 35, 38, 60, 65,
AIN	/INGNE /	2, 16, 26, 35,
ALGJ	/ /*	11,
AM	/OUTPUT/*	67,
AMACH	/ /	16, 21, 23, 32, 39, 42, 65,
AMP	/BLNE /	8, 16,
AMPP	/BLNF /	8, 16,
ANGLE	/RDWEDG/	2, 30, 58, 67,
AR	/ /	16, 23, 35,
ARBA	/ /	16, 27, 35,
ARBB	/ /	16, 27, 35,
ASTAR	/TRANS4/	50, 58,
ATPI	/AREA /	16, 25, 58,
AXIMOD	/READAT/	2, 58,
AXISYM	/RDWEDG/	23, 30, 32, 35, 58,
AZ	/ /	
B	/TRANS3/	28, 29, 66,
BE	/ /	16, 35, 39,
BET	/ /	16, 20, 21, 26, 33,
BETA	/ /	11, 21, 35, 39,
BLBK	/ /	16, 35,
BLINT	/BL /	8, 16, 35,
BLNEAR	/BLNE /*	35,
BLVAR	/BLOUT /*	23,
BPAR	/TRANSO/	11, 16, 58,
BR	/TRANS3/	28,
BSTAR	/TRANS4/	50, 58,
BTA	/ /*	
BZERO	/ /	16, 20, 33, 41, 58,
BZEROI	/READAT/	2, 58,
C	/ /	1, 16, 24, 66,
CAI	/ /	11, 16, 30, 35, 58,
CAPO	/ /*	26, 30, 58,
CAPX	/ /	16, 26, 33, 36, 42,
CAPXTH	/ /	16, 36,
CARB	/ /	16, 35,
CAS	/SS /	8, 23, 42,
CATFAC	/MODVAR/	2, 16, 30, 32, 58,
CCHI	/RENF /	2, 30, 35, 58,
CCI	/ /*	26,
CCPJ	/ /	1, 8, 11, 13, 16, 21, 38, 39, 63, 64, 66,
CDIJ	/ /	20, 21, 26, 33, 35,
CEACT	/ /	11, 16, 30, 58,

VARIABLE	COMMON	REFERENCING ROUTINES
CF	/CONVRT/	1, 16, 32, 38,
CGI	/ /	11, 16, 20, 26, 33, 66,
CGMU	/ /	16, 26, 33,
CGMW	/COLDSP/	1, 8, 58,
CH	/ /	1, 11, 16, 23, 26, 32, 33, 38, 41,
CHA	/ /	1, 8, 11, 16, 23, 26, 32, 33, 35, 38, 41, 59, 60,
CHANAM	/RDOUT /	38, 58,
CHB	/TNERK /	35, 60,
CHI	/ /	11, 16, 38, 42,
CHII	/ /	16, 20, 33,
CLNPI	/ /	11, 16,
CLNT	/ /	13, 16, 64,
CLNTD	/TNDC /	11, 13, 16,
CLN1MC	/ /	11, 16,
CM	/ /	1, 8, 9, 11, 16, 23, 26, 33, 38, 39,
CMA	/ /	1, 8, 11, 16, 21, 23, 26, 32, 33, 35, 38, 39, 41,
	/ /	59, 63,
CMW	/ /	16, 26, 30, 58,
COUPLD	/BINE /	8, 16, 35,
CP	/CHAN /*	58,
CPWALL	/DL /	1, 8, 16, 32, 38, 67,
CP1	/CHAN /	2,
CP2	/CHAN /	2,
CP3	/CHAN /	
CP4	/CHAN /	
CP5	/CHAN /	
CRA	/ /	1, 11, 16, 26, 27, 32, 63,
CRP	/ /	1, 8, 11, 16, 21, 27, 32, 35, 38, 59, 63,
CRRB	/ /	16, 23, 26, 33,
CRS	/ /	16, 35,
CSTA	/ /	11, 16, 35,
CT	/ /	1, 6, 8, 9, 11, 13, 16, 20, 21, 23, 27, 32, 33,
	/ /	35, 36, 38, 39, 41, 42, 60, 63, 64, 66,
CTAP	/ /	1, 8, 11, 16, 20, 21, 23, 26, 27, 32, 35, 38, 39,
	/ /	58, 59, 63, 64, 66,
CTAPI	/READAT/	58,
CTB	/ /	16, 35, 60,
CTC	/ /	16,
CTD	/TNDC /	11, 13, 16,
CTE	/TNONEO/	11, 13, 16, 21, 27, 35, 38, 42, 60,
CTE3	/TNERK /	16, 35, 60,
CTMAX	/ /	1, 6, 16, 23, 35, 36,
CTMXX	/ /	16, 27, 30, 58, 63, 64,
CTMXXI	/MASSFL/	2, 27, 58,
CTP	/ /	16, 64,
CTPL	/ /	16, 27, 64,
CTT	/ /	16, 35, 42,
CX	/ /	1, 6, 11, 13, 16, 23, 24, 35, 38, 60, 65,
CXB	/ /	8, 16, 35, 60,
CXMAX	/ /	16, 23, 30, 35, 58,
CXMAXI	/READAT/	2, 58,
CZ	/ /	
DATAPE	/TAPOUT/	1, 2, 16, 32, 38, 58,

VARIABLE	COMMON	REFERENCING ROUTINE
DATEST	/ /	16. 27. 35.
DBTEST	/ /	16. 27.
DCA	/ /*	35.
DCHA	/TNERK /	8. 21. 60.
DCHLL	/RENE /	2. 35. 58.
DCHRAT	/RENE /	2. 35. 58.
DCI	/ /*	35.
DDBLP	/BLNE /	8. 16.
DDELBL	/NEQ /	4. 8. 16. 35. 42.
DELBL	/BL /	1. 4. 6. 8. 13. 16. 23. 35. 38. 42. 65.
DELBLP	/BLNE /	8. 16.
DELTAX	/ /	16. 35. 58. 60.
DELTXI	/READAT/	2. 58.
DELTI	/ /	1. 16. 23. 35. 36. 58.
DELTII	/READAT/	2. 58.
DELT2	/ /	16.
DGJ	/ /*	21. 38. 39. 60.
DH	/TRANS3/	29. 66.
DIAM	/BL /	1. 16. 38. 58.
DLGDX	/ /*	38.
DLOGA	/ /	11. 13. 16. 21. 24. 35. 42. 60. 65.
DLOGR	/ /	11. 16. 21. 24. 35. 42. 60.
DOMAX	/TNERK /	35. 60.
DT	/ /	8. 16. 21. 42. 60.
DTE	/TNONEQ/	8. 16. 21. 60.
DUM	/TNCE /*	58.
DWX	/RDWEDG/	2. 30. 58. 67.
DXMODP	/MODPAR/	
DXMP	/MODPAR/	
DXP	/SWITCH/*	8.
DXPB	/SWITCH/	16. 23. 35.
EEP	/EELEM /	
EEPRP	/EELEM /*	58.
EEP1	/EELEM /	58.
EEP10	/EELEM /	58.
EEP2	/EELEM /	58.
EEP3	/EELEM /	58.
EEP4	/EELEM /	58.
EEP5	/EELEM /	58.
EEP6	/EELEM /	58.
EEP7	/EELEM /	58.
EEP8	/EELEM /	58.
EEP9	/EELEM /	58.
ELJ	/ /	27. 30. 58. 64.
ELMENT	/ /	30. 58.
ENT	/ /	16. 23. 41.
EPAR	/TNONEQ/	16. 19. 30. 58.
EPRP	/ELEM /*	58.
EPSF	/STAG /	16. 27. 32. 35.
EPSLON	/STAG /	16. 23. 32. 35.

VARIABLE	COMMON	REFERENCING ROUTINES
EP1	/ELEM /	2.
EP10	/ELEM /	
EP2	/ELEM /	
EP3	/ELEM /	2.
EP4	/ELEM /	2.
EP5	/ELEM /	2.
EP6	/ELEM /	2.
EP7	/ELEM /	2.
EP8	/ELEM /	
EP9	/ELEM /	
ERR	/ERROR /	1. 4. 5. 8. 11. 13. 15. 20. 21. 22. 23. 24. 26.
	/ERROR /	32. 33. 34. 35. 36. 38. 39. 42. 59. 60. 65. 66. 69.
ETAI	/ /	11. 16. 30. 35. 58.
ETAJ	/ /	16. 27. 30. 58. 63. 64.
FACMP	/NEWMP /	16. 34. 58.
FACNAM	/RDOUT /	38. 58.
FAILED	/CERROR/	11. 13. 16. 35. 60.
FLEWIS	/TRPROP/	1. 16. 32. 66.
FLEWR	/RESPRP/	1.
FLOW	/RDOUT /	1. 38. 58.
FLUX	/ /	1. 16. 23. 33. 35. 36. 41.
FSTAG	/MODPAR/	2. 16. 30. 32. 58.
FVOUT	/OUTPUT/	38. 67.
GAMIN	/RENE /	2. 35. 58.
GAMMA	/OUTPUT/*	67.
GELJ	/ /	30. 58. 64.
GJ	/*	1. 8. 9. 11. 21. 23. 32. 33. 35. 36. 38. 39. 60.
GJA	/ /	1. 16. 23. 26. 35. 36. 41. 63.
GJB	/ /	16. 35. 60.
GJMF	/OUTPUT/	
GMW	/OUTPUT/*	67.
GPRP	/MIXT /*	58.
GP1	/MIXT /	2.
GP2	/MIXT /	2.
GP3	/MIXT /	2.
GP4	/MIXT /	2.
GP5	/MIXT /	2.
GP6	/MIXT /	2.
GSO	/*	35.
GTEST	/RENE /	2. 35. 58.
H	/*	8. 38. 59.
HDFLX	/ /	16. 35. 60.
HJ	/ /	30. 32. 35. 38. 58. 63.
HR	/BLOUT /	2. 8. 16. 38.
HS	/ROMAIN/	32. 38. 58. 59.
HSTAG	/READAT/	58.
HTEST	/RENE /	2. 35. 58.
I	/TRANS2/	7. 1 8. 63. 69.
IAMBIP	/RDMCD /	30. 66.
IATOM	/LN /	69.
IC	/ /	16. 32. 33. 38. 58.

VARIABLE	COMMON	REFERENCING ROUTINES
ICASE	/READAT/	2, 58,
ICH	/TNONEO/	11, 16, 58,
ICHAN	/READAT/	2, 58,
ICHANO	/RDLIST/	2, 58,
ICI	/READAT/	58,
ID1	/TRANS5/	66, 69,
ID2	/TRANS5/	66, 69,
IE	/READAT/	58,
IEEP	/READAT/	58,
IELEC	/TRANS5/	17, 69,
IFAIL	/TNERK /	35, 60,
IFLOW	/TAPOUT/	
IGAS	/RDLIST/	2, 58,
IGASO	/RDLIST/	2, 58,
IGJ	/ /	16, 27, 30, 58, 63, 64,
IGM	/ /	16, 27, 30, 58, 64,
IIIII	/TRANS2/	
II0	/TRANI /	63, 69,
II1	/TRANI /	3, 69,
II10	/TRANI /	3,
II100	/TRANI /	
II11	/TRANI /	3,
II12	/TRANI /	3,
II13	/TRANI /	3,
II14	/TRANI /	3,
II15	/TRANI /	3,
II16	/TRANI /	3,
II17	/TRANI /	3,
II18	/TRANI /	3,
II19	/TRANI /	3,
II2	/TRANI /	3,
II20	/TRANI /	3,
II21	/TRANI /	3,
II22	/TRANI /	3,
II23	/TRANI /	3,
II24	/TRANI /	3,
II25	/TRANI /	3,
II26	/TRANI /	3,
II27	/TRANI /	3,
II28	/TRANI /	3,
II29	/TRANI /	3,
II3	/TRANI /	3,
II30	/TRANI /	3,
II31	/TRANI /	3,
II32	/TRANI /	3,
II33	/TRANI /	3,
II34	/TRANI /	3,
II35	/TRANI /	3,
II36	/TRANI /	3,
II37	/TRANI /	3,

VARIABLE	COMMON	REFERENCING ROUTINE
II38	/TRANI /	3.
II39	/TRANI /	3.
II4	/TRANI /	3.
II40	/TRANI /	3.
II41	/TRANI /	3.
II42	/TRANI /	3.
II43	/TRANI /	3.
II44	/TRANI /	3.
II45	/TRANI /	3.
II46	/TRANI /	3.
II47	/TRANI /	3.
II48	/TRANI /	3.
II49	/TRANI /	3.
II5	/TRANI /	3.
II50	/TRANI /	3.
II51	/TRANI /	3.
II52	/TRANI /	3.
II53	/TRANI /	3.
II54	/TRANI /	3.
II55	/TRANI /	3.
II56	/TRANI /	3.
II57	/TRANI /	3.
II58	/TRANI /	3.
II59	/TRANI /	3.
II6	/TRANI /	3.
II60	/TRANI /	3.
II61	/TRANI /	3.
II62	/TRANI /	3.
II63	/TRANI /	3.
II64	/TRANI /	
II65	/TRANI /	
II66	/TRANI /	
II67	/TRANI /	
II68	/TRANI /	
II69	/TRANI /	
II7	/TRANI /	3.
II70	/TRANI /	
II71	/TRANI /	
II72	/TRANI /	
II73	/TRANI /	
II74	/TRANI /	
II75	/TRANI /	
II76	/TRANI /	
II77	/TRANI /	
II78	/TRANI /	
II79	/TRANI /	
II8	/TRANI /	3.
II80	/TRANI /	
II81	/TRANI /	
II82	/TRANI /	

TABLE	COMMON	REFERENCING ROUTINES
II83	/TRANI /	
II84	/TRANI /	
II85	/TRANI /	
II86	/TRANI /	
II87	/TRANI /	
II88	/TRANI /	
II89	/TRANI /	
II9	/TRANI /	3.
II90	/TRANI /	
II91	/TRANI /	
II92	/TRANI /	
II93	/TRANI /	
II94	/TRANI /	
II95	/TRANI /	
II96	/TRANI /	
II97	/TRANI /	
II98	/TRANI /	
II99	/TRANI /	
IJCS	/OLDSP/	1. 8. 32. 58.
IM	/ /	1. 16. 27. 33.
IMOL	/LN /	69.
IMP	/TAPOUT/	32. 38.
INEQ	/ /	8. 9. 11. 16. 21. 23. 24. 27. 35. 38. 42.
INEQV	/ /	16. 27. 30. 58. 64.
INEQVI	/READAT/	2. 58.
INT	/READAT/	2. 58.
IP	/ /	16.
IPA	/TNONEQ/	11. 16. 30. 58.
IPOINT	/BL /	8. 16. 23. 35.
IQ	/TRANS7/	7. 43. 49. 55. 56. 63. 69.
IR	/READAT/	58.
IRUN	/ /	1. 16. 58.
IS	/MASSFL/	27. 58. 69.
ISATOM	/LN /	58.
ISC	/ /	16. 20. 21. 26. 30. 32. 33. 35. 36. 39. 58.
ISCI	/READAT/	58.
ISCP1	/ /	16. 21. 35. 39.
ISEQ	/TRANS2/	3. 58. 63. 69.
ISHAPE	/AREA /	16. 25. 58.
ISMC	/ /	1. 16. 20. 26. 27. 35.
ISMCNR	/ /	1. 16. 27. 33.
ISMO	/BLNE /	8. 16. 35.
ISMOL	/LN /	58.
ISR	/ /	11. 16. 21. 30. 35. 38. 39. 58.
ISRI	/READAT/	58.
ISS	/ /	1. 8. 9. 11. 13. 16. 20. 21. 23. 26. 27. 30. 32.
	/ /	33. 35. 36. 38. 39. 41. 58. 60. 63. 64. 66. 69.
ISSI	/READAT/	58.
ISSNR	/ /	1. 16. 27. 33.
ISSP1	/ /	16. 21. 35. 39. 60.
ISSP2	/ /	16. 21. 35. 39. 60.

VARIABLE	COMMON	REFERENCING ROUTINES
ISSP3	/ /	16. 21. 35. 39. 60.
ISSP4	/ /	16. 21. 35. 60.
ISUPR	/ /*	35.
ISW1A	/ /	1. 16. 58.
ISW1B	/ /	1. 16. 58. 63. 66.
ISW2A	/ /	1. 16. 58.
ISW2B	/ /	1. 16. 38. 58. 59.
ISW3A	/ /	1. 16. 58.
ISW3B	/ /	1. 6. 9. 13. 16. 23. 24. 30. 35. 38. 42. 58. 65.
ISW4A	/ /	1. 16. 58
ISW4B	/ /	1. 8. 16. 58.
ISW5A	/ /	1. 16. 58. 59.
ISW5B	/ /	1. 11. 16. 35. 42. 58. 60.
ISW6A	/ /	1. 16. 58.
ISW6B	/ /	1. 16. 38. 58.
ISW7B	/READAT/	2. 58.
ISW8B	/RDTR /	3. 43. 47. 58. 66.
ISW9B	/RDWEDG/	2. 58. 67.
ITB	/ /	16. 35.
ITPOUT	/TAPOUT/	1. 2. 16. 32. 38. 58.
ITR	/TNONEQ/	16. 19. 30. 58.
ITYPER	/TAPOUT/	1. 32. 38.
IUPD	/ /	8. 16. 21. 35. 42. 58. 65.
IZERO	/ /	1. 16. 30. 58. 64.
I1	/TRANI /	58.
I10	/TRANI /	58.
I100	/TRANI /	58.
I11	/TRANI /	58.
I12	/TRANI /	58.
I13	/TRANI /	58.
I14	/TRANI /	58.
I15	/TRANI /	58.
I16	/TRANI /	58.
I17	/TRANI /	58.
I18	/TRANI /	58.
I19	/TRANI /	58.
I2	/TRANI /	58.
I20	/TRANI /	58.
I21	/TRANI /	58.
I22	/TRANI /	58.
I23	/TRANI /	58.
I24	/TRANI /	58.
I25	/TRANI /	58.
I26	/TRANI /	58.
I27	/TRANI /	58.
I28	/TRANI /	58.
I29	/TRANI /	58.
I3	/TRANI /	58.
I30	/TRANI /	58.
I31	/TRANI /	58.

VARIABLE	COMMON	REFERENCING ROUTINES
I32	/TRANI /	58.
I33	/TRANI /	58.
I34	/TRANI /	58.
I35	/TRANI /	58.
I36	/TRANI /	58.
I37	/TRANI /	58.
I38	/TRANI /	58.
I39	/TRANI /	58.
I4	/TRANI /	58.
I40	/TRANI /	58.
I41	/TRANI /	58.
I42	/TRANI /	58.
I43	/TRANI /	58.
I44	/TRANI /	58.
I45	/TRANI /	58.
I46	/TRANI /	58.
I47	/TRANI /	58.
I48	/TRANI /	58.
I49	/TRANI /	58.
I5	/TRANI /	58.
I50	/TRANI /	58.
I51	/TRANI /	58.
I52	/TRANI /	58.
I53	/TRANI /	58.
I54	/TRANI /	58.
I55	/TRANI /	58.
I56	/TRANI /	58.
I57	/TRANI /	58.
I58	/TRANI /	58.
I59	/TRANI /	58.
I6	/TRANI /	58.
I60	/TRANI /	58.
I61	/TRANI /	58.
I62	/TRANI /	58.
I63	/TRANI /	58.
I64	/TRANI /	58.
I65	/TRANI /	58.
I66	/TRANI /	58.
I67	/TRANI /	58.
I68	/TRANI /	58.
I69	/TRANI /	58.
I7	/TRANI /	58.
I70	/TRANI /	58.
I71	/TRANI /	58.
I72	/TRANI /	58.
I73	/TRANI /	58.
I74	/TRANI /	58.
I75	/TRANI /	58.
I76	/TRANI /	58.
I77	/TRANI /	58.

VARIABLE	COMMON	REFERENCING ROUTINES
I78	/TRANI /	58.
I79	/TRANI /	58.
I8	/TRANI /	58.
I80	/TRANI /	58.
I81	/TRANI /	58.
I82	/TRANI /	58.
I83	/TRANI /	58.
I84	/TRANI /	58.
I85	/TRANI /	58.
I86	/TRANI /	58.
I87	/TRANI /	58.
I88	/TRANI /	58.
I89	/TRANI /	58.
I9	/TRANI /	58.
I90	/TRANI /	58.
I91	/TRANI /	58.
I92	/TRANI /	58.
I93	/TRANI /	58.
I94	/TRANI /	58.
I95	/TRANI /	58.
I96	/TRANI /	58.
I97	/TRANI /	58.
I98	/TRANI /	58.
I99	/TRANI /	58.
JATOM	/LN /	30. 58.
JCS	/READAT/	58.
JDIM	/BL /	1. 2. 4. 5. 8. 16. 24. 25. 38. 58.
JJK	/ /	1. 16. 27. 33.
JJ0	/TRANJ /	63. 69.
JJ1	/TRANJ /	3.
JJ10	/TRANJ /	3.
JJ100	/TRANJ /	
JJ11	/TRANJ /	3.
JJ12	/TRANJ /	3.
JJ13	/TRANJ /	3.
JJ14	/TRANJ /	3.
JJ15	/TRANJ /	3.
JJ16	/TRANJ /	3.
JJ17	/TRANJ /	3.
JJ18	/TRANJ /	3.
JJ19	/TRANJ /	3.
JJ2	/TRANJ /	3.
JJ20	/TRANJ /	3.
JJ21	/TRANJ /	3.
JJ22	/TRANJ /	3.
JJ23	/TRANJ /	3.
JJ24	/TRANJ /	3.
JJ25	/TRANJ /	3.
JJ26	/TRANJ /	3.
JJ27	/TRANJ /	3.

VARIABLE	COMMON	REFERENCING ROUTINES
JJ28	/TRANJ /	3.
JJ29	/TRANJ /	3.
JJ3	/TRANJ /	3.
JJ30	/TRANJ /	3.
JJ31	/TRANJ /	3.
JJ32	/TRANJ /	3.
JJ33	/TRANJ /	3.
JJ34	/TRANJ /	3.
JJ35	/TRANJ /	3.
JJ36	/TRANJ /	3.
JJ37	/TRANJ /	3.
JJ38	/TRANJ /	3.
JJ39	/TRANJ /	3.
JJ4	/TRANJ /	3.
JJ40	/TRANJ /	3.
JJ41	/TRANJ /	3.
JJ42	/TRANJ /	3.
JJ43	/TRANJ /	3.
JJ44	/TRANJ /	3.
JJ45	/TRANJ /	3.
JJ46	/TRANJ /	3.
JJ47	/TRANJ /	3.
JJ48	/TRANJ /	3.
JJ49	/TRANJ /	3.
JJ5	/TRANJ /	3.
JJ50	/TRANJ /	3.
JJ51	/TRANJ /	3.
JJ52	/TRANJ /	3.
JJ53	/TRANJ /	3.
JJ54	/TRANJ /	3.
JJ55	/TRANJ /	3.
JJ56	/TRANJ /	3.
JJ57	/TRANJ /	3.
JJ58	/TRANJ /	3.
JJ59	/TRANJ /	3.
JJ6	/TRANJ /	3.
JJ60	/TRANJ /	3.
JJ61	/TRANJ /	3.
JJ62	/TRANJ /	3.
JJ63	/TRANJ /	3.
JJ64	/TRANJ /	
JJ65	/TRANJ /	
JJ66	/TRANJ /	
JJ67	/TRANJ /	
JJ68	/TRANJ /	
JJ69	/TRANJ /	
JJ7	/TRANJ /	3.
JJ70	/TRANJ /	
JJ71	/TRANJ /	
JJ72	/TRANJ /	

VARIABLE	COMMON	REFERENCING ROUTINES
JJ73	/TRANJ /	
JJ74	/TRANJ /	
JJ75	/TRANJ /	
JJ76	/TRANJ /	
JJ77	/TRANJ /	
JJ78	/TRANJ /	
JJ79	/TRANJ /	
JJ8	/TRANJ /	3.
JJ80	/TRANJ /	
JJ81	/TRANJ /	
JJ82	/TRANJ /	
JJ83	/TRANJ /	
JJ84	/TRANJ /	
JJ85	/TRANJ /	
JJ86	/TRANJ /	
JJ87	/TRANJ /	
JJ88	/TRANJ /	
JJ89	/TRANJ /	
JJ9	/TRANJ /	3.
JJ90	/TRANJ /	
JJ91	/TRANJ /	
JJ92	/TRANJ /	
JJ93	/TRANJ /	
JJ94	/TRANJ /	
JJ95	/TRANJ /	
JJ96	/TRANJ /	
JJ97	/TRANJ /	
JJ98	/TRANJ /	
JJ99	/TRANJ /	
JMOL	/LN /	30. 58.
JQ	/TRANS7/	7. 43. 49. 55. 56. 63. 69.
J1	/TRANJ /	58.
J10	/TRANJ /	58.
J100	/TRANJ /	58.
J11	/TRANJ /	58.
J12	/TRANJ /	58.
J13	/TRANJ /	58.
J14	/TRANJ /	58.
J15	/TRANJ /	58.
J16	/TRANJ /	58.
J17	/TRANJ /	58.
J18	/TRANJ /	58.
J19	/TRANJ /	58.
J2	/TRANJ /	58.
J20	/TRANJ /	58.
J21	/TRANJ /	58.
J22	/TRANJ /	58.
J23	/TRANJ /	58.
J24	/TRANJ /	58.
J25	/TRANJ /	58.

VARIABLE	COMMON	REFERENCING ROUTINES
J26	/TRANJJ/	SR**
J27	/TRANJJ/	SR**
J28	/TRANJJ/	SR**
J29	/TRANJJ/	SR**
J3	/TRANJJ/	SR**
J30	/TRANJJ/	SR**
J31	/TRANJJ/	SR**
J32	/TRANJJ/	SR**
J33	/TRANJJ/	SR**
J34	/TRANJJ/	SR**
J35	/TRANJJ/	SR**
J36	/TRANJJ/	SR**
J37	/TRANJJ/	SR**
J38	/TRANJJ/	SR**
J39	/TRANJJ/	SR**
J4	/TRANJJ/	SR**
J40	/TRANJJ/	SR**
J41	/TRANJJ/	SR**
J42	/TRANJJ/	SR**
J43	/TRANJJ/	SR**
J44	/TRANJJ/	SR**
J45	/TRANJJ/	SR**
J46	/TRANJJ/	SR**
J47	/TRANJJ/	SR**
J48	/TRANJJ/	SR**
J49	/TRANJJ/	SR**
J5	/TRANJJ/	SR**
J50	/TRANJJ/	SR**
J51	/TRANJJ/	SR**
J52	/TRANJJ/	SR**
J53	/TRANJJ/	SR**
J54	/TRANJJ/	SR**
J55	/TRANJJ/	SR**
J56	/TRANJJ/	SR**
J57	/TRANJJ/	SR**
J58	/TRANJJ/	SR**
J59	/TRANJJ/	SR**
J6	/TRANJJ/	SR**
J60	/TRANJJ/	SR**
J61	/TRANJJ/	SR**
J62	/TRANJJ/	SR**
J63	/TRANJJ/	SR**
J64	/TRANJJ/	SR**
J65	/TRANJJ/	SR**
J66	/TRANJJ/	SR**
J67	/TRANJJ/	SR**
J68	/TRANJJ/	SR**
J69	/TRANJJ/	SR**
J7	/TRANJJ/	SR**
J70	/TRANJJ/	SR**

VARIABLE	COMMON	REFERENCING ROUTINES
J71	/TRANJ /	58.
J72	/TRANJ /	58.
J73	/TRANJ /	58.
J74	/TRANJ /	58.
J75	/TRANJ /	58.
J76	/TRANJ /	58.
J77	/TRANJ /	58.
J78	/TRANJ /	58.
J79	/TRANJ /	58.
J8	/TRANJ /	58.
J80	/TRANJ /	58.
J81	/TRANJ /	58.
J82	/TRANJ /	58.
J83	/TRANJ /	58.
J84	/TRANJ /	58.
J85	/TRANJ /	58.
J86	/TRANJ /	58.
J87	/TRANJ /	58.
J88	/TRANJ /	58.
J89	/TRANJ /	58.
J9	/TRANJ /	58.
J90	/TRANJ /	58.
J91	/TRANJ /	58.
J92	/TRANJ /	58.
J93	/TRANJ /	58.
J94	/TRANJ /	58.
J95	/TRANJ /	58.
J96	/TRANJ /	58.
J97	/TRANJ /	58.
J98	/TRANJ /	58.
J99	/TRANJ /	58.
KDIM	/MODPAR/	2. 16. 30. 32. 58.
KKQ	/TRANS2/	3. 58. 63. 69.
KQ	/TRANS7/	10. 43. 63. 69.
KTF	/TNONEQ/	11. 16. 30. 58.
KTR	/TNONEQ/	11. 16. 30. 58.
KUR	/ /	11. 30. 58.
LC	/ /	16. 35.
LEWIS	/RDMOD /	2. 30. 32. 58.
LIMOUT	/RDOUT /	38. 58.
LPIJ	/ /	21. 26. 30. 32. 35. 39. 58. 69.
LSS	/TRANS5/	69.
MBL	/MODPT /	34. 58.
MEIT'	/RDMAIN/	1. 58.
MODLPT	/MODP2 /	8. 35. 42.
M1	/ /	16. 20. 33. 36.
N	/TRANS8/	43. 69.
NANGLE	/RDWEDG/	2. 30. 58. 67.
NBL	/AREA /	8. 16. 38. 58.
NCS	/COLDSP/	1. 8. 32. 58.

VARIABLE	COMMON	REFERENCING ROUTINES
NIELS	/READAT/	2, 58,
NELT	/ /	16, 27, 30, 58,
NIT	/ /	16, 33,
NKQ	/TRANS7/	7, 10, 43, 63, 69,
NMODPT	/NEWMP /	2, 16, 30, 58,
NN	/TRANS8/	69,
NNKQ	/TRANS2/	3, 58, 63, 69,
NNN	/ /	16, 35,
NNQ	/TRANS2/	7, 58, 63, 69,
NNS	/ /	16, 35,
NOREAC	/GLIM2 /	11, 16, 35,
NOTRAN	/ROMAIN/	1, 32, 38, 58,
NOZZLE	/AREA /*	58,
NOZZO	/RDLIST/	2, 58,
NPRFLS	/AREA /	1, 2, 4, 5, 8, 11, 13, 16, 23, 24, 25, 30, 34,
	/AREA /	35, 38, 58, 65,
NPROFL	/AREA /	2, 16, 58,
NQ	/TRANS7/	7, 43, 63, 69,
NQS	/ /	16, 35, 58,
NQSI	/READAT/	2, 58,
NQT	/ /	16, 27,
NRADLE	/RDWEDG/	2, 30, 58, 67,
NRCOUT	/TAPOUT/	1, 16, 32, 35, 38,
NRECO	/READAT/	2, 58,
NSECT	/AREA /	16, 25, 58,
NSECTS	/READAT/	58,
NSECTU	/AREA /	24, 58,
NSTEPS	/STEPS /	35, 38, 42,
NT	/TNONEQ/	8, 11, 13, 16, 21, 30, 35, 38, 58, 60,
NTEST	/ /	16, 20, 27, 33,
NTS	/MODPT /	16, 30, 34, 58,
NV	/TRANS8/	3, 10, 43, 63, 69,
NWX	/RDWEDG/	2, 30, 58, 67,
OM	/TRANS4/	47, 50,
OMOST	/NEQ /	1, 4, 5, 16, 23, 24, 35, 38, 42,
OMEGA1	/TRANS4/	50, 58,
ORDINP	/ LNE /	8, 16,
P	/ /*	8, 20, 38, 60, 66,
PARAM	/AREA /	16, 24, 25, 58,
PARAMI	/READAT/	58,
PCT	/ /	9, 16, 35, 39, 42,
PCTEST	/ /	16, 27, 35,
PERTGJ	/ /	9, 16, 35, 39, 42,
PGJ	/ /	16, 33,
PI	/ /	11, 16, 35, 38, 39, 42,
PICHI	/ /	11, 16, 21, 35, 38,
PINF	/OUTPUT/*	67,
POM	/TNEQ /	40, 58,
PR	/TRPROP/	1, 8, 16, 32,
PRES	/ /	1, 11, 16, 23, 32, 33, 36, 38, 41, 60,
PRESA	/ /	1, 8, 11, 16, 23, 26, 32, 38, 58, 59, 63,

VARIABLE	COMMON	REFERENCING ROUTINES
PRESAI	/READAT/	58.
PRESB	/ /	16. 23.
PRESTH	/ /	1. 16. 23. 36.
PRF	/TRPROP/	66.
PRHO	/ /	9. 16. 35. 39. 42.
PRP	/RESPRP/	1.
PRREF	/BLOUT /	2. 8. 16. 38.
Q	/TRANS1/	7. 17. 28. 29. 43. 49. 51. 54. 55. 56. 63. 66. 69.
QDRE	/TNCE /	11. 16. 21. 35. 38. 60.
QDPR	/TNCE /	11. 16. 21. 38.
QEMAX	/TNERK /	16.
QM	/ /	16. 20. 26. 33.
QPJ	/COLDSP/	1. 8. 32. 58.
QQ	/ /	11. 16. 30. 58.
QTEST	/RENE /	2. 35. 58.
QWDOT	/BLOUT /	2. 8. 16. 38.
RADLE	/RDWEDG/	2. 30. 55. 67.
READG	/READAT/	2. 58.
READXS	/READAT/	2. 58.
REFF	/BLOUT /	2. 8. 16. 38.
RH	/ /*	8.
RHAP	/ /	1. 8. 11. 16. 21. 26. 32. 38. 39. 59.
RHO	/ /	1. 9. 11. 16. 21. 23. 24. 32. 33. 35. 36. 38. 39.
	/ /	41. 42. 60.
RHOB	/ /	16. 23. 35.
RHOBAR	/ /	16. 23. 33. 41.
RHOC	/ /	16.
RHOINF	/OUTPUT/*	67.
RHOP	/ /	11. 16. 21.
RHPL	/ /	11. 16.
RHTH	/ /	1. 16. 23. 24. 36.
RHPARA	/ /	11. 16. 23. 26. 33. 4[illegible].
RHBARP	/ /	16. 26. 33. 41.
RPRP	/REAC / *	58.
RP1	/REAC /	2. 58.
RP10	/REAC /	2. 58.
RP11	/REAC /	2. 58.
RP12	/REAC /	2. 58.
RP13	/REAC /	2. 58.
RP14	/REAC /	2. 58.
RP15	/REAC /	2. 58.
RP16	/REAC /	2. 58.
RP17	/REAC /	2. 58.
RP18	/REAC /	2. 58.
RP19	/REAC /	2. 58.
RP2	/REAC /	2. 58.
RP20	/REAC /	2. 58.
RP21	/REAC /	2. 58.
RP22	/REAC /	2. 58.
RP23	/REAC /	2. 58.
RP24	/REAC /	2. 58.

VARIABLE	COMMON	REFERENCING ROUTINES
RP25	/REAC /	2. 58.
RP26	/REAC /	2. 58.
RP27	/REAC /	58.
RP28	/REAC /	58.
RP29	/REAC /	58.
RP3	/REAC /	2. 58.
RP30	/REAC /	58.
RP31	/REAC /	58.
RP32	/REAC /	58.
RP33	/REAC /	58.
RP34	/REAC /	58.
RP35	/REAC /	2. 58.
RP36	/REAC /	2. 58.
RP37	/REAC /	2. 58.
RP38	/REAC /	2. 58.
RP39	/REAC /	2. 58.
RP4	/REAC /	2. 58.
RP40	/REAC /	2. 58.
RP41	/REAC /	2. 58.
RP42	/REAC /	2. 58.
RP43	/REAC /	2. 58.
RP44	/REAC /	2. 58.
RP45	/REAC /	2. 58.
RP46	/REAC /	2. 58.
RP47	/REAC /	2. 58.
RP48	/REAC /	2. 58.
RP49	/REAC /	2. 58.
RP5	/REAC /	2. 58.
RP50	/REAC /	2. 58.
RP51	/REAC /	2. 58.
RP52	/REAC /	2. 58.
RP53	/REAC /	2. 58.
RP54	/REAC /	2. 58.
RP55	/REAC /	2. 58.
RP56	/REAC /	2. 58.
RP57	/REAC /	2. 58.
RP58	/REAC /	2. 58.
RP59	/REAC /	2. 58.
RP6	/REAC /	2. 58.
RP60	/REAC /	2. 58.
RP61	/REAC /	2. 58.
RP62	/REAC /	2. 58.
RP63	/REAC /	2. 58.
RP64	/REAC /	2. 58.
RP65	/REAC /	2. 58.
RP66	/REAC /	2. 58.
RP67	/REAC /	2. 58.
RP68	/REAC /	2. 58.
RP69	/REAC /	2. 58.
RP7	/REAC /	2. 58.

VARIABLE	COMMON	REFERENCING ROUTINES
RP70	/REAC /	2. 58.
RP71	/REAC /	2. 58.
RP72	/REAC /	2. 58.
RP73	/REAC /	2. 58.
RP74	/REAC /	2. 58.
RP75	/REAC /	2. 58.
RP76	/REAC /	2. 58.
RP77	/REAC /	2. 58.
RP78	/REAC /	2. 58.
RP79	/REAC /	2. 58.
RP8	/REAC /	2. 58.
RP80	/REAC /	2. 58.
RP81	/REAC /	2. 58.
RP82	/REAC /	2. 58.
RP83	/REAC /	2. 58.
RP84	/REAC /	2. 58.
RP85	/REAC /	2. 58.
RP86	/REAC /	2. 58.
RP87	/REAC /	2. 58.
RP88	/REAC /	2. 58.
RP89	/REAC /	2. 58.
RP9	/REAC /	2. 58.
RP90	/REAC /	2. 58.
RP91	/REAC /	2. 58.
RP92	/REAC /	2. 58.
RSA	/THRT /	16. 25. 35. 58. 65.
RTHCM	/AREA /	1. 4. 5. 8. 16. 24. 25. 38. 58.
RC	/BL /	1. 4. 5. 8. 16. 24. 38. 57.
SAJ	/ /	16. 27. 64.
SAVEC	/TEMPRY/	1. 8. 9. 16. 23. 32. 38. 63. 66.
SBJ	/ /*	30. 58.
SC	/ /	16. 35.
SCPG	/ /	8. 11. 16. 21.
SCPGH	/TNCE /	8. 11. 16. 21.
SDCHA	/TNERK /	35. 60.
SDCHI	/ /	16. 35. 39. 42.
SDGJ	/ /*	35. 60.
SDQ	/ /*	60.
SDT	/ /	16. 35. 60.
SDTE	/TNERK /	16. 35. 60.
SEN	/ /	1. 11. 16. 26. 35. 38.
SENT	/ /	11. 13. 16. 20. 32. 33. 39. 41. 63. 64.
SHJ	/ /	11. 13. 16. 20. 21. 32. 33. 39. 41. 63. 64.
SHJA	/ /	16. 27. 32. 63. 64.
SHJAP	/ /*	30. 58.
SHJDGJ	/DRV /	8. 16. 21.
SHPG	/ /	11. 16. 41.
SHV	/STAG /*	32.
SIGMA	/TRPROP/	1. 16. 23. 32. 39. 66.
SIGR	/RESPRP/	1. 23.

VARIABLE	COMMON	REFERENCING ROUTINES
SKIL	/ /	16. 20. 33.
SL	/ /	16. 35. 58.
SL64	/ /	11. 16. 35.
SM	/ /	1. 16. 23. 24. 35. 36. 38. 59.
SMASS	/MASSFL/	1. 58. 59.
SN	/BLOUT /	2. 8. 16. 38.
SPRP	/SPEC /	27. 58. 63.
SP1	/SPEC /	2. 58.
SP10	/SPEC /	2. 58.
SP11	/SPEC /	2. 58.
SP12	/SPEC /	2. 58.
SP13	/SPEC /	2. 58.
SP14	/SPEC /	2. 58.
SP15	/SPEC /	2. 58.
SP16	/SPEC /	2. 58.
SP17	/SPEC /	2. 58.
SP18	/SPEC /	2. 58.
SP19	/SPEC /	2. 58.
SP2	/SPEC /	2. 58.
SP20	/SPEC /	2. 58.
SP21	/SPEC /	2. 58.
SP22	/SPEC /	2. 58.
SP23	/SPEC /	2. 58.
SP24	/SPEC /	2. 58.
SP25	/SPEC /	2. 58.
SP26	/SPEC /	2. 58.
SP27	/SPEC /	2. 58.
SP28	/SPEC /	2. 58.
SP29	/SPEC /	58.
SP3	/SPEC /	2. 58.
SP30	/SPEC /	58.
SP4	/SPEC /	2. 58.
SP5	/SPEC /	2. 58.
SP6	/SPEC /	2. 58.
SP7	/SPEC /	2. 58.
SP8	/SPEC /	2. 58.
SP9	/SPEC /	2. 58.
SQRTA	/AEGEOM/	4. 16. 24.
SS	/ /	16. 23.
SU	/ /	1. 8. 11. 16. 21. 23. 24. 32. 33. 38. 41. 42. 60.
SUMG	/ /	11. 16. 21.
SUMGH	/TNCE /	11. 16. 21.
SUPGO	/RDMAIN/	1. 30. 58.
SU2	/ /	11. 16. 21. 33. 39. 41.
SW	/BL /	1. 8. 15.
S1	/AEGEOM/	4. 16. 24. 65.
S2	/AEGEOM/	4. 16. 24. 65.
T	/TRANS1/	44. 45. 46. 48. 50. 52. 53. 66. 69.
TAUW	/BLOUT /	2. 8. 16. 38.
TO	/ /	16. 35.

VARIABLE	COMMON	REFERENCING ROUTINES
TBT	/OUTMOD/	32. 38.
TEST	/ /	16. 20. 27. 33.
TESTB	/ /	16. 23. 27. 36.
TETEST	/RENE /	2. 35. 58.
TFA	/ /	16. 30. 58. 64.
TFB	/ /	16. 27. 30. 58. 64.
TFC	/ /	16. 27. 30. 58. 64.
TFD	/ /	16. 27. 30. 58. 64.
TFE	/ /	16. 27. 30. 58. 64.
TFK	/ /	16. 30. 58. 64.
THETA	/BLOUT /	2. 8. 10. 38.
THEV	/ /	16. 27. 64.
THEVE	/POLYAT/	27. 30. 58. 64.
THEVP	/ /*	30. 58.
TINF	/OUTPUT/*	67.
TL	/TRANS4/	47. 58.
TLIST	/TNEQ /	40. 58.
TL1	/TRANS4/	3.
TL10	/TRANS4/	
TL11	/TRANS4/	3.
TL12	/TRANS4/	3.
TL13	/TRANS4/	3.
TL14	/TRANS4/	3.
TL15	/TRANS4/	3.
TL16	/TRANS4/	3.
TL17	/TRANS4/	
TL18	/TRANS4/	
TL19	/TRANS4/	
TL2	/TRANS4/	3.
TL20	/TRANS4/	3.
TL21	/TRANS4/	3.
TL22	/TRANS4/	3.
TL23	/TRANS4/	3.
TL24	/TRANS4/	3.
TL25	/TRANS4/	3.
TL26	/TRANS4/	3.
TL27	/TRANS4/	
TL28	/TRANS4/	
TL29	/TRANS4/	
TL3	/TRANS4/	3.
TL30	/TRANS4/	3.
TL31	/TRANS4/	3.
TL32	/TRANS4/	3.
TL33	/TRANS4/	3.
TL34	/TRANS4/	3.
TL35	/TRANS4/	3.
TL36	/TRANS4/	3.
TL37	/TRANS4/	
TL38	/TRANS4/	
TL39	/TRANS4/	

VARIABLE	COMMON	REFERENCING ROUTINES
TL4	/TRANS4/	3.
TL40	/TRANS4/	3.
TL5	/TRANS4/	3.
TL6	/TRANS4/	3.
TL7	/TRANS4/	
TL8	/TRANS4/	
TL9	/TRANS4/	
TMODEL	/MODPAR/	2. 16. 30. 32. 58.
TNFP	/TNF /*	58.
TN1	/TNF /	2.
TN2	/TNF /	2.
TPLATE	/MODPAR/	2. 30. 58.
TPRINT	/ /	16. 35. 58.
TPRNTI	/READAT/	2. 58.
TPROP	/ /*	63.
TR0	/TRPROP/*	8.
TS	/STAG /	16. 32.
TSAR	/MODPT /	16. 34. 58.
TSDIAM	/MODPT /	2. 16. 30. 58.
TSF	/STAG /	16.
TSTOP	/ /	16. 23. 27. 35. 58.
TSTOPI	/MASSFL/	2. 27. 58.
TT	/ /*	63.
TTEST	/RENE /	2. 35. 58.
TWALL	/BL /	1. 2. 8. 16. 38. 58.
TWEDGE	/RDWEDG/	2. 30. 58. 67.
T2E	/STAG /	23. 32. 35.
T2F	/STAG /	23. 32. 35.
U	/ /*	8.
UINF	/OUTPUT/*	67.
UP	/ /	16. 35.
US	/SS /	1. 8. 21. 23. 42.
V	/TRANS7/	10. 43. 63. 69.
VISC	/TRPROP/	1. 8. 16. 23. 32. 38. 66. 67.
VISCR	/RESPRP/	1. 23.
VISROT	/BL /	1. 16.
VMUINF	/OUTPUT/*	67.
VV	/TRANV /	63. 69.
VV1	/TRANV /	3.
VV10	/TRANV /	3.
VV100	/TRANV /	
VV11	/TRANV /	3.
VV12	/TRANV /	3.
VV13	/TRANV /	3.
VV14	/TRANV /	3.
VV15	/TRANV /	3.
VV16	/TRANV /	3.
VV17	/TRANV /	3.
VV18	/TRANV /	3.
VV19	/TRANV /	3.

VARIABLE	COMMON	REFERENCING ROUTINES
VV2	/TRANV /	3.
VV20	/TRANV /	3.
VV21	/TRANV /	3.
VV22	/TRANV /	3.
VV23	/TRANV /	3.
VV24	/TRANV /	3.
VV25	/TRANV /	3.
VV26	/TRANV /	3.
VV27	/TRANV /	3.
VV28	/TRANV /	3.
VV29	/TRANV /	3.
VV3	/TRANV /	3.
VV30	/TRANV /	3.
VV31	/TRANV /	3.
VV32	/TRANV /	3.
VV33	/TRANV /	3.
VV34	/TRANV /	3.
VV35	/TRANV /	3.
VV36	/TRANV /	3.
VV37	/TRANV /	3.
VV38	/TRANV /	3.
VV39	/TRANV /	3.
VV4	/TRANV /	3.
VV40	/TRANV /	3.
VV41	/TRANV /	3.
VV42	/TRANV /	3.
VV43	/TRANV /	3.
VV44	/TRANV /	3.
VV45	/TRANV /	3.
VV46	/TRANV /	3.
VV47	/TRANV /	3.
VV48	/TRANV /	3.
VV49	/TRANV /	3.
VV5	/TRANV /	3.
VV50	/TRANV /	3.
VV51	/TRANV /	3.
VV52	/TRANV /	3.
VV53	/TRANV /	3.
VV54	/TRANV /	3.
VV55	/TRANV /	3.
VV56	/TRANV /	3.
VV57	/TRANV /	3.
VV58	/TRANV /	3.
VV59	/TRANV /	3.
VV6	/TRANV /	3.
VV60	/TRANV /	3.
VV61	/TRANV /	3.
VV62	/TRANV /	3.
VV63	/TRANV /	3.
VV64	/TRANV /	

RIABLE	COMMON	REFERENCING ROUTINES
VV65	/TRANV /	
VV66	/TRANV /	
VV67	/TRANV /	
VV68	/TRANV /	
VV69	/TRANV /	
VV7	/TRANV /	3.
VV70	/TRANV /	
VV71	/TRANV /	
VV72	/TRANV /	
VV73	/TRANV /	
VV74	/TRANV /	
VV75	/TRANV /	
VV76	/TRANV /	
VV77	/TRANV /	
VV78	/TRANV /	
VV79	/TRANV /	
VV8	/TRANV /	3.
VV80	/TRANV /	
VV81	/TRANV /	
VV82	/TRANV /	
VV83	/TRANV /	
VV84	/TRANV /	
VV85	/TRANV /	
VV86	/TRANV /	
VV87	/TRANV /	
VV88	/TRANV /	
VV89	/TRANV /	
VV9	/TRANV /	3.
VV90	/TRANV /	
VV91	/TRANV /	
VV92	/TRANV /	
VV93	/TRANV /	
VV94	/TRANV /	
VV95	/TRANV /	
VV96	/TRANV /	
VV97	/TRANV /	
VV98	/TRANV /	
VV99	/TRANV /	
V1	/TRANV /	58.
V10	/TRANV /	58.
V100	/TRANV /	58.
V11	/TRANV /	58.
V12	/TRANV /	58.
V13	/TRANV /	58.
V14	/TRANV /	58.
V15	/TRANV /	58.
V16	/TRANV /	58.
V17	/TRANV /	58.
V18	/TRANV /	58.
V19	/TRANV /	58.

VARIABLE	COMMON	REFERENCING ROUTINES
V2	/TRANV /	58.
V20	/TRANV /	58.
V21	/TRANV /	58.
V22	/TRANV /	58.
V23	/TRANV /	58.
V24	/TRANV /	58.
V25	/TRANV /	58.
V26	/TRANV /	58.
V27	/TRANV /	58.
V28	/TRANV /	58.
V29	/TRANV /	58.
V3	/TRANV /	58.
V30	/TRANV /	58.
V31	/TRANV /	58.
V32	/TRANV /	58.
V33	/TRANV /	58.
V34	/TRANV /	58.
V35	/TRANV /	58.
V36	/TRANV /	58.
V37	/TRANV /	58.
V38	/TRANV /	58.
V39	/TRANV /	58.
V4	/TRANV /	58.
V40	/TRANV /	58.
V41	/TRANV /	58.
V42	/TRANV /	58.
V43	/TRANV /	58.
V44	/TRANV /	58.
V45	/TRANV /	59.
V46	/TRANV /	58.
V47	/TRANV /	58.
V48	/TRANV /	58.
V49	/TRANV /	58.
V5	/TRANV /	58.
V50	/TRANV /	58.
V51	/TRANV /	58.
V52	/TRANV /	58.
V53	/TRANV /	58.
V54	/TRANV /	58.
V55	/TRANV /	59.
V56	/TRANV /	58.
V57	/TRANV /	58.
V58	/TRANV /	58.
V59	/TRANV /	58.
V6	/TRANV /	58.
V60	/TRANV /	58.
V61	/TRANV /	58.
V62	/TRANV /	58.
V63	/TRANV /	58.
V64	/TRANV /	58.

RIABLE	COMMON	REFERENCING ROUTINES
V65	/TRANV /	58.
V66	/TRANV /	58.
V67	/TRANV /	58.
V68	/TRANV /	58.
V69	/TRANV /	58.
V7	/TRANV /	58.
V70	/TRANV /	58.
V71	/TRANV /	58.
V72	/TRANV /	58.
V73	/TRANV /	58.
V74	/TRANV /	58.
V75	/TRANV /	58.
V76	/TRANV /	58.
V77	/TRANV /	58.
V78	/TRANV /	58.
V79	/TRANV /	58.
V8	/TRANV /	58.
V80	/TRANV /	58.
V81	/TRANV /	58.
V82	/TRANV /	58.
V83	/TRANV /	58.
V84	/TRANV /	58.
V85	/TRANV /	58.
V86	/TRANV /	58.
V87	/TRANV /	58.
V88	/TRANV /	58.
V89	/TRANV /	58.
V9	/TRANV /	58.
V90	/TRANV /	58.
V91	/TRANV /	58.
V92	/TRANV /	58.
V93	/TRANV /	58.
V94	/TRANV /	58.
V95	/TRANV /	58.
V96	/TRANV /	58.
V97	/TRANV /	58.
V98	/TRANV /	58.
V99	/TRANV /	58.
W	/SWITCH/	8. 16. 23. 35.
WEDGEM	/RDWEDG/	30. 32. 58.
WK	/RDWEDG/	2. 30. 58. 67.
WSAVE	/AVG /	2. 23. 35. 58.
WX1	/RDWEDG/	2. 30. 58. 67.
WX1	/RDWEDG/	2. 30. 58. 67.
X	/ /*	8. 25. 38. 66.
XI	/BLNF /	8. 16.
XMJAT	/ /	13. 16. 20. 33. 63. 64.
XMJATO	/TNDC /	11. 13. 16.
XMODP1	/MODPAR/	2. 16. 30. 58.
XMP1	/MODPAR/	16. 23. 35. 58.

VARIABLE	COMMON	REFERENCING ROUTINES
XNUI	/ /	11. 16. 35.
XNUIJ	/ /	11. 30. 35. 58.
XNUIJP	/ /	11. 30. 35. 58.
XP	/SWITCH/*	8. 23.
XPB	/SWITCH/	16. 23. 35.
XSN	/BLOUT /	2. 8. 16. 35. 38.
XXX	/TAPOUT/	1. 38.
XZERO	/BL /	8. 16. 23. 58.
XZEROI	/READAT/	58.
YOZO	/BLRAD /	8. 57.
ZCAP	/EQC /	16. 20. 26. 32.
ZCH	/EQC /	16. 20. 26. 32.
ZCM	/EQC /	16. 20. 26. 32.
ZGMU	/EQC2 /	20. 26.
ZM2	/TRANS1/	28. 66. 69.
ZP	/ /	16. 33. 41.
ZPA	/ /	16. 23. 26. 33.
ZPRO	/NOZZ /*	58.
ZPZ	/EQC2 /	20. 26.
ZP1	/NOZZ /	2.
ZP10	/NOZZ /	2.
ZP11	/NOZZ /	2.
ZP12	/NOZZ /	2.
ZP13	/NOZZ /	2.
ZP14	/NOZZ /	
ZP15	/NOZZ /	
ZP16	/NOZZ /	
ZP17	/NOZZ /	
ZP18	/NOZZ /	
ZP19	/NOZZ /	
ZP2	/NOZZ /	2.
ZP20	/NOZZ /	
ZP3	/NOZZ /	2.
ZP4	/NOZZ /	2.
ZP5	/NOZZ /	2.
ZP6	/NOZZ /	2.
ZP7	/NOZZ /	2.
ZP8	/NOZZ /	2.
ZP9	/NOZZ /	2.
ZRBP	/EQC /	16. 20. 26.
ZRHO	/EQC /	16. 20. 26. 32.
ZSEN	/EQC /	16. 20. 26. 32.

COMPLETED

6. LISTING OF THE NATA CODE

The present section consists of a complete source program listing of the NATA code. The routines appear in the same order in which their names are listed at the beginning of Section 5, i.e., main program, block data routines, and then all subroutines and functions in alphanumeric order of their names.

```
C ~  MAIN PROGRAM FOR NOZZLE FLOW SOLUTION ——                                   MAI    1
     LOGICAL SUPGO,DATAPE,NOTRAN                                                MAI    2
     LOGICAL ERR                                                                MAI    3
     REAL ET,ETO                                                                MAI    4
     REAL ACOM(30),ELMENT(10),HP(20)                                            MAI    5
     DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                                       MAI    6
     COMMON AA(22,24),CDIJ(20,10),CAPX(20)                                      MAI    7
     COMMON        A, AFNTS,   AFNX, AMACH,     AR,   ARBA,   ARBB, BZERO,     MAI    8
    1      C,  CARB,     CH,    CHA,   CLNT,     CM,    CMA,    CRA,    CRP,   MAI    9
    2  CRRB,    CRS,   CSTA,     CT,   CTAP,    CTB,    CTC, CTMAX, CTMXX,     MAI   10
    3   CTP,   CTPL,    CTT,     CX,    CXB, CXMAX,DATEST,DBTEST, DELT1,       MAI   11
    4 DELT2,DELTAX, DLOGA, DLOGR,     DT,    ENT,   FLUX, HDELX,    PCT,     MAI   12
    5PCTEST,   PRES, PRESA, PRESB,PRESTH,   PRHO,   RHAP,    RHO,   RHOB,     MAI   13
    6RHOBAR,   RHOC,   RHOP,   RHPL,  RHTH,ROBARA,ROBARP,   SCPG,    SDT,     MAI   14
    7   SEN,   SHPG,     SC,     SL,  SL64,     SM,     SU,    SU2,   SUMG,   MAI   15
    8  TEST, TESTB,TPRINT, TSTOP,     UP,     ZP,    ZPA                       MAI   16
     COMMON          BE(64),    BET(20),  BLBK(31),    CAI(64),CAPXTH(20),     MAI   17
    1  CCPJ(20), CFACT(64),    CGI(20),  CGMU(20),    CHI(64),  CHII(20),     MAI   18
    2CLN1MC(64), CLNPI(64),    CMW(20),  ETAI(64),   FTAJ(20),    GJA(20),     MAI   19
    3   GJB(20),PERTGJ(20),    PGJ(20),     PI(64), PICHI(64),    QM(20),     MAI   20
    4    QSQ(20),    SAJ(20), SDCHI(64),  SENT(20),    SHJ(20),  SHJA(20),     MAI   21
    5  SKIL(20),     SS(20),     TB(30),    TFA(20),    TFB(20),    TFC(20),     MAI   22
    6   TFD(20),    TFE(20),    TFK(20),  THEV(20), XMJAT(20),    XNUI(64)     MAI   23
     COMMON          B TA(64,20),   ELJ(10,20),  GELJ(10,20),                   MAI   24
    1 XNUIJ(64,20),XNUIJP(64,20)                                                MAI   25
     COMMON       IC,     IM,   INEQ,  INEQV,     IP,   IRUN,    ISC,  ISCP1,   MAI   26
    1   ISMC,ISMCNR,    ISR,    ISS,  ISSNR,  ISSP1,  ISSP2,  ISSP3,  ISSP4,   MAI   27
    2 ISW1A,  ISW1B,  ISW2A,  ISW2B,  ISW3A,  ISW3B,  ISW4A,  ISW4B,  ISW5A,     MAI   28
    3 ISW5B,  ISW6A,  ISW6B,   IUPD,  IZERO,    JJK,     LC,     M1,    NFIT,   MAI   29
    4    NIT,    NNN,    NNS,    NQS,    NQT,  NTEST                           MAI   30
     COMMON    IGJ(20),    IJA(20),    ITB(5),    KUR(64,20),   LPIJ(20,10)    MAI   31
     COMMON ACOM,ELMENT,HP                                                      MAI   32
     DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBMAI   33
    1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                         MAI   34
     EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),                       MAI   35
    1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),               MAI   36
    2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),                    MAI   37
    3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                               MAI   38
     EQUIVALENCE (NOZZLE,NPROFL(1))                                             MAI   39
     COMMON /ERROR/ ERR                                                         MAI   40
     COMMON /TEMPRY/ SAVEC(20)                                                  MAI   41
     COMMON /MASSFL/ SMASS,CTMXXI,TSTOPI,IS(20)                                 MAI   42
     COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),           MAI   43
    1      SW,RO,U,IM,IPCINT                                                   MAI   44
     DIMENSION GMF(20)                                                          MAI   45
     COMMON /DOMAIN/ HS,SUPGO,MFITER,NOTRAN                                     MAI   46
     COMMON /NEQ/ ONDST(2),DDELBL(2)                                            MAI   47
     COMMON /COLDCP/ CGMW,QPJ(10),IJCS(10),NCS                                  MAI   48
     COMMON /TRPROP/ VISC,PR,SIGMA,FLEWIS                                       MAI   49
     COMMON /RESPRP/ VISCR,PRR,SIGR,FLEWR                                       MAI   50
     COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAPE              MAI   51
     COMMON /AREA/ API(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),                   MAI   52
    1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                         MAI   53
     COMMON /RELIST/ IGAS,IGASO,NOZZO,IMANO                                     MAI   54
     COMMON /CONVRT/ CF(6)                                                      MAI   55
```

```
      COMMON /OUTPUT/ FVOUT(35),GJMF(20)                                MAI   56
      COMMON /RDOUT/ FLOW,FACNAM,CHANAM,LIMOU                           MAI   57
      COMMON /SS/ CAS,US                                                MAI   58
      COMMON /SPEC/ SPRP(43,30)                                         MAI   59
      COMMON /LN/ ISATOM,ISMOL,JATOM,JMOL                               MAI   60
      COMMON /EELEM/ EEP(2,10)                                          MAI   61
      COMMON /THRT/ RSA                                                 MAI   62
      DATA NCASE,NERROR,NRCTOT,IRCO /4*0/, ONE /1./, PIV /3.1A159/      MAI   63
      ISW1A=1                                                           MAI   64
      ISW2A=1                                                           MAI   65
      ISW3A=1                                                           MAI   66
      ISW4A=0                                                           MAI   67
      ISW5A=0                                                           MAI   68
      ISW6A=0                                                           MAI   69
      ISW1B=0                                                           MAI   70
      ISW2B=0                                                           MAI   71
      IS3=1                                                             MAI   72
      ISW4B=0                                                           MAI   73
      ISW5B=0                                                           MAI   74
      ISW6B=1                                                           MAI   75
      IRUN=0                                                            MAI   76
      MFITER=1                                                          MAI   77
      NOTRAN=.FALSE.                                                    MAI   78
10    CALL ELTIME (ET0,0)                                               MAI   79
      ISW3B=IS3                                                         MAI   80
      CALL READ                                                         MAI   81
      IS3=ISW3B                                                         MAI   82
      IF (.NOT.NOTRAN) GO TO 20                                         MAI   83
      ISW3B=0                                                           MAI   84
      VISC=0.                                                           MAI   85
      PR=0.                                                             MAI   86
      SIGMA=0.                                                          MAI   87
      FLEWIS=0.                                                         MAI   88
20    CALL LIST                                                         MAI   89
      IF (NOTRAN) WRITE (6,330)                                         MAI   90
      ET=ET0                                                            MAI   91
      CALL ELTIME (ET,1)                                                MAI   92
      NRCOUT=0                                                          MAI   93
      NCASE=NCASE+1                                                     MAI   94
      ERR=.FALSE.                                                       MAI   95
      OMDST(1)=1.                                                       MAI   96
      OMDST(2)=1.                                                       MAI   97
      IF (ISW2B.GT.0) SM=0.                                             MAI   98
      CALL INGAS                                                        MAI   99
      IF (ERR) GO TO 290                                                MAI  100
      IF (ISW2B.LE.0) CALL RESTMP                                       MAI  101
      IF (ERR) GO TO 290                                                MAI  102
      IF (ISW2B.LT.0) GO TO 30                                          MAI  103
      CALL INIT                                                         MAI  104
      CALL INTA                                                         MAI  105
      IF (ERR) GO TO 290                                                MAI  106
30    IF (ISW6A.GE.0) GO TO 40                                          MAI  107
      CALL STUNTS                                                       MAI  108
      IRCO=1                                                            MAI  109
      GO TO 300                                                         MAI  110
```

```
      IF (ISW6A.EQ.2) GO TO 140                                          MAI 111
      IF (SUPGO.AND.IRCO.EQ.0) GO TO 50                                  MAI 112
      IF (NOTRAN) GO TO 60                                               MAI 113
      CALL TRANSX                                                        MAI 114
      IF (ERR) GO TO 290                                                 MAI 115
      IRCO=0                                                             MAI 116
50    IF (ISW1B.EQ.0) GO TO 60                                           MAI 117
      CALL STUNT2                                                        MAI 118
      IF (ISW1B.LT.0) GO TO 140                                          MAI 119
60    CT=TWALL/CTAP                                                      MAI 120
      CALL THERM1                                                        MAI 121
      CPWALL=0.                                                          MAI 122
      DO 70 K=1,NCS                                                      MAI 123
      I=IJCS(K)                                                          MAI 124
70    CPWALL=CPWALL+QPJ(K)*CCPJ(I)                                       MAI 125
      CPWALL=CPWALL*CRA/CGMW                                             MAI 126
      WRITE (6,360) CPWALL,TWALL                                         MAI 127
      IF (NCASE.EQ.1) CALL OUT                                           MAI 128
      IF (NPRFLS.EQ.2) GO TO 80                                          MAI 129
      RO=RTHCM(1)                                                        MAI 130
      GO TO 90                                                           MAI 131
80    RO=SQRT(1.27324*RTHCM(1)*RTHCM(2))                                 MAI 132
90    IF (ISW3B.EQ.0.OR.ISW2B.GT.0.OR.MFITER.EQ.0) GO TO 140             MAI 133
      DO 100 I=1,ISS                                                     MAI 134
100   SAVEC(I)=GJA(I)*CMA                                                MAI 135
      CALL TRANSP (CTAP,PRESA)                                           MAI 136
      IF (ERR) GO TO 290                                                 MAI 137
      VISROT=VISC/CTAP                                                   MAI 138
      HOP=CHA*CRP/CMA                                                    MAI 139
      SW=CPWALL*TWALL/HOP-1.                                             MAI 140
      US=SQRT(8.31436E7*CTAP/CMA)                                        MAI 141
      CALL EQUIL (1)                                                     MAI 142
      IF (NPRFLS.EQ.1) GO TO 110                                         MAI 143
      ATH=4.*(RTHCM(1)-RO*DELBL(1))*(RTHCM(2)-RO*DELBL(2))               MAI 144
      GO TO 130                                                          MAI 145
110   IF (JDIM.EQ.0) GO TO 120                                           MAI 146
      ATH=PIV*(RO*OMDST(1))**2                                           MAI 147
      GO TO 130                                                          MAI 148
120   ATH=2.*RO*OMDST(1)                                                 MAI 149
130   SMASS=453.59*FLOW/ATH                                              MAI 150
      CALL RESTMP                                                        MAI 151
      IF (ISW2B.LT.0) GO TO 140                                          MAI 152
      CALL INIT                                                          MAI 153
      CALL INTA                                                          MAI 154
140   CF(1)=RHAP*(30.48)**3/453.59                                       MAI 155
      US=SQRT(8.31436E7*CTAP/CMA)                                        MAI 156
      CF(2)=US/30.48                                                     MAI 157
      CF(3)=30.48/453.59                                                 MAI 158
      CF(4)=1./2.54                                                      MAI 159
      CF(5)=1.8*CTAP*1.98647/CMA                                         MAI 160
      CF(6)=RO/2.54                                                      MAI 161
      DO 150 I=1,ISS                                                     MAI 162
150   SAVEC(I)=GJA(I)*CMA                                                MAI 163
      CT=1.                                                              MAI 164
      PRES=1.                                                            MAI 165
```

```
      RHO=1.                                                            MAI 166
      SU=0.                                                             MAI 167
      CH=CHA                                                            MAI 168
      CM=CMA                                                            MAI 169
      WRITE (6,370)                                                     MAI 170
      CALL OUT1                                                         MAI 171
      IF (ERR) GO TO 290                                                MAI 172
      CALL ELTIME (ET,1)                                                MAI 173
      IF (ISW6A.EQ.2.OR.ISW18.LT.0) GO TO 300                           MAI 174
      IF (NOTRAN) GO TO 160                                             MAI 175
      CALL TRANSP (CTAP,PRESA)                                          MAI 176
      IF (ERR) GO TO 290                                                MAI 177
      VISCC=VISC*CF(3)                                                  MAI 178
      WRITE (6,380) VISCC,PR,SIGMA,FLEWIS                               MAI 179
  160 IF (ISW6A.NE.0) GO TO 300                                         MAI 180
      VISCR=VISC                                                        MAI 181
      PRR=PR                                                            MAI 182
      SIGR=SIGMA                                                        MAI 183
      FLEWR=FLEWIS                                                      MAI 184
      VISROT=VISC/CTAP                                                  MAI 185
      HOP=CHA*CRP/CMA                                                   MAI 186
      SW=CPWALL*TWALL/HOP-1.                                            MAI 187
      IF (ISW1A.EQ.IZERO) GO TO 190                                     MAI 188
      IF (.NOT.DATAPE) GO TO 180                                        MAI 189
      ISOLN=1                                                           MAI 190
  170 ITYPER=1                                                          MAI 191
      XXX(1)=NOZZLE+0.1                                                 MAI 192
      XXX(2)=FLOW                                                       MAI 193
      XXX(3)=PRESA                                                      MAI 194
      XXX(4)=CF(5)*CHA                                                  MAI 195
      XXX(5)=DIAM(1)                                                    MAI 196
      XXX(6)=CTAP                                                       MAI 197
      XXX(7)=CF(1)                                                      MAI 198
      XXX(8)=SEN                                                        MAI 199
      XXX(9)=CMA                                                        MAI 200
      XXX(10)=DIAM(2)                                                   MAI 201
      NRCOUT=NRCOUT+1                                                   MAI 202
      WRITE (ITPOUT) ITYPER,ISOLN,(XXX(I),I=1,10)                       MAI 203
      GO TO (180,210,280), ISOLN                                        MAI 204
  180 CALL FROZEN                                                       MAI 205
      IF (ERR) GO TO 290                                                MAI 206
      CT=CTMAX                                                          MAI 207
      PRES=PRESTH                                                       MAI 208
      RHO=RHTH                                                          MAI 209
      SU=SM/RHTH                                                        MAI 210
      CH=CHA-0.5*SU**2                                                  MAI 211
      WRITE (6,390)                                                     MAI 212
      CALL OUT1                                                         MAI 213
      CALL ELTIME (ET,1)                                                MAI 214
  190 CALL NRMAX                                                        MAI 215
      IF (ERR) GO TO 290                                                MAI 216
      PRESTH=PRES                                                       MAI 217
      UTH=SU                                                            MAI 218
      CHTH=CH                                                           MAI 219
      CMTH=CM                                                           MAI 220
```

```
      DO 200 I=1,ISS                                                    MAI  221
2     GMF(I)=GJ(I)*CM                                                   MAI  222
      IF (ISW3A.EQ.0) GO TO 220                                         MAI  223
      ISOLN=2                                                           MAI  224
      IF (DATAPE) GO TO 170                                             MAI  225
210   CALL EQUIL (2)                                                    MAI  226
      IF (ERR) GO TO 290                                                MAI  227
      IM=1                                                              MAI  228
      JJK=0                                                             MAI  229
      ISMCNR=ISMC                                                       MAI  230
      ISSNR=ISS                                                         MAI  231
220   WRITE (6,400)                                                     MAI  232
      CT=CTMAX                                                          MAI  233
      PRES=PRESTH                                                       MAI  234
      RHO=RHTH                                                          MAI  235
      SU=UTH                                                            MAI  236
      CH=CHTH                                                           MAI  237
      CM=CMTH                                                           MAI  238
      DO 230 I=1,ISS                                                    MAI  239
230   SAVEC(I)=GMF(I)                                                   MAI  240
      CALL OUT1                                                         MAI  241
      IF (ERR) GO TO 290                                                MAI  242
      CALL ELTIME (ET,1)                                                MAI  243
      IF (ISW2A.EQ.IZERO) GO TO 300                                     MAI  244
      DO 240 I=1,ISS                                                    MAI  245
240   GJ(I)=GJA(I)                                                      MAI  246
      PRES=1.0                                                          MAI  247
      N=0                                                               MAI  248
      SUM1=ALOG(RHTH)                                                   MAI  249
      A=4.+8.*SUM1                                                      MAI  250
250   N=N+1                                                             MAI  251
      IF (N-50) 260,270,270                                             MAI  252
260   SUM2=A+2.                                                         MAI  253
      SUM3=RHTH**A                                                      MAI  254
      D=SUM3*SUM2-2.                                                    MAI  255
      D1=SUM3*(SUM2*SUM1+1.)                                            MAI  256
      ATEST=A                                                           MAI  257
      A=A-D/D1                                                          MAI  258
      IF (ABS(D).GT..00001) GO TO 250                                   MAI  259
      IF (ABS(ATEST-A).GT..00001) GO TO 250                             MAI  260
270   C=(RHTH**2)*(1.-RHTH**A)                                          MAI  261
      DELT=DELT1                                                        MAI  262
      IF (ISW3B.NE.0) DELT=0.049*(1.-CTMAX)                             MAI  263
      CT=1.0-DELT                                                       MAI  264
      CM=CMA                                                            MAI  265
      CALL NEWRAP                                                       MAI  266
      IF (ERR) GO TO 290                                                MAI  267
      AFNTS=SM/FLUX                                                     MAI  268
      CALL FINDX (AFNTS,-ONE,CX)                                        MAI  269
      WRITE (6,340) A,C                                                 MAI  270
      ISOLN=3                                                           MAI  271
      IF (DATAPE) GO TO 170                                             MAI  272
280   CALL NONEQ                                                        MAI  273
      CALL ELTIME (ET,1)                                                MAI  274
      IF (.NOT.ERR) GO TO 300                                           MAI  275
```

```
      CALL DUMPEX                                                       MAI  276
      NERROR=NERROR+1                                                   MAI  277
300   NRCTOT=NRCTOT+NRCOUT                                              MAI  278
      WRITE (6,420) IRUN,NCASE,NRCOUT,ITPOUT,NRCTOT                     MAI  279
      ELTMIN=(ET-ET0)/60.                                               MAI  280
      WRITE (6,410) ELTMIN                                              MAI  281
      IF (ISW4A) 10,310,10                                              MAI  282
310   NCOMPL=NCASE-NERROR                                               MAI  283
      WRITE (6,350) NCOMPL,NERROR                                       MAI  284
      IF (.NOT.DATAPE) GO TO 320                                        MAI  285
      END FILE ITPOUT                                                   MAI  286
      REWIND ITPOUT                                                     MAI  287
320   CALL EXIT                                                         MAI  288
C                                                                       MAI  289
C                                                                       MAI  290
C                                                                       MAI  291
330   FORMAT (43H0TRANSPORT PROPERTY CALCULATIONS SUPPRESSED)           MAI  292
340   FORMAT (20H0 DENSITY FIT-ALPHA=,1PE15.7,10H CONSTANT=,E15.7)      MAI  293
350   FORMAT (///I5,16H CASES COMPLETED/I5,13H CASES FAILED)            MAI  294
360   FORMAT (27H0SPECIFIC HEAT OF COLD GAS=,F7.4,16H BTU/LB-DEG R AT,F8MAI  295
     1.2,6H DEG K)                                                      MAI  296
370   FORMAT (1H1,9X,24H- RESERVOIR CONDITIONS -)                       MAI  297
380   FORMAT (///31H RESERVOIR TRANSPORT PROPERTIES//19H VISCOSITY      MAI  298
     1  =,3X,1PE9.2,11H LBM/FT-SEC/19H PRANDTL NUMBER    =,0PF12.5/19H SIMAI  299
     2GMA              =,3X,1PE9.2,7H MHO/CM/19H LEWIS NUMBER      =,0PF10.MAI  300
     33)                                                                MAI  301
390   FORMAT (1H1,7X,28H- FROZEN THROAT CONDITIONS -)                   MAI  302
400   FORMAT (1H1,6X,33H- EQUILIBRIUM THROAT CONDITIONS -)              MAI  303
410   FORMAT (24H CASE EXECUTION TIME WAS,F6.2,8H MINUTES)              MAI  304
420   FORMAT (8H0RUN NO.,I7,5X,4HCASE,I4,23H OF THIS JOB COMPLETED.,I5.2MAI  305
     14H RECORDS WRITTEN ON TAPE,I3,1H.,I7,36H RECORDS WRITTEN SO FAR INMAI  306
     2 THIS JOB.)                                                       MAI  307
      END                                                               MAI  308
```

```
      BLOCK DATA
      LOGICAL DATAPE,WEDGEM,AXISYM,READXS,RE[illegible],AAMS,AXIMOD
      DOUBLE PRECISION AIN
      COMMON /ELEM/ EP1(2),EP2(2),EP3(2),EP4(2),EP5(2),EP6(2),EP7(2),
     1      EP8(2),EP9(2),EP10(2)
      COMMON /SPEC/ SP1(43),SP2(43),SP3(43),SP4(43),SP5(43),SP6(43),
     1      SP7(43),SP8(43),SP9(43),SP10(43),SP11(43),SP12(43),SP13(43),
     2      SP14(43),SP15(43),SP16(43),SP17(43),SP18(43),SP19(43),
     3      SP20(43),SP21(43),SP22(43),SP23(43),SP24(43),SP25(43),
     4      SP26(43),SP27(43),SP28(43),SP29(43),SP30(43)
      COMMON /REAC/ RP1(29),RP2(29),RP3(29),RP4(29),RP5(29),RP6(29),
     1      RP7(29),RP8(29),RP9(29),RP10(29),RP11(29),RP12(29),RP13(29),
     2      RP14(29),RP15(29),RP16(29),RP17(29),RP18(29),RP19(29),
     3      RP20(29),RP21(29),RP22(29),RP23(29),RP24(29),RP25(29),
     4      RP26(29),RP27(29),RP28(29),RP29(29),RP30(29),RP31(29),
     5      RP32(29),RP33(29),RP34(29),RP35(29),RP36(29),RP37(29),
     6      RP38(29),RP39(29),RP40(29),RP41(29),RP42(29),RP43(29),
     7      RP44(29),RP45(29),RP46(29),RP47(29),RP48(29),RP49(29),
     8      RP50(29),RP51(29),RP52(29),RP53(29),RP54(29),RP55(29),
     9      RP56(29),RP57(29),RP58(29),RP59(29),RP60(29),RP61(29),
     A      RP62(29),RP63(29),RP64(29),RP65(29),RP66(29),RP67(29),
     B      RP68(29),RP69(29),RP70(29),RP71(29),RP72(29),RP73(29),
     C      RP74(29),RP75(29),RP76(29),RP77(29),RP78(29),RP79(29),
     D      RP80(29),RP81(29),RP82(29),RP83(29),RP84(29),RP85(29),
     E      RP86(29),RP87(29),RP88(29),RP89(29),RP90(29),RP91(29),
     F      RP92(29)
      COMMON /MIXT/ GP1(124),GP2(124),GP3(124),GP4(124),GP5(124),
     1      GP6(124)
      COMMON /NOZZ/ ZP1(64),ZP2(64),ZP3(64),ZP4(64),ZP5(64),ZP6(64),
     1      ZP7(64),ZP8(64),ZP9(64),ZP10(64),ZP11(64),ZP12(64),ZP13(64),
     2      ZP14(64),ZP15(64),ZP16(64),ZP17(64),ZP18(64),ZP19(64),
     3      ZP20(64)
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),
     1      NSECTU(2),ISHAPE(12,2),NP[illegible]CTL(2),NPRFLS,NBL
      COMMON /MASSFL/ SMASS,CTMXXI,TSTJPI,IS(20)
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),
     1      SW,RO,JDIM,IPCINT
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,    CATFAC,TMODEL,XMODP1,DXMODP,
     1      TPLATE,KDIM
      COMMON /RDLIST/ IGAS,IGASO,NOZZO,ICHANO
      COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,MBL
      COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAPE
      COMMON /BLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOT(2),
     1      TAUW(2)
      COMMON /NEWMP/ FACMP,NMODPT
      COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,
     1      NWX,NANGLE,NRADLE,WEDGEM,AXISYM,ISW98
      COMMON /CHAN/ CP1(5),CP2(5),CP3(5),CP4(5),CP[illegible](5)
      COMMON /AVG/ WSAVE
      COMMON /TNE/ TN1(186),TN2(186)
      COMMON /RDMOD/ LEWIS,IAMBIP
      COMMON /READAT/ PRESAI,DELT1I,TPRNTI,DELTXI,CXMAXI,
     1      CTAPI,XZEROI,PARAMI(3,12,2),BZEROI,
     2      HSTAG,READXS,READG,AAMS,AXIMOD,
     3      ICASE,NEELS,ISW78,INT,ICHAN,NOSI,INEQVI,NREC0,JCS(10),
```

```
4      ISCI,ISSI,ISRI,ICI,IE(10),IR(64),ICTS(2,2),IEEP(10)
 COMMON /INGNE/ AIN(10,10)
 COMMON /RENE/ GAMIN,HTEST,TETEST,QTEST,DCHLL,DCHRAT,CCHI,TTEST,
1      GTEST
 DATA AIN /100*0./, GAMIN /1.E-10/
 DATA CTMXXI /5000./, DELT1I,TPRNTI,DELTXI /3*0.01/,CXMAXI /1.E5/,
2      BZEROI /0./, READXS,READG /2*.FALSE./, AAMS,AXIMOD/2*.TRUE./,
3      ICASE,NEELS,INT,ICHAN,INEQVI,NRECO,ISW7B /7*0/,
4      NQSI /4/
 DATA GAMIN /1.E-10/, HTEST /0.01/, TTEST /0.05/, GTEST /0.1/,
1      TETEST /0.05/, QTEST /0.1/, DCHLL /1.E-4/, DCHRAT /1.E-4/,
2      CCHI /0.1/
 DATA WSAVE /3.0/, LEWIS /1/
 DATA                    ANGLE /10*0./, RADLE /5*0./, NANGLE /0/,
1      NRADLE /0/, WX1,DWX /2*1./, WXI /20*1.E30/, NWX /0/,
2      TWEDGE /300./, WK /1.333/, ISW9B /0/
 DATA NMODPT /20/, IGAS /1/
 DATA THETA,SN,XSN,PRREF,HR,REPF,QWDOT,TAUW /13*0./
 DATA ITPOUT,DATAPE /8,.FALSE./
 DATA TSDIAM /20*1.E20/
 DATA FSTAG /1./,                CATFAC /1./, TMODEL /300./,
1      XMODP1 /1.E20/
 DATA TWALL,TPLATE /2*300./, IGASO,NOZZO,ICHANO /3*-11111/
 DATA KDIM /1/
 DATA JDIM /1/
 DATA CTMXXI /5000./, TSTOPI /0./
 DATA NPROFL /2*0/, NPRFLS /1/
 DATA EP1 /2HE-,5.48597E-4/                                          E-
 DATA EP3 /2HHE,4.0026/                                              HE
 DATA EP4 /1HC,12.011/                                               C
 DATA EP5 /1HN,14.007/                                               N
 DATA EP6 /1HO,16./                                                  O
 DATA EP7 /2HAR,39.948/                                              AR
 DATA SP1 /2HE-,1.1,1.1,2*0.,1.,2*0.,6*0.,0.,1.,-14.9276,0.,1.1,0.,  E-
1      2.,9*0.,10*0.,3*0./                                           E-
 DATA SP2 /1HN,1.1,5.1,0.,0.,1.,0.,0.,6*0.,112520.,1.,.2944,         EN
1      0.,5.1,0.,4.,10.,6.,12.,18.,5*0.,0.,54962.,82456.,238270.,   EN
2      250140.,5*0.,3*0./                                            EN
 DATA SP3 /1HO,1.1,6.1,0.,0.,1.0,0.,0.,6*0.,58990.,1.,.4938,0.,      O
1      7.1,0.,5.,3.,1.,5.,1.,8.,24.,3*0.,0.,453.,648.,45367.,       O
2      96616.,213990.,249820.,3*0.,3*0./                             O
 DATA SP4 /2HAR,1.1,7.1,0.,0.,1.,0.,0.,6*0.,0.,1.,1.8663,0.,1.1,0.,  AR
1      1.,9*0.,10*0.,3*0./                                           AR
 DATA SP5 /2HN2,1.1,5.1,0.,0.,2.,0.,0.,3.451483,3.088332E-4,         N2
1      -4.251428E-8,2.739295E-12,-5.468320E-17,3.071269,0.,2.,       N2
2      -.4106,3352.,5.1,1.1,1.,3.,6.,1.,2.,5*0.,0.,143540.,170480., N2
3      171500.,198110.,5*0.,3*0./                                    N2
 DATA SP6 /2HO2,1.1,6.1,0.,0.,2.,0.,0.,3.249473,4.963449E-4,         O2
1      -6.701753E-8,4.443339E-12,-1.000281E-16,5.915022,0.,2.,       O2
2      .1140.2239.,5.1,1.1,3.,2.,1.,3.,3.,5*0.,0.,22639.,37726.,    O2
3      103200.,142390.,5*0.,3*0./                                    O2
 DATA SP7 /2HNO,2.1,5.1,6.1,0.,1.0,1.0,0.,3.756216,2.083961E-4,      NO
1      -2.639548E-8,1.690332E-12,-3.611523E-17,3.611167,21460.,2.,   NO
2      .5455,2699.,7.1,1.1,2.,2.,2.,4.,2.,2.,2.,3*0.,0.,346.,       NO
3      125700.,131320.,149100.,151770.,173340.,3*0.,3*0./            NO
```

```
 DATA  SP8 /3HNO+,3,1,1,1,5,1,6,1,-1,,1,,1,,3,397385,3,749384E-4,      NO+
1      -6,062030E-8,4,637506E-12,-1,107704E-16,4,200563,236660,,      NO+
2      2,,,3841,3373,,4,1,1,1,1,,6,,3,,2,,6*0,,0,,106000,,160000,,   NO+
3      200000,,6*0,,3*0,/                                             NO+
 DATA  SP9 /2HN+,2,1,1,1,5,1,0,,-1,,1,,0,,6*0,,447600,,1,,,2943,      N+
1      0,,7,1,0,,1,,3,,5,,5,,1,,5,,15,,3*0,,0,,140,,375,,43789,,     N+
2      93456,,134860,,263740,,3*0,,3*0,/                              N+
 DATA  SP10 /2HO+,2,1,1,1,1,6,1,0,,-1,,1,,0,,6*0,,372940,,1,,,4938,0,, O+
1      3,1,0,,4,,10,,6,,7*0,,0,,76670,,115700,,7*0,,3*0,/             O+
 DATA  SP11 /3HN2+,2,1,1,1,5,1,0,,-1,,2,,0,,3,23806,4,47257E-4,       N2+
1      -3,95880E-8,1,52963E-12,-0,21145E-16,4,95160,357680,,          N2+
2      2,,-,3763,3124,,4,1,1,1,2,,4,,2,,2,,6*0,,0,,25890,,72797,,    N2+
3      184760,,6*0,,3*0,/                                             N2+
 DATA  SP12 /3HO2+,2,1,1,1,6,1,0,,-1,,2,,0,,3,49213,3,37873E-4,      O2+
1      -5,20841E-8,4,16207E-12,-0,97275E-16,4,66750,288000,,2,,       O2+
2      -,0317,2628,,5,1,1,1,2,,2,,8,,4,,4,,5*0,,0,,558,,91206,,      O2+
3      109760,,138860,,5*0,,3*0,/                                     O2+
 DATA  SP13 /3HCO2,2,1,4,1,6,1,0,,1,,2,,0,,6*0,,0,00000,3,,1,8958,   CO2
1      1977,,1,1,0,,1,,9*0,,10*0,,960,,960,,3380,/                    CO2
 DATA  SP14 /2HCO,2,1,4,1,6,1,0,,2*1,,0,,3,39468,3,22824E-4,          CO
1      -3,94364E-8,2,17519E-12,-0,42966E-16,4,204,                    CO
2      66770,,  2,,,3169,3083,,5,1,1,1,1,,6,,3,,6,,2,,5*0,,0,,       CO
3      139200,,159830,,178120,,186055,,5*0,,3*0,/                     CO
 DATA  SP15 /2HCN,2,1,4,1,5,1,0,,2*1,,0,,3,25545,4,33773E-4,          CN
1      -3,93346E-8,1,69712E-12,-0,23789E-16,5,6334,                   CN
2      197170,,2,,,2226,2939,,8,1,1,1,2,,4,,2,,4,,2,,4,,4,,4,,2*0,,  CN
3      0,,26069,,73759,,154263,,168570,,170640,,174230,,185220,,     CN
4      2*0,,3*0,/                                                     CN
 DATA  SP16 /2HHE,1,1,3,1,2*0,,1,,2*0,,6*0,,0,,1,,-1,5846,0,,1,1,0,,  HE
1      1,,9*0,,10*0,,3*0,/                                            HE
 DATA  SP17 /1HC,1,1,4,1,2*0,,1,,2*0,,6*0,,263550,,1,,,0637,0,,       C
1      10,1,0,,9,,5,,1,,5,,9,,3,,15,,34,,99,,401,,85,,29146,,61894,, C
2      96452,,172580,,177213,,183240,,200100,,214400,,243000,,3*0,/   C
 DATA  SP18 /2HC+,2,1,1,1,4,1,0,,-1,,1,,0,,6*0,,523310,,1,,,0636,0,,  C+
1      4,1,0,,2,,4,,12,,10,,6*0,,0,,183,,123040,,214240,,6*0,,3*0,/   C+
 DATA  SP19 /3HHE+,2,1,1,1,3,1,0,,-1,,1,,0,,6*0,,566840,,1,,-1,5846,  HE+
1      0,,1,1,0,,2,,9*0,,10*0,,3*0,/                                  HE+
 DATA  SP20 /3HAR+,2,1,1,1,7,1,0,,-1,,1,,0,,6*0,,363330,,1,,          AR+
1      1,8663,0,,2,1,0,,4,,2,,8*0,,0,,4094,,8*0,,3*0,/                AR+
 DATA  SP21 /4HHE3S,1,1,3,1,2*0,,1,,2*0,,6*0,,456910,,1,,-1,5846,0,,  HE3S
1      1,1,0,,3,,9*0,,10*0,,3*0,/                                     HE3S
 DATA  SP22 /4HHE1S,1,1,3,1,2*0,,1,,2*0,,6*0,,475260,,1,,-1,5846,0,,  HE1S
1      1,1,0,,1,,9*0,,10*0,,3*0,/                                     HE1S
 DATA  SP23 /4HHE2+,2,1,1,1,3,1,0,,-1,,2,,0,,6*0,,511720,,2,,         HE2+
1      -3,5619,2343,,  1,1,0,,2,,9*0,,10*0,,3*0,/                     HE2+
 DATA  SP24 /3HHE2,1,1,3,1,2*0,,2,,2*0,,6*0,,413650,,2,,-3,6287,      HE2
1      2492,,   1,1,0,,3,0,9*0,,10*0,,3*0,/                           HE2
 DATA  SP25/3HCO+,3,1,1,1,4,1,6,1,-1,,2*1,,3,49411,2,10083E-4,        CO+
1      -1,11714E-8,0,58582E-12,0,13605E-16,4,2967,                    CO+
2      389950,,2,,,2931,3142,,3,1,1,1,2,,4,,2,,7*0,,0,,59278,,       CO+
3      131166,,7*0,,3*0,/                                             CO+
 DATA  SP26 /4HAR*M,1,1,7,1,2*0,,1,,2*0,,6*0,,266350,,1,,1,8663,0,,  AR*M
1      1,1,0,,6,,9*0,,10*0,,3*0,/                                     AR*M
 DATA  SP27 /4HAR*R,1,1,7,1,2*0,,1,,2*0,,6*0,,267970,,1,,1,8663,0,,  AR*R
1      1,1,0,,6,,9*0,,10*0,,3*0,/                                     AR*R
```

```
      DATA SP28 /4HAR2+,2.1,1.1,7.1,0.,-1.,2.,0.,6*0.,337040.,2.,3.597,     AR2+
     1     115.,1.1,0.,2.,9*0.,10*0.,3*0./                                  AR2+
C     AIR-1 IS HIGH-TEMPERATURE AIR MODEL
      DATA GP1 /5HAIR-1,3.1,11.1,26.1,5.1,1.1,5.1,6.1,     7*0.,.78823,     AIR-1
     1     .21177,  8*0.,1.1,5.1,6.1,2.1,3.1,7.1,                          AIR-1
     2     8.1,9.1,10.1,11.1,12.1,9*0.,1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,    AIR-1
     3     9.1,10.1,11.1,12.1,13.1,14.1,15.1,16.1,17.1,18.1,19.1,20.1,     AIR-1
     4     21.1,22.1,23.1,24.1,25.1,26.1,38*0.,5.1,6.1,8*0.,2.1,3.1,       AIR-1
     5     5.1,0.,1.1/                                                     AIR-1
C     AIR-2 IS LOW-TEMP. MODEL WITH NO+ THE SOLE ION SPECIES.
      DATA GP2 /5HAIR-2,3.1,7.1,15.1,1.1,1.1,5.1,6.1,7*0.,.78823,          AIR-2
     1  .21177,8*0.,1.1,5.1,6.1,2.1,3.1,7.1,                               AIR-2
     2     8.1,13*0.,1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1,10.1,11.1,12.1,   AIR-2
     3     19.1,20.1,26.1,49*0.,5.1,6.1,8*0.,2.1,3.1,5.1,0.,1.1/           AIR-2
C     NONEQUILIBRIUM MODEL FOR ARGON                                       ARGON
      DATA GP3 /5HARGON,2.1,6.1,17.1,2.1,1.1,7.1,8*0.,1.,9*0.,1.1,4.1,    ARGON
     1     26.1,27.1,20.1,28.1,14*0.,76.1,77.1,78.1,79.1,80.1,81.1,82.1,   ARGON
     2     83.1,84.1,85.1,86.1,87.1,88.1,89.1,90.1,91.1,92.1,47*0.,        ARGON
     3     4.1,9*0.,1.1,20.1,4.1,2.1,2.1/                                  ARGON
C     NONEQUILIBRIUM MODEL FOR HELIUM                                       HELIUM
      DATA GP4 /6HHELIUM,2.1,7.1,19.1,2.1,1.1,3.1,8*0.,1.,9*0.,1.1,16.1,   HELIUM
     1     21.1,22.1,24.1,19.1,23.1,13*0.,35.1,36.1,37.1,38.1,39.1,40.1,   HELIUM
     2     41.1,42.1,43.1,44.1,45.1,46.1,47.1,48.1,49.1,50.1,51.1,52.1,    HELIUM
     3     53.1,45*0.,16.1,9*0.,1.1,19.1,16.1,1.1,2.1/                     HELIUM
C     PLANETARY ATMOSPHERE MODEL - MOLE FRACS 0.75 CO2, 0.2 AR, 0.05 N2    CONAR
      DATA GP5 /5HCONAR,5.1,19.1,48.1,8.1,1.1,4.1,5.1,6.1,7.1,5*0.,       CONAR
     1     0.75,0.2,0.05,7*0.,1.1,4.1,13.1,5.1,6.1,2.1,3.1,7.1,14.1,       CONAR
     2     15.1,17.1,8.1,9.1,10.1,11.1,12.1,18.1,20.1,25.1,0.,             CONAR
     3     1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1,10.1,11.1,12.1,13.1,14.1,   CONAR
     4     15.1,16.1,17.1,18.1,19.1,20.1,21.1,22.1,23.1,24.1,25.1,26.1,    CONAR
     5     54.1,55.1,56.1,57.1,58.1,59.1,60.1,61.1,62.1,63.1,64.1,65.1,    CONAR
     6     66.1,67.1,68.1,69.1,70.1,71.1,72.1,73.1,74.1,75.1,16*0.,        CONAR
     7     13.1,4.1,5.1,7*0.,3.1,3.1,14.1,0.,2.1/                          CONAR
C     PLANETARY ATMOSPHERE MODEL FOR TEMPERATURES TO 7000 DEG. K            CONAR2
      DATA GP6 /6HCONAR2,5.1,14.1,33.1,3.1,1.1,4.1,5.1,6.1,7.1,5*0.,      CONAR2
     1     .75,.2,.05,7*0.,1.1,4.1,13.1,5.1,6.1,2.1,3.1,7.1,14.1,15.1,    CONAR2
     2     17.1,8.1,18.1,25.1,6*0.,1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1,   CONAR2
     3     10.1,11.1,12.1,19.1,20.1,26.1,54.1,55.1,56.1,57.1,58.1,59.1,    CONAR2
     4     60.1,61.1,62.1,65.1,66.1,67.1,68.1,70.1,72.1,73.1,74.1,75.1,    CONAR2
     5     31*0.,13.1,4.1,5.1,7*0.,3.1,3.1,14.1,0.,2.1/                    CONAR2
      DATA RP1 /3.6E14,-1.,117980.,1.,1.,1.,6.,1.,2*0.,3.,1.,2*0.,1.,2*0.,REAC 001
     1     2.,2*0.,7.1.2.1.7.1.4.1.13.1.14.1.15.1.17.1.3*0./               REAC 001
      DATA RP2 /9.E15,-1.,117980.,0.,2.1.1.1.6.1.3.1.0.,3.1.2*0.,1.,1.,   REAC 002
     1     0.,3.,2*0.,0.,10*0./                                            REAC 002
      DATA RP3 /3.2E15,-1.,117980.,0.,1.1.2.1.6.1.2*0.,3.1.6.1.0.,2.,     REAC 003
     1     2*0.,2.,1.,0.,0.,10*0./                                         REAC 003
      DATA RP4 /7.2E14,-1.,117980.,0.,2.1.2.1.6.1.5.1.0.,3.1.5.1.0.,      REAC 004
     1     1.,1.,0.,2.,1.,0.,0.,10*0./                                     REAC 004
      DATA RP5 /1.9E15,-0.5,225040.,1.,1.1.1.1.5.1.2*0.,2.1.2*0.,1.,      REAC 005
     1     2*0.,2.,2*0.,8.1.3.1.7.1.6.1.4.1.13.1.14.1.15.1.17.1.2*0./      REAC 005
      DATA RP6 /4.1E16,-1.5,225040.,0.,2.1.1.1.5.1.2.1.0.,2.1.2*0.,1.,     REAC 006
     1     1.,0.,3.,2*0.,0.,10*0./                                         REAC 006
      DATA RP7 /4.7E15,-0.5,225040.,0.,1.1.2.1.5.1.2*0.,2.1.5.1.0.,2.,     REAC 007
     1     2*0.,2.,1.,0.,0.,10*0./                                         REAC 007
      DATA RP8 /3.9E14,-1.5,150050.,1.,1.1.2.1.7.1.2*0.,2.1.3.1.0.,1.,     REAC 008
```

```
1       2*0.,2*1.,0.,6.1,5.1,6.1,4.1,13.1,14.1,15.1,4*0./                    REAC 008
 DATA RP9 /7.8E14,-1.5,150050.,1.,1.1,1,2.1,7.1,2*0.,2.1,3.1,0.,1.,         REAC 009
1       2*0.,2*1.,0.,4.1,3.1,2.1,7.1,17.1,6*0./                              REAC 009
 DATA RP10 /3.2E13,1.,39150.,0.,2.1,2.1,3.1,7.1,0.,2.1,6.1,0.,1.,           REAC 010
1       1.,0.,1.,1.,0.,0.,10*0./                                             REAC 010
 DATA RP11 /7.E13,0.,75510.,0.,2.1,2.1,3.1,5.1,0.,2.1,7.1,0.,1.,            REAC 011
1       1.,0.,1.,1.,0.,0.,10*0./                                             REAC 011
 DATA RP12 /6.7E15,-1.5,0.,0.,2.1,2.1,8.1,1.1,0.,2.1,3.1,0.,1.,1.,          REAC 012
1       0.,1.,1.,0.,0.,10*0./                                                REAC 012
 DATA RP13 /2.2E22,-4.5,0.,0.,2.1,2.1,10.1,1.1,0.,3.1,1.1,0.,1.,            REAC 013
1       2.,0.,1.,1.,0.,0.,10*0./                                             REAC 013
 DATA RP14 /2.2E22,-4.5,0.,0.,2.1,2.1,9.1,1.1,0.,2.1,1.1,0.,1.,             REAC 014
1       2.,0.,1.,1.,0.,0.,10*0./                                             REAC 014
 DATA RP15 /8.E15,-1.5,0.,0.,2.1,1.1,12.1 1.1,0.,3.1,2*0.,1.,1.,0.,REAC 015
1       2.,2*0.,0.,10*0./                                                    REAC 015
 DATA RP16 /7.8E13,0.5,0.,0.,2.1,2.1,6.1,10.1,0.,3.1,12.1,0.,1.,             REAC 016
1       1.,0.,1.,1.,0.,0.,10*0./                                             REAC 016
 DATA RP17 /7.8E13,0.5,0.,0.,2.1,2.1,2.1 11.1,0.,5.1,9.1,0.,1.,1.,          REAC 017
1       0.,1.,1.,0.,0.,10*0./                                                REAC 017
 DATA RP18 /1.5E16,-1.5,0.,0.,2.1,1.1 11.1,1.1 0.,2.1,2*0.,1.,1.,           REAC 018
1       0.,2.,2*0.,0.,10*0./                                                 REAC 018
 DATA RP19 /1.E14,-2.5,0.,0.,3.1,2.1,7.1,8.1,1.1,5.1,6.1,0.,1.,1.,          REAC 019
1       1.,1.,1.,0.,0.,10*0./                                                REAC 019
 DATA RP20 /2.2E16,-2.5,0.,0.,3.1,2.1,8.1,1.1,5.1,7.1,5.1,0.,1.,            REAC 020
1       1.,1.,1.,0.,0.,10*0./                                                REAC 020
 DATA RP21 /1.5E13,0.,0.,0.,2.1,2.1,7.1,10,1,0.,3.1,8.1,0.,1.,1.,           REAC 021
1       0.,1.,1.,0.,0.,10*0./                                                REAC 021
 DATA RP22 /3.4E11,-2.,45700.,0.,2.1,2.1,5.1,10.1,0.,3.1,11.1,0.,           REAC 022
1       1.,1.,0.,1.,1,.0,.0.,10*0./                                          REAC 022
 DATA RP23 /4.8E14,0.,0.,0.,2.1,2.1,7.1,9.1,0.,2.1,8.1,0.,1.,1.,0.,REAC 023
1       1.,1.,0.,0.,10*0./                                                   REAC 023
 DATA RP24 /1.8E15,0.5,0.,0.,2.1,2.1,7.1,12.1,0.,6.1,8.1,0.,1.,1.,           REAC 024
1       0.,1.,1.,0.,0.,10*0./                                                REAC 024
 DATA RP25 /1.E14,0.,0.,0.,2.1,2.1,6.1,9.1,0.,3.1,8.1,0.,1.,1.,0.,          REAC 025
1       1.,1.,0.,0.,10*0./                                                   REAC 025
 DATA RP26 /8.8E16,-2.5,0.,0.,3.1,2.1,8.1,1.1,6.1,7.1,6.1,0.,1.,1.,REAC 026
1       1.,1.,1.,0.,0.,10*0./                                                REAC 026
 DATA RP35 /5.46E21,-4.3,0.,1.,2.1,1.1,1.1,19.1,0.,21.1,2*0.,2*1.,          REAC 035
1       0.,1.,2*0.,1.1.1.1,9*0./                                             REAC 035
 DATA RP36 /1.82E21,-4.3,0.,1.,2.1,1.1,1.1,19.1,0.,22.1,2*0.,2*1.,          REAC 036
1       0.,1.,2*0.,1,1,1.1,9*0./                                             REAC 036
 DATA RP37 /1.27E11,-0.81,0.,0.,2.1,1.1,1.1,19.1,0.,21.1,2*0.,1.,           REAC 037
1       1.,0.,1.,2*0.,0.,10*0./                                              REAC 037
 DATA RP38 /3.80E10,-0.85,0.,0.,2.1,1.1,1.1,19.1,0.,22.1,2*0.,2*1.,REAC 038
1       0.,1.,2*0.,0.,10*0./                                                 REAC 038
 DATA RP39 /3.92E16,2*0.,1.,2.1,1.1,16.1,19.1,0.,23.1,2*0.,2*1.,             REAC 039
1       0.,1.,2*0.,1.1.16.1,9*0./                                            REAC 039
 DATA RP40 /1.54E21,-4.3,0.,1.,2.1,1.1,1.1,23.1,0.,24.1,2*0.,1.,1.,REAC 040
1       0.,1.,2*0.,1.1.1.1,9*0./                                             REAC 040
 DATA RP41 /5.13E20,-4.3,0.,1.,2.1,1.1,1.1,23.1,0.,16.1,2*0.,2*1.,           REAC 041
1       0.,2.,2*0.,1.1.1.1,9*0./                                             REAC 041
 DATA RP42 /2.26E14,3*0.,2*2.1,1.1,23.1,0.,16.1,21.1,0.,2*1.,0.,             REAC 042
1       1.,1.,0.,0.,10*0./                                                   REAC 042
 DATA RP43 /7.5E13,3*0.,2*2.1,1.1,23.1,0.,16.1,22.1,0.,2*1.,0.,1.,           REAC 043
1       1.,0.,0.,10*0./                                                      REAC 043
```

```
 DATA RP44 /3.65E16,-0.5,0.,1.,2*1.1,22.1,2*0.,21.1,2*0.,1.,2*0.,  REAC 044
1      1.,2*0.,1.1.1.1,9*0./                                        REAC 044
 DATA RP45 /8.E14,-0.25,1272.,1.,2*1.1,21.1,2*0.,16.1,2*0.,1.,2*0.,REAC 045
1      1.,2*0.,1.1.1.1,9*0./                                        REAC 045
 DATA RP46 /8.E14,-0.25,1272.,1.,2*1.1,22.1,2*0.,16.1,2*0.,1.,2*0.,REAC 046
1      1.,2*0.,1.1.1.1,9*0./                                        REAC 046
 DATA RP47 /5.20E10,0.,1590.,1.,2*1.1,22.1,2*0.,16.1,2*0.,1.,2*0., REAC 047
1      1.,2*0.,1.1.16.1,9*0./                                       REAC 047
 DATA RP48 /1.87E15,0.167,0.,0.,1.1,3.1,21.1,2*0.,1.1,16.1,19.1,   REAC 048
1      2.,2*0.,3*1.,0.,10*0./                                       REAC 048
 DATA RP49 /5.05E15,0.167,2*0.,2.1,3.1,21.1,22.1,0.,1.1,16.1,      REAC 049
1      19.1,2*1.,0.,3*1.,0.,10*0./                                  REAC 049
 DATA RP50 /6.28E15,0.167,2*0.,1.1,3.1,22.1,2*0.,1.1,16.1,19.1,    REAC 050
1      2.,2*0.,3*1.,0.,10*0./                                       REAC 050
 DATA RP51 /5.2E14,0.5,0.,1.,2.1,1.1,16.1,21.1,0.,24.1,2*0.,2*1.,  REAC 051
1      0.,1.,2*0.,1.1,16.1,9*0./                                    REAC 051
 DATA RP52 /1.87E15,0.167,2*0.,1.1,3.1,24.1,2*0.,1.1,16.1,19.1,    REAC 052
1      2.,2*0.,1.,3.,1.,0.,10*0./                                   REAC 052
 DATA RP53 /8.E14,-0.25,1272.,1.,2*1.1,24.1,2*0.,16.1,2*0.,1.,2*0.,REAC 053
1      2.,2*0.,1.,1.1.1,9*0./                                       REAC 053
 DATA RP54 /4.48E15,-1.,256000.,1.,1.1,2.1,14.1,2*0.,17.1,3.1,0.,  REAC 054
1      1.,2*0.,2*1.,0.,9.1,2.1,3.1,4.1,5.1,6.1,13.1,14.1,15.1,17.1,REAC 054
2      0./                                                          REAC 054
 DATA RP55 /8.81E14,-2.,125600.,1.,1.1,2.1,13.1,2*0.,3.1,14.1,0.,  REAC 055
1      1.,2*0.,2*1.,0.,9.1,2.1,3.1,4.1,5.1,6.1,13.1,14.1,15.1,17.1,REAC 055
2      0./                                                          REAC 055
 DATA RP56 /2.33E11,.5,130500.,0.,1.1,2.1,14.1,2*0.,13.1,17.1,0.,  REAC 056
1      2.,2*0.,2*1.,0.,11*0./                                       REAC 056
 DATA RP57 /2.73E13,.5,138100.,0.,2.1,2,1,3.1,14.1,0.,6.1,17.1,0.,  REAC 057
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 057
 DATA RP58 /2.86E13,.5,106500.,0.,2.1,2.1,2.1,14.1,0.,7.1,17.1,0.,  REAC 058
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 058
 DATA RP59 /4.59E10,.5,23970.,0.,2.1,2.1,7.1,14.1,0.,13.,2.1,0.,    REAC 059
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 059
 DATA RP60 /2.55E11,.5,7606.,0.,2.1,2.1,13,1,3.1,0.,6.1,14.1,0.,    REAC 060
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 060
 DATA RP61 /1.07E14,.5,67050.,0.,2.1,2.1,14.1,25.1,0.,13.1,18.1,0.,REAC 061
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 061
 DATA RP62 /6.03E13,.5,63360.,0.,2.1,2.1,14.1,18.1,0.,17.1,25.1,0.,REAC 062
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 062
 DATA RP63 /6.66E14,.5,54160.,0.,2.1,2.1,3.1,18.1,0.,17.1,10.1,0.,  REAC 063
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 063
 DATA RP64 /1.09E14,.5,9222.,0.,2.1,2.1,14.1,10.1,0.,3.1,25.1,0.,   REAC 064
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 064
 DATA RP65 /5.47E14,.5,74700.,0.,2.1,2.1,3.1,25.1,0.,6.1,18.1,0.,   REAC 065
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 065
 DATA RP66 /1.5E16,-1.5,0.,0.,2.1,2.1,25.1,1.1,0.,17.1,3.1,0.,      REAC 066
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 066
 DATA RP67 /2.2E22,-4.5,0.,0.,2.1,2.1,18.1,1.1,0.,17.1,1.1,0.,      REAC 067
1      1.,2.,0.,2*1.,0.,11*0./                                      REAC 067
 DATA RP68 /4.59E14,.5,109500.,0.,2.1,2.1,14.1,8.1,0.,25.1,7.1,0.,  REAC 068
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 068
 DATA RP69 /4.53E14,.5,44490.,0.,2.1,2.1,14.1,12.1,0.,25.1,6.1,0.,  REAC 069
1      2*1.,0.,2*1.,0.,11*0./                                       REAC 069
 DATA RP70 /5.96E14,.5,3227.,0.,2.1,2.1,17.1,8.1,0.,25.1,2.1,0.,    REAC 070
```

```
1       2*1.,0.,2*1.,0.,11*0./                                                  REAC 070
 DATA RP71 /2.2E22,-4.5,0.,0.,2.1,2.1,20.1,1.1,1.0.,4.1,1.1.0.,                 REAC 071
1       1.,2.,0.,2*1.,0.,11*0./                                                 REAC 071
 DATA RP72 /3.6E15,-1.,131160.,1.,1.1,2.1,15.1,2*0.,17.1,2.1,0.,                REAC 072
1       1.,2*0.,2*1.,0.,10.1,2.1,3.1,4.1,5.1,6.1,7.1,13.1,14.1,15.1,           REAC 072
2       17.1/                                                                   REAC 072
 DATA RP73 /4.3E14,.5,69670.,0.,2.1,2.1,14.1,2.1,0.,15.1,3.1,0.,                REAC 073
1       2*1.,0.,2*1.,0.,11*0./                                                  REAC 073
 DATA RP74 /1.5E13,.5,51670.,0.,2.1,2.1,5.1,17.1,0.,15.1,2.1,0.,                REAC 074
1       2*1.,0.,2*1.,0.,11*0./                                                  REAC 074
 DATA RP75 /2.E14,1.,55640.,0.,2.1,2.1,15.1,3.1,0.,17.1,7.1,0.,                 REAC 075
1       2*1.,0.,2*1.,0.,11*0./                                                  REAC 075
 DATA RP76 /3.64E21,-4.3,0.,0.,2.1,2.1,1.1,20.1,0.,1.1,26.1,0.,                 REAC 076
1       2.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 076
 DATA RP77 /3.64E21,-4.3,0.,0.,2.1,2.1,1.1,20.1,0.,1.1,27.1,0.,                 REAC 077
1       2.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 077
 DATA RP78 /8.22E10,-0.81,0.,0.,2.1,1.1,1.1,20.1,0.,26.1,2*0.,                  REAC 078
1       1.,1.,0.,1.,2*0.,0.,10*0./                                              REAC 078
 DATA RP79 /8.22E10,-0.81,0.,0.,2.1,1.1,1.1,20.1,0.,27.1,2*0.,                  REAC 079
1       1.,1.,0.,1.,2*0.,0.,10*0./                                              REAC 079
 DATA RP80 /6.E10,-0.5,0.,0.,2.1,1.1,1.1,20.1,0.,4.1,2*0.,                      REAC 080
1       1.,1.,0.,1.,2*0.,0.,10*0./                                              REAC 080
 DATA RP81 /5.E14,0.5,0.,0.,2.1,2.1,1.1,26.1,0.,1.1,4.1,0.,                     REAC 081
1       1.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 081
 DATA RP82 /7.2E13,0.5,0.,0.,2.1,2.1,1.1,27.1,0.,1.1,4.1,0.,                    REAC 082
1       1.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 082
 DATA RP83 /8.E4,0.,0.,0.,1.1,1.1,27.1,2*0.,4.1,2*0.,                           REAC 083
1       1.,2*0.,1.,2*0.,0.,10*0./                                               REAC 083
 DATA RP84 /1.E17,-0.5,0.,0.,2.1,2.1,1.1,27.1,0.,1.1,26.1,0.,                   REAC 084
1       1.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 084
 DATA RP85 /3.5E9,0.5,0.,0.,2.1,1.1,4.1,26.1,0.,4.1,2*0.,                       REAC 085
1       1.,1.,0.,2.,2*0.,0.,10*0./                                              REAC 085
 DATA RP86 /8.7E14,-0.56,0.,0.,2.1,1.1,4.1,26.1,0.,4.1,2*0.,                    REAC 086
1       2.,1.,0.,3.,2*0.,0.,10*0./                                              REAC 086
 DATA RP87 /3.5E9,0.5,0.,0.,2.1,1.1,4.1,27.1,0.,4.1,2*0.,                       REAC 087
1       1.,1.,0.,2.,2*0.,0.,10*0./                                              REAC 087
 DATA RP88 /8.7E14,-0.56,0.,0.,2.1,1.1,4.1,27.1,0.,4.1,2*0.,                    REAC 088
1       2.,1.,0.,3.,2*0.,0.,10*0./                                              REAC 088
 DATA RP89 /5.2E15,-0.75,0.,0.,2.1,2.1,4.1,20.1,0.,4.1,28.1,0.,                 REAC 089
1       2.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 089
 DATA RP90 /2.8E16,-0.67,1252.,0.,2.1,2.1,1.1,28.1,0.,4.1,26.1,0.,              REAC 090
1       1.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 090
 DATA RP91 /2.8E16,-0.67,1252.,0.,2.1,2.1,1.1,28.1,0.,4.1,27.1,0.,              REAC 091
1       1.,1.,0.,1.,1.,0.,0.,10*0./                                             REAC 091
 DATA RP92 /2.E21,-4.3,0.,0.,2.1,2.1,1.1,28.1,0.,1.1,4.1,0.,                    REAC 092
1       2.,1.,0.,1.,2.,0.,0.,10*0./                                             REAC 092
 DATA TN1 /4*2.1,1.1,7*2.1,6*1.1,2.1,6*0.,                                      TNE/HE
1       2*2.1,2*0.,1.1,2*2.1,2*1.1,3*2.1,0.,3*2.1,1.1,2*2.1,6*0.,              TNE/HE
2       2*5.1,2*2.1,3.1,2*5.1,2*4.1,3*5.1,6.1,3*5.1,3.1,2*5.1,6*0.,            TNE/HE
3       109890.,91540.,109890.,91540.,0.,98040.,458270.,2*0.,                  TNE/HE
4       18350.,456730.,2*475080.,346840.,365190.,383540.,0.,                   TNE/HE
5       260350.,413480.,6*0.,                                                   TNE/HE
6       25*0.,
7       0.,100.,200.,400.,600.,800.,1000.,1500.,2000.,3000.,4000.,             TNE/HE
8       5000.,6000.,7000.,8000.,9000.,10000.,12000.,14000.,16000.,             TNE/HE
```

```
9      18000.,20000.,25000.,30000.,35000.,40000.,45000.,50000.,      TNE/HE
A      70000.,100000.,                                               TNE/HE
B      0.,5.E-16,5.59E-16,5.83E-16,5.99E-16,6.11E-16,6.21E-16,       TNE/HE
C      6.39E-16,6.51E-16,6.67E-16,6.76E-16,6.81E-16,6.84E-16,        TNE/HE
D      6.84E-16,6.83E-16,6.80E-16,6.77E-16,6.67E-16,6.55E-16,        TNE/HE
E      6.42E-16,6.29E-16,6.15E-16,5.80E-16,5.46E-16,5.14E-16,        TNE/HE
F      4.85E-16 4.60E-16,4.32E-16,3.4E-16,2.4E-16,                   TNE/HE
G      0.4/                                                          TNE/HE
 DATA TN2 /4*2.1,4.1,2*2.1,5.1,2.1,5*1.1,2*3.1,2.1,8*0.,             TNE/AR
1      2*2.1,3*0.,2*2.1,0.,2.1,1.1,0.,1.1,0.,1.1,3*2.1,8*0.,         TNE/AR
2      2*5.1,3*1.1,2*5.1,6.1,5.1,3.1,6.1,3.1,6.1,3.1,3*5.1,8*0.,     TNE/AR
3      96970.,95360.,96970.,95360.,363330.,266350.,267970.,267970.,  TNE/AR
4      1600.,0.,226000.,0.,226000.,0.,70680.,69070.,337040.,8*0.,    TNE/AR
5      2*0.,2*0.7,1.,20*0.,                                          TNE/AR
5      0.,25.,50.,100.,200.,300.,400.,600.,800.,900.,1000.,1200.,    TNE/AR
6      1400.,1600.,2000.,2500.,4000.,6000.,8000.,10000.,12000.,      TNE/AR
7      15000.,20000.,25000.,30000.,35000.,40000.,45000.,50000.,      TNE/AR
8      100000.,                                                      TNE/AR
9      10.E-16,10.E-16,6.E-16,3.57E-16,2.12E-16,1.39E-16,0.97E-16,   TNE/AR
A      0.57E-16,0.40E-16,0.35E-16,0.32E-16,0.29E-16,0.29E-16,        TNE/AR
B      0.31E-16,0.38E-16,0.51E-16,0.98E-16,1.69E-16,2.40E-16,        TNE/AR
C      3.08E-16,3.73E-16,4.65E-16,6.02E-16,7.19E-16,8.11E-16,        TNE/AR
D      8.81E-16,9.30E-16,9.63E-16,9.82E-16,10.E-16,                  TNE/AR
E      2.E7/                                                         TNE/AR
C     DCA 0.75-INCH NOZZLE                                           NOZZ  1
      DATA ZP 1 /   0.95250,   -5.53974,    6.1,    2.1,            NOZZ  1
1        1.1, 2.1, 1.1, 3.1, 1.1, 2.1, 2.1, 1.1, 0.1, 0.1, 0.1, 0.1, NOZZ  1
2  -5.568490, -5.455796, -1.523172, -1.264826, -1.099852,  0.0     , NOZZ  1
3   0.329700,  0.0     ,  0.0     ,  0.0     ,  0.0     ,           NOZZ  1
4-12.978253, -2.747477,  0.0     ,  2.364483, -5.449149,  0.127000, NOZZ  1
5   1.951731, -0.052408,  0.0     ,  1.714493, -1.539789,  0.317500, NOZZ  1
6  -0.317500, -1.732051,  0.0     ,  2.222500,  0.0     ,  1.270000, NOZZ  1
7   2.222500,  0.0     ,  1.270000,  0.907699,  0.267949,  0.0     , NOZZ  1
8   0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     , NOZZ  1
9   0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     , NOZZ  1
A        4HDCA /                                                     NOZZ  1
C     DCA 1.5-INCH NOZZLE                                            NOZZ  2
      DATA ZP 2 /   1.90500,   -9.14400,    4.1,    2.1,            NOZZ  2
1        1.1, 2.1, 1.1, 2.1, 2.1, 1.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, NOZZ  2
2  -9.431499, -8.304556, -0.006647,  0.0     ,  0.032870,  0.0     , NOZZ  2
3   0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     ,           NOZZ  2
4-22.738861, -2.747476,  0.0     ,  3.608504, -8.238090,  1.270000, NOZZ  2
5   1.904823, -0.052408,  0.0     ,  2.031999,  0.0     ,  0.127000, NOZZ  2
6   2.031999,  0.0     ,  0.127000,  1.900517,  0.267949,  0.0     , NOZZ  2
7   0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     , NOZZ  2
8   0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     , NOZZ  2
9   0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     ,  0.0     , NOZZ  2
A        4HDCA /                                                     NOZZ  2
C     MRA 2.25-INCH NOZZLE                                           NOZZ  3
      DATA ZP 3 /   2.85750,   -2.71780,    6.1,    2.1,            NOZZ  3
1        1.1, 3.1, 1.1, 2.1, 1.1, 2.1, 2.1, 1.1, 0.1, 0.1, 0.1, 0.1, NOZZ  3
2  -2.658384, -2.447308, -1.127258, -0.610565, -0.006647,  0.0     , NOZZ  3
3   0.032870,  0.0     ,  0.0     ,  0.0     ,  0.0     ,           NOZZ  3
4   5.171379, -0.176327,  0.0     ,  5.339946, -2.711272,  0.304800, NOZZ  3
5   1.253495, -1.732051,  0.0     ,  3.523453, -0.577332,  0.635000, NOZZ  3
```

```
      6  2.857325, -0.052408,   0.0      ,  2.984499,  0.0      ,  0.127000, NOZZ   3
      7  2.984499,  0.0      ,  0.127000,  2.853019,  0.267949,  0.0      , NOZZ   3
      8  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   3
      9  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   3
      A       4HMRA /                                                          NOZZ   3
C      MRA 1.00-INCH NOZZLE                                                    NOZZ   4
      DATA ZP 4 /   1.27000, -3.58140,   6.1,   2.1,                           NOZZ   4
      1       1.1, 3.1, 1.1, 2.1, 1.1, 2.1, 2.1, 1.1, 0.1, 0.1, 0.1, 0.1,    NOZZ   4
      2 -3.528553, -3.308723, -1.075787, -0.559095, -0.006647,  0.0      ,    NOZZ   4
      3  0.032870,  0.0      ,  0.0      ,  0.0      ,  0.0      ,           NOZZ   4
      4  5.015062, -0.176327,  0.0      ,  5.324564, -3.583686,  0.317500,    NOZZ   4
      5 -0.247563, -1.732051,  0.0      ,  1.933256, -0.525861,  0.635000,    NOZZ   4
      6  1.269825, -0.052408,  0.0      ,  1.397000,  0.0      ,  0.127000,    NOZZ   4
      7  1.397000,  0.0      ,  0.127000,  1.265519,  0.267949,  0.0      ,    NOZZ   4
      8  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   4
      9  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   4
      A       4HMRA /                                                          NOZZ   4
C      EOS 0.32-INCH NOZZLE                                                    NOZZ   5
      DATA ZP 5 /   0.40640, -4.82600,   6.1,   4.1,                           NOZZ   5
      1       1.1, 3.1, 1.1, 3.1, 1.1, 2.1, 2.1, 1.1, 3.1, 1.1, 0.1, 0.1,    NOZZ   5
      2 -4.836277, -4.806674, -1.720592, -1.718576, -0.053673,  0.0      ,    NOZZ   5
      3  0.058642,  1.134806,  1.141119,  0.0      ,  0.0      ,           NOZZ   5
      4  1.725971, -0.176327,  0.0      ,  2.453567, -4.858330,  0.127000,    NOZZ   5
      5  0.429518, -0.445229,  0.0      ,  1.079555, -1.772248,  0.127000,    NOZZ   5
      6  0.393271, -0.466308,  0.0      ,  0.533400,  0.0      ,  0.127000,    NOZZ   5
      7  0.533400,  0.0      ,  0.127000,  0.390222,  0.520567,  0.0      ,    NOZZ   5
      8  0.868315,  1.193448,  0.127000,  0.467999,  0.452217,  0.0      ,    NOZZ   5
      9  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   5
      A       4HEOS /                                                          NOZZ   5
C      EOS 0.775-INCH NOZZLE                                                   NOZZ   6
      DATA ZP 6 /   0.98171, -5.96900,   8.1,   2.1,                           NOZZ   6
      1       1.1, 3.1, 1.1, 3.1, 1.1, 2.1, 1.1, 2.1, 2.1, 1.1, 0.1, 0.1,    NOZZ   6
      2 -5.974197, -5.944594, -2.863580, -2.851736, -2.792465, -2.735612,    NOZZ   6
      3 -0.006647,  0.0      ,  0.052330,  0.0      ,  0.0      ,           NOZZ   6
      4  1.509393, -0.176327,  0.0      ,  2.437734, -5.996250,  0.127000,    NOZZ   6
      5 -0.092950, -0.445229,  0.0      ,  1.065978, -2.915236,  0.127000,    NOZZ   6
      6 -0.470487, -0.577350,  0.0      ,  1.251729, -2.728965,  0.127000,    NOZZ   6
      7  0.981535, -0.052408,  0.0      ,  1.108710,  0.0      ,  0.127000,    NOZZ   6
      8  1.108710,  0.0      ,  0.127000,  0.969327,  0.452217,  0.0      ,    NOZZ   6
      9  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   6
      A       4HEOS /                                                          NOZZ   6
C      MRA 0.75-INCH NOZZLE                                                    NOZZ   7
      DATA ZP 7 /   0.95250, -3.81000,   6.1,   2.1,                           NOZZ   7
      1       1.1, 3.1, 1.1, 2.1, 1.1, 2.1, 2.1, 1.1, 0.1, 0.1, 0.1, 0.1,    NOZZ   7
      2 -3.624795, -3.413759, -0.999293, -0.482600, -0.006647,  0.0      ,    NOZZ   7
      3  0.032870,  0.0      ,  0.0      ,  0.0      ,  0.0      ,           NOZZ   7
      4  4.984842, -0.176327,  0.0      ,  5.323822, -3.677723,  0.304800,    NOZZ   7
      5 -0.436579, -1.732050,  0.0      ,  1.511746, -0.449367,  0.635000,    NOZZ   7
      6  0.952325, -0.052408,  0.0      ,  1.079499,  0.0      ,  0.127000,    NOZZ   7
      7  1.079499,  0.0      ,  0.127000,  0.948019,  0.267949,  0.0      ,    NOZZ   7
      8  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   7
      9  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ   7
      A       4HMRA /                                                          NOZZ   7
C      MRA 1.5-INCH NOZZLE                                                     NOZZ   8
      DATA ZP 8 /   1.90500, -3.17500,   6.1,   2.1,                           NOZZ   8
```

```
1         1.1,  3.1,  1.1,  2.1,  1.1,  2.1,  2.1,  1.1,  0.1,  0.1,  0.1,  0.1,  NOZZ   8
2 -3.157434,  -2.946399,  -1.075492,  -0.558800,  -0.006647,   0.0      ,  NOZZ   8
3   0.032870,   0.0      ,   0.0      ,   0.0      ,   0.0      ,                  NOZZ   8
4   5.082271,  -0.176327,   0.0      ,   5.338842,  -3.210362,   0.304800,  NOZZ   8
5   0.387933,  -1.732050,   0.0      ,   2.568238,  -0.525567,   0.635000,  NOZZ   8
6   1.904823,  -0.052408,   0.0      ,   2.031999,   0.0      ,   0.127000,  NOZZ   8
7   2.031999,   0.0      ,   0.127000,   1.900517,   0.267949,   0.0      ,  NOZZ   8
8   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ   8
9   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ   8
A         4HMRA /                                                                 NOZZ   8
C       10MW 2.25-INCH NOZZLE                                                     NOZZ   9
        DATA ZP 9 /    2.85750,  -8.00100,    6.1,    2.1,                       NOZZ   9
1         1.1,  3.1,  1.1,  2.1,  1.1,  2.1,  2.1,  1.1,  0.1,  0.1,  0.1,  0.1,  NOZZ   9
2 -7.918200,  -7.698371,  -7.446036,  -5.714999,  -0.013283,   0.0      ,  NOZZ   9
3   0.039444,   0.0      ,   0.0      ,   0.0      ,   0.0      ,                  NOZZ   9
4   3.654496,  -0.176327,   0.0      ,   4.738010,  -7.973334,   0.317500,  NOZZ   9
5  -3.437203,  -1.732050,   0.0      ,   5.570952,  -5.521296,   2.222500,  NOZZ   9
6   2.856915,  -0.087489,   0.0      ,   3.009897,   0.0      ,   0.152400,  NOZZ   9
7   3.009897,   0.0      ,   0.152400,   2.852120,   0.267949,   0.0      ,  NOZZ   9
8   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ   9
9   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ   9
A         4H10MW/                                                                 NOZZ   9
C       EOS  1.09-INCH NOZZLE                                                     NOZZ  10
        DATA ZP10 /    1.38176,  -3.55600,    8.1,    2.1,                       NOZZ  10
1         1.1,  3.1,  1.1,  2.1,  1.1,  2.1,  1.1,  2.1,  2.1,  1.1,  0.1,  0.1,  NOZZ  10
2 -3.540351,  -3.473112,  -2.430964,  -2.393882,  -1.280036,  -1.259742,  NOZZ  10
3 -0.005397,   0.0      ,   0.006574,   0.0      ,   0.0      ,                  NOZZ  10
4   1.921388,  -0.176327,   0.0      ,   2.295506,  -3.584458,   0.254000,  NOZZ  10
5   0.829850,  -0.487733,   0.0      ,   2.243805,  -2.319619,   0.254000,  NOZZ  10
6   1.269022,  -0.305731,   0.0      ,   1.903271,  -1.205774,   0.254000,  NOZZ  10
7   1.381163,  -0.217432,   0.0      ,   1.407157,   0.0      ,   0.025400,  NOZZ  10
8   1.407157,   0.0      ,   0.025400,   1.380859,   0.267949,   0.0      ,  NOZZ  10
9   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ  10
A         4HEOS /                                                                 NOZZ  10
C         DCA   T12 CHANNEL, PROFILE 1                                            NOZZ  11
        DATA ZP11 /    2.54000,  -7.62000,    4.1,    2.1,                       NOZZ  11
1         1.1,  2.1,  1.1,  2.1,  2.1,  1.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  NOZZ  11
2 -6.730998,  -6.095999,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ  11
3   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,                  NOZZ  11
4  -1.175995,  -0.577350,   0.0      ,   3.809999,  -6.095999,   1.270000,  NOZZ  11
5   2.539999,   0.0      ,   0.0      , 105.079999,   0.0      , 102.540000,  NOZZ  11
6 105.079999,   0.0      , 102.540000,   2.539999,   0.0      ,   0.0      ,  NOZZ  11
7   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ  11
8   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ  11
9   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ  11
A         4H DCA/                                                                 NOZZ  11
C         DCA   T12 CHANNEL, PROFILE 2                                            NOZZ  12
        DATA ZP12 /    1.27000,  -7.62000,    2.1,    2.1,                       NOZZ  12
1         1.1,  2.1,  2.1,  1.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  NOZZ  12
2 -6.771664,   0.0      ,   2.646396,   0.0      ,   0.0      ,   0.0      ,  NOZZ  12
3   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,                  NOZZ  12
4  -0.501576,  -0.495987,   0.0      ,  16.509995,   0.0      ,  15.240000,  NOZZ  12
5  16.509995,   0.0      ,  15.240000,   1.034893,   0.176327,   0.0      ,  NOZZ  12
6   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ  12
7   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,   0.0      ,  NOZZ  12
```

```
     8  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ 12
     9  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ 12
     A        4H DCA/                                                                      NOZZ 12
C      DCA  T22 CHANNEL, PROFILE 2                                                        NOZZ 13
      DATA ZP13 /    2.54000,  -7.62000,    2.1,    2.1,                                  NOZZ 13
     1         1.1,  2.1,  2.1,  1.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  NOZZ 13
     2 -6.771664,  0.0      ,  2.646396,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ 13
     3  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,                       NOZZ 13
     4  0.768428, -0.495987,  0.0      , 17.779999,  0.0      , 15.240000,              NOZZ 13
     5 17.779999,  0.0      , 15.240000,  2.304897,  0.176327,  0.0      ,  0.0      , NOZZ 13
     6  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ 13
     7  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ 13
     8  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ 13
     9  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      ,  0.0      , NOZZ 13
     A        4H DCA/                                                                      NOZZ 13
C      T12 CHANNEL FOR DCA                                                                CHAN 1
      DATA CP1 /11.1,12.1,3HT12,1.1,3HDCA/                                               CHAN 1
C      T22 CHANNEL FOR 10MW                                                               CHAN 2
      DATA CP2 /11.1,13.1,3HT22,1.1,4H10MW/                                              CHAN 2
      END
```

```
      BLOCK DATA                                                        BD2    1
      COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,I(50)            BD2    2
      COMMON /TRANV/ VV1 (5),VV2 (5),VV3 (5),VV4 (5),VV5 (5),          BD2    3
     1               VV6 (5),VV7 (5),VV8 (5),VV9 (5),VV10(5),          BD2    4
     2               VV11(5),VV12(5),VV13(5),VV14(5),VV15(5),          BD2    5
     3               VV 6(5),VV17(5),VV18(5),VV19(5),VV20(5),          BD2    6
     4               VV21(5),VV22(5),VV23(5),VV24(5),VV25(5),          BD2    7
     5               VV26(5),VV27(5),VV28(5),VV29(5),VV30(5),          BD2    8
     6               VV31(5),VV32(5),VV33(5),VV34(5),VV35(5),          BD2    9
     7               VV36(5),VV37(5),VV38(5),VV39(5),VV40(5),          BD2   10
     8               VV41(5),VV42(5),VV43(5),VV44(5),VV45(5),          BD2   11
     9               VV46(5),VV47(5),VV48(5),VV49(5),VV50(5),          BD2   12
     $               VV51(5),VV52(5),VV53(5),VV54(5),VV55(5),          BD2   13
     $               VV56(5),VV57(5),VV58(5),VV59(5),VV60(5),          BD2   14
     $               VV61(5),VV62(5),VV63(5),VV64(5),VV65(5),          BD2   15
     $               VV66(5),VV67(5),VV68(5),VV69(5),VV70(5),          BD2   16
     $               VV71(5),VV72(5),VV73(5),VV74(5),VV75(5),          BD2   17
     $               VV76(5),VV77(5),VV78(5),VV79(5),VV80(5),          BD2   18
     $               VV81(5),VV82(5),VV83(5),VV84(5),VV85(5),          BD2   19
     $               VV86(5),VV87(5),VV88(5),VV89(5),VV90(5),          BD2   20
     $               VV91(5),VV92(5),VV93(5),VV94(5),VV95(5),          BD2   21
     $               VV96(5),VV97(5),VV98(5),VV99(5),VV100(5)          BD2   22
      COMMON /TRANJ/ JJ1 (5),JJ2 (5),JJ3 (5),JJ4 (5),JJ5 (5),          BD2   23
     1               JJ6 (5),JJ7 (5),JJ8 (5),JJ9 (5),JJ10(5),          BD2   24
     2               JJ11(5),JJ12(5),JJ13(5),JJ14(5),JJ15(5),          BD2   25
     3               JJ16(5),JJ17(5),JJ18(5),JJ19(5),JJ20(5),          BD2   26
     4               JJ21(5),JJ22(5),JJ23(5),JJ24(5),JJ25(5),          BD2   27
     5               JJ26(5),JJ27(5),JJ28(5),JJ29(5),JJ30(5),          BD2   28
     6               JJ31(5),JJ32(5),JJ33(5),JJ34(5),JJ35(5),          BD2   29
     7               JJ36(5),JJ37(5),JJ38(5),JJ39(5),JJ40(5),          BD2   30
     8               JJ41(5),JJ42(5),JJ43(5),JJ44(5),JJ45(5),          BD2   31
     9               JJ46(5),JJ47(5),JJ48(5),JJ49(5),JJ50(5),          BD2   32
     $               JJ51(5),JJ52(5),JJ53(5),JJ54(5),JJ55(5),          BD2   33
     $               JJ56(5),JJ57(5),JJ58(5),JJ59(5),JJ60(5),          BD2   34
     $               JJ61(5),JJ62(5),JJ63(5),JJ64(5),JJ65(5),          BD2   35
     $               JJ66(5),JJ67(5),JJ68(5),JJ69(5),JJ70(5),          BD2   36
     $               JJ71(5),JJ72(5),JJ73(5),JJ74(5),JJ75(5),          BD2   37
     $               JJ76(5),JJ77(5),JJ78(5),JJ79(5),JJ80(5),          BD2   38
     $               JJ81(5),JJ82(5),JJ83(5),JJ84(5),JJ85(5),          BD2   39
     $               JJ86(5),JJ87(5),JJ88(5),JJ89(5),JJ90(5),          BD2   40
     $               JJ91(5),JJ92(5),JJ93(5),JJ94(5),JJ95(5),          BD2   41
     $               JJ96(5),JJ97(5),JJ98(5),JJ99(5),JJ100(5)          BD2   42
      COMMON /TRANI/ II1 (5),II2 (5),II3 (5),II4 (5),II5 (5),          BD2   43
     1               II6 (5),II7 (5),II8 (5),II9 (5),II10(5),          BD2   44
     2               II11(5),II12(5),II13(5),II14(5),II15(5),          BD2   45
     3               II16(5),II17(5),II18(5),II19(5),II20(5),          BD2   46
     4               II21(5),II22(5),II23(5),II24(5),II25(5),          BD2   47
     5               II26(5),II27(5),II28(5),II29(5),II30(5),          BD2   48
     6               II31(5),II32(5),II33(5),II34(5),II35(5),          BD2   49
     7               II36(5),II37(5),II38(5),II39(5),II40(5),          BD2   50
     8               II41(5),II42(5),II43(5),II44(5),II45(5),          BD2   51
     9               II46(5),II47(5),II48(5),II49(5),II50(5),          BD2   52
     $               II51(5),II52(5),II53(5),II54(5),II55(5),          BD2   53
     $               II56(5),II57(5),II58(5),II59(5),II60(5),          BD2   54
     $               II61(5),II62(5),II63(5),II64(5),II65(5),          BD2   55
```

```
$                II66(5),II67(5),II68(5),II69(5),II70(5),                BD2   56
$                II71(5),II72(5),II73(5),II74(5),II75(5),                BD2   57
$                II76(5),II77(5),II78(5),II79(5),II80(5),                BD2   58
$                II81(5),II82(5),II83(5),II84(5),II85(5),                BD2   59
                 II86(5),II87(5),II88(5),II89(5),II90(5),                BD2   60
$                II91(5),II92(5),II93(5),II94(5),II95(5),                BD2   61
$                II96(5),II97(5),II98(5),II99(5),II100(5)                BD2   62
 COMMON /TRANS4/                                                         BD2   63
1 TL 1(100), TL 2(100), TL 3(100), TL 4(100), TL 5(100), TL 6(100),BD2   64
2 TL 7(100), TL 8(100), TL 9(100), TL10(100), TL11(100), TL12(100),BD2   65
3 TL13(100), TL14(100), TL15(100), TL16(100), TL17(100), TL18(100),BD2   66
4 TL19(100), TL20(100), TL21(100), TL22(100), TL23(100), TL24(100),BD2   67
5 TL25(100), TL26(100), TL27(100), TL28(100), TL29(100), TL30(100),BD2   68
6 TL31(100), TL32(100), TL33(100), TL34(100), TL35(100), TL36(100),BD2   69
7 TL37(100), TL38(100), TL39(100), TL40(100)                             BD2   70
 COMMON /TRANS8/ NV(37),N                                                BD2   71
 COMMON /RDTR/ ISW8B                                                     BD2   72
 DATA ISW8B /0/                                                          BD2   73
 DATA NV/3.2.3.4.3.2.0.3.5.0.2.4.3.3.0.                                  BD2   74
 DATA NNKQ/33/                                                           BD2   75
 DATA ISEQ/ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16.       BD2   76
1 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32.        BD2   77
2 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48.        BD2   78
3 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 37*0/BD2   79
 DATA KKQ/ 3. 3. 3. 3. 6.14.14.14. 5. 6. 6. 5. 5. 5. 5. 5. 5. 5. 5.BD2   80
1 5. 5. 5. 5. 9.14.14.14.14.14.14.14.14.14.14.14.14.14.14. 9. 5.BD2      81
2 5. 5. 5. 5. 5. 5. 5.13.13. 3. 3. 3. 3. 5.10. 4. 4. 6. 4. 4. 4. 4.BD2  82
3 37*0/                                                                  BD2   83
 DATA NNQ/ 3. 1. 1. 1. 3. 1. 1. 1. 1. 1. 1. 1. 3. 1. 5. 5. 5. 4. 5.BD2   84
1 2. 1. 4. 4. 2. 2. 2. 1. 1. 1. 1. 2. 2. 1. 1. 1. 1. 2. 2. 5. 2. 1.BD2   85
2 1. 2. 2. 1. 1. 1. 1. 4. 5. 5. 2. 5. 3. 4. 0. 5. 4. 2. 4. 2. 4. 1.BD2   86
3 37*0/                                                                  BD2   87
 DATA II  1/ 4.26.27. 0. 0/                                              BD2   88
 DATA JJ  1/ 4.26.27. 0. 0/                                              BD2   89
 DATA VV  1/2.553E7. .292. 1.. 0.. 0./                                   BD2   90
 DATA II  2/ 4. 0. 0. 0. 0/                                              BD2   91
 DATA JJ  2/20. 0. 0. 0. 0/                                              BD2   92
 DATA VV  2/6.244E10. .1969. 1.. 0.. 0./                                 BD2   93
 DATA II  3/ 4. 0. 0. 0. 0/                                              BD2   94
 DATA JJ  3/ 5. 0. 0. 0. 0/                                              BD2   95
 DATA VV  3/1.497E7. .327. 1.0. 0.0. 0.0/                                BD2   96
 DATA II  4/ 2. 0. 0. 0. 0/                                              BD2   97
 DATA JJ  4/ 4. 0. 0. 0. 0/                                              BD2   98
 DATA VV  4/ 2.414E6. .364. 1.0. 0.. 0./                                 BD2   99
 DATA II  5/17. 2. 3. 0. 0/                                              BD2  100
 DATA JJ  5/17.17.17. 0. 0/                                              BD2  101
 DATA VV  5/999.. 6.88. 0.. 0.. 0./                                      BD2  102
 DATA II  6/17. 0. 0. 0. 0/                                              BD2  103
 DATA JJ  6/17. 0. 0. 0. 0/                                              BD2  104
 DATA VV  6/173.. 1.1156. 1.0. 0.. 0./                                   BD2  105
 DATA II  7/ 2. 0. 0. 0. 0/                                              BD2  106
 DATA JJ  7/17. 0. 0. 0. 0/                                              BD2  107
 DATA VV  7/166.. 1.1145. 1.0. 0.. 0./                                   BD2  108
 DATA II  8/ 3. 0. 0. 0. 0/                                              BD2  109
 DATA JJ  8/17. 0. 0. 0. 0/                                              BD2  110
```

```
DATA VV  8/142.. 1.1408. 1.0. 0.. 0./                                   BD2 111
DATA II  9/ 3. 0. 0. 0. 0/                                              BD2 112
DATA JJ  9/17. 0. 0. 0. 0/                                              BD2 113
DATA VV  9/1.0. 51.. 2.. 0.. 0./                                        BD2 114
DATA II 10/ 2. 0. 0. 0. 0/                                              BD2 115
DATA JJ 10/ 9. 0. 0. 0. 0/                                              BD2 116
DATA VV 10/ 998.. 4.824. 0.. 0.. 0./                                    BD2 117
DATA II 11/ 3. 0. 0. 0. 0/                                              BD2 118
DATA JJ 11/10. 0. 0. 0. 0/                                              BD2 119
DATA VV 11/ 997.. 4.722. 0.. 0.. 0./                                    BD2 120
DATA II 12/ 2. 0. 0. 0. 0/                                              BD2 121
DATA JJ 12/ 2. 0. 0. 0. 0/                                              BD2 122
DATA VV 12/ 1.0.  81.. 20.. 0.. 0./                                     BD2 123
DATA II 13/ 2. 2. 2. 0. 0/                                              BD2 124
DATA JJ 13/15. 7. 5. 0. 0/                                              BD2 125
DATA VV 13/ 1.0. 101.. 20.. 0.. 0./                                     BD2 126
DATA II 14/ 2. 0. 0. 0. 0/                                              BD2 127
DATA JJ 14/ 3. 0. 0. 0. 0/                                              BD2 128
DATA VV 14/ 1.0. 121.. 20.. 0.. 0./                                     BD2 129
DATA II 15/14.13. 5. 6. 7/                                              BD2 130
DATA JJ 15/14.14.14.14.14/                                              BD2 131
DATA VV 15/ 1.0. 141.. 20.. 0.. 0./                                     BD2 132
DATA II 16/14.13. 7.13. 5/                                              BD2 133
DATA JJ 16/15.15.13.13.13/                                              BD2 134
DATA VV 16/1.0. 141.. 20.. 0.. 0./                                      BD2 135
DATA II 17/ 6. 7. 5. 5. 5/                                              BD2 136
DATA JJ 17/13. 7. 7.15. 5/                                              BD2 137
DATA VV 17/1.0. 141.. 20.. 0.. 0./                                      BD2 138
DATA II 18/ 3. 3. 3. 2. 0/                                              BD2 139
DATA JJ 18/ 5.15. 7. 6. 0/                                              BD2 140
DATA VV 18/ 1.0. 161.. 20.. 0.. 0./                                     BD2 141
DATA II 19/ 6. 6. 5. 5. 5/                                              BD2 142
DATA JJ 19/15. 7.15. 7. 7/                                              BD2 143
DATA VV 19/ 1.0. 181.. 20.. 0.. 0./                                     BD2 144
DATA II 20/ 7. 5. 0. 0. 0/                                              BD2 145
DATA JJ 20/ 7. 6. 0. 0. 0/                                              BD2 146
DATA VV 20/1.0. 181.. 20.. 0.. 0./                                      BD2 147
DATA II 21/ 3. 0. 0. 0. 0/                                              BD2 148
DATA JJ 21/ 3. 0. 0. 0. 0/                                              BD2 149
DATA VV 21/ 1.0. 201.. 20.. 0.. 0./                                     BD2 150
DATA II 22/ 3. 3. 2. 3. 0/                                              BD2 151
DATA JJ 22/15. 7. 6. 6. 0/                                              BD2 152
DATA VV 22/ 1.0. 221.. 20.. 0.. 0./                                     BD2 153
DATA II 23/ 6. 6. 6. 7. 0/                                              BD2 154
DATA JJ 23/15. 7. 6. 7. 0/                                              BD2 155
DATA VV 23/ 1.0. 241.. 20.. 0.. 0./                                     BD2 156
DATA II 24/ 2. 2. 0. 0. 0/                                              BD2 157
DATA JJ 24/15. 7. 0. 0. 0/                                              BD2 158
DATA VV 24/ 2.0. 6.0. 0.5. 1.0. 1.0/                                    BD2 159
DATA II 25/ 2. 2. 0. 0. 0/                                              BD2 160
DATA JJ 25/15. 7. 0. 0. 0/                                              BD2 161
DATA VV 25/0.5. 1.. 1.. 0.. 0./                                         BD2 162
DATA II 26/ 2. 2. 0. 0. 0/                                              BD2 163
DATA JJ 26/ 5. 3. 0. 0. 0/                                              BD2 164
DATA VV 26/1.0. 1.. 1.. 0.. 0./                                         BD2 165
```

```
      DATA II 27/14. 0. 0. 0. 0/                                        BD2 166
      DATA JJ 27/14. 0. 0. 0. 0/                                        BD2 167
      DATA VV 27/.913. 1.117. 1.0. 0.. 0./                              BD2 168
      DATA II 28/13. 0. 0. 0. 0/                                        BD2 169
      DATA JJ 28/14. 0. 0. 0. 0/                                        BD2 170
      DATA VV 28/.954. 1.195. 1.0. 0.. 0./                              BD2 171
      DATA II 29/ 5. 0. 0. 0. 0/                                        BD2 172
      DATA JJ 29/14. 0. 0. 0. 0/                                        BD2 173
      DATA VV 29/.951. 1.124. 1.0. 0.. 0./                              BD2 174
      DATA II 30/ 6. 0. 0. 0. 0/                                        BD2 175
      DATA JJ 30/14. 0. 0. 0. 0/                                        BD2 176
      DATA VV 30/.891. 1.119. 1.0. 0.. 0./                              BD2 177
      DATA II 31/ 7.14. 0. 0. 0/                                        BD2 178
      DATA JJ 31/14.15. 0. 0. 0/                                        BD2 179
      DATA VV 31/.923. 1.062. 1.0. 0.. 0./                              BD2 180
      DATA II 32/13. 7. 0. 0. 0/                                        BD2 181
      DATA JJ 32/15.13. 0. 0. 0/                                        BD2 182
      DATA VV 32/.964. 1.082. 1.0. 0.. 0./                              BD2 183
      DATA II 33/13. 0. 0. 0. 0/                                        BD2 184
      DATA JJ 33/13. 0. 0. 0. 0/                                        BD2 185
      DATA VV 33/1.00. 1.274. 1.0. 0.. 0./                              BD2 186
      DATA II 34/ 5. 0. 0. 0. 0/                                        BD2 187
      DATA JJ 34/13. 0. 0. 0. 0/                                        BD2 188
      DATA VV 34/.982. 1.117. 1.0. 0.. 0./                              BD2 189
      DATA II 35/ 6. 0. 0. 0. 0/                                        BD2 190
      DATA JJ 35/13. 0. 0. 0. 0/                                        BD2 191
      DATA VV 35/.941. 1.115. 1.0. 0.. 0./                              BD2 192
      DATA II 36/ 7. 0. 0. 0. 0/                                        BD2 193
      DATA JJ 36/ 7. 0. 0. 0. 0/                                        BD2 194
      DATA VV 36/ .25. 1.. 1.. 0.. 0./                                  BD2 195
      DATA II 37/ 5. 5. 0. 0. 0/                                        BD2 196
      DATA JJ 37/ 7.15. 0. 0. 0/                                        BD2 197
      DATA VV 37/0.50. 1.. 1.. 0.. 0./                                  BD2 198
      DATA II 38/ 5. 3. 0. 0. 0/                                        BD2 199
      DATA JJ 38/ 5. 5. 0. 0. 0/                                        BD2 200
      DATA VV 38/1.00. 1.. 1.. 0.. 0./                                  BD2 201
      DATA II 39/ 3. 3. 2. 6. 6/                                        BD2 202
      DATA JJ 39/15. 7. 6.15. 7/                                        BD2 203
      DATA VV 39/0.50. 1.. 1.. 0.. 0./                                  BD2 204
      DATA II 40/15. 7. 0. 0. 0/                                        BD2 205
      DATA JJ 40/15.15. 0. 0. 0/                                        BD2 206
      DATA VV 40/7.. 7.. 1.0. 1.0. 1.0/                                 BD2 207
      DATA II 41/ 1. 0. 0. 0. 0/                                        BD2 208
      DATA JJ 41/14. 0. 0. 0. 0/                                        BD2 209
      DATA VV 41/1.0. 261.. 20.. 0.. 0./                                BD2 210
      DATA II 42/ 1. 0. 0. 0. 0/                                        BD2 211
      DATA JJ 42/13. 0. 0. 0. 0/                                        BD2 212
      DATA VV 42/1.0. 281.. 20.. 0.. 0./                                BD2 213
      DATA II 43/ 1. 1. 0. 0. 0/                                        BD2 214
      DATA JJ 43/ 2.17. 0. 0. 0/                                        BD2 215
      DATA VV 43/ 1.. 71.. 2.. 0.. 0./                                  BD2 216
      DATA II 44/ 1. 1. 0. 0. 0/                                        BD2 217
      DATA JJ 44/15. 7. 0. 0. 0/                                        BD2 218
      DATA VV 44/ 1.. 301.. 20.. 0.. 0./                                BD2 219
      DATA II 45/ 1. 0. 0. 0. 0/                                        BD2 220
```

```
      DATA JJ 45/ 5. 0. 0. 0. 0/                                        BD2 221
      DATA VV 45/ 1.. 321.. 20.. 0.. 0./                                BD2 222
      DATA II 46/ 1. 0. 0. 0. 0/                                        BD2 223
      DATA JJ 46/ 3. 0. 0. 0. 0/                                        BD2 224
      DATA VV 46/ 1.. 341.. 20.. 0.. 0./                                BD2 225
      DATA II 47/ 1. 0. 0. 0. 0/                                        BD2 226
      DATA JJ 47/ 6. 0. 0. 0. 0/                                        BD2 227
      DATA VV 47/ 1.. 361.. 20.. 0.. 0./                                BD2 228
      DATA II 48/ 1. 0. 0. 0. 0/                                        BD2 229
      DATA JJ 48/ 4. 0. 0. 0. 0/                                        BD2 230
      DATA VV 48/ 1.. 381.. 20.. 0.. 0./                                BD2 231
      DATA II 49/ 1. 1. 1. 1. 0/                                        BD2 232
      DATA JJ 49/17.15.14.13. 0/                                        BD2 233
      DATA VV 49/ 2.. 1.. 1.. 0.. 0./                                   BD2 234
      DATA II 50/ 1. 1. 1. 1. 1/                                        BD2 235
      DATA JJ 50/ 2. 7. 5. 3. 6/                                        BD2 236
      DATA VV 50/ 2.. 1.. 1.. 0.. 0./                                   BD2 237
      DATA II 51/16.16.16.21.21/                                        BD2 238
      DATA JJ 51/16.21.22.22.21/                                        BD2 239
      DATA VV 51/ 2.645E6. .2353. 1.. 0..0./                            BD2 240
      DATA II 52/22.24. 0. 0. 0/                                        BD2 241
      DATA JJ 52/22.24. 0. 0. 0/                                        BD2 242
      DATA VV 52/ 2.645E6. .2353. 1.. 0..0./                            BD2 243
      DATA II 53/16.16.19.19.19/                                        BD2 244
      DATA JJ 53/19.23.21.22.24/                                        BD2 245
      DATA VV 53/ 5.153E5. .4636. 1.. 0.. 0./                           BD2 246
      DATA II 54/21.22.23. 0. 0/                                        BD2 247
      DATA JJ 54/23.23.24. 0. 0/                                        BD2 248
      DATA VV 54/ 5.153E5. .4636. 1.. 0.. 0./                           BD2 249
      DATA II 55/ 1. 1. 1. 1. 0/                                        BD2 250
      DATA JJ 55/16.21.22.24. 0/                                        BD2 251
      DATA VV 55/ 1.. 471.. 20.. 0.. 0./                                BD2 252
      DATA II 56/ 0. 0. 0. 0. 0/                                        BD2 253
      DATA JJ 56/ 0. 0. 0. 0. 0/                                        BD2 254
      DATA VV 56/5*0.0/                                                 BD2 255
      DATA II 57/16.16.16.16.19/                                        BD2 256
      DATA JJ 57/16.19.21.22.21/                                        BD2 257
      DATA VV 57/ 13.458. 1.413. 4.003. 1.. 0./                         BD2 258
      DATA II 58/19.21.21.22. 0/                                        BD2 259
      DATA JJ 58/22.21.22.22. 0/                                        BD2 260
      DATA VV 58/ 13.458. 1.413. 4.003. 1.. 0./                         BD2 261
      DATA II 59/17. 7. 0. 0. 0/                                        BD2 262
      DATA JJ 59/18. 8. 0. 0. 0/                                        BD2 263
      DATA VV 59/ 996.. 5.. 0.. 0.. 0./                                 BD2 264
      DATA II 60/17.17. 2. 2. 0/                                        BD2 265
      DATA JJ 60/17.18. 2. 9. 0/                                        BD2 266
      DATA VV 60/ 15.3. 1.6. 14.008. 1.. 0./                            BD2 267
      DATA II 61/ 3. 3. 0. 0. 0/                                        BD2 268
      DATA JJ 61/ 3.10. 0. 0. 0/                                        BD2 269
      DATA VV 61/ 14.9. 1.57. 16.000. 1.. 0./                           BD2 270
      DATA II 62/ 4. 4.26.27. 0/                                        BD2 271
      DATA JJ 62/ 4.20.26.27. 0/                                        BD2 272
      DATA VV 62/ 15.93. 1.7. 39.944. 1.. 0./                           BD2 273
      DATA II 63/ 7. 0. 0. 0. 0/                                        BD2 274
      DATA JJ 63/ 8. 0. 0. 0. 0/                                        BD2 275
```

```
 DATA VV 63/ 14.7, 1.67, 30.008, 1., 0./                                  BD2 276
 DATA TL1/                                                                BD2 277
13.5000E 00,3.7500E 00,4.0000E 00,4.2500E 00,4.5000E 00,4.7500E 00,BD2 278
25.0000E 00,5.2500E 00,5,5000E 00,5.7500E 00,6.0000E 00,6.2500E C0,BD2 279
 .5000E 00,6.7500E 00,7.0000E 00,7.2500E 00,7.5000E 00,7.7500E 00,BD2 280
 8.0000E 00,8.2500E 00,8.5000E 00,8.7500E 00,9.0000E 00,9.5000E C0,BD2 281
51,0000E 01,1.0500E 01,1.1000E 01,1.1500E 01,1.2000E 01,1.2500E 01,BD2 282
61.3000E 01,1.3500E 01,1.4000E 01,1.4500E 01,1.5000E 01,1.5500E 01,BD2 283
71.6000E 01,1.7000E 01,1.8000E 01,1.9000E 01,2.0000E 01,2.1000E 01,BD2 284
82,2000E 01,2.3000E 01,2,4000E 01,2.5000E 01,2.6000E 01,2.7000E 01,BD2 285
92.8000E 01,2.8500E 01,  1000.,  30000.,  18*0.,  1000.,  30000.,  8*0.,BD2 286
$                     1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,BD2 287
$5.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,BD2 288
$1,2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,2,0000E 04,2.2000E 04,BD2 289
$2.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04/                            BD2 290
 DATA TL2/                                                               BD2 291
11.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,BD2 292
27.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,BD2 293
31.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,BD2 294
42,8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,BD2 295
55.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,BD2 296
61.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,BD2 297
72.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,BD2 298
83.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,BD2 299
99.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,BD2 300
$2.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04,BD2 301
$1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,BD2 302
$7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,1,2000E 04,1.4000E 04,BD2 303
$1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,2,4000E 04,2.6000E 04,BD2 304
$2.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,BD2 305
$5.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,9,0000E 03,1.0000E 04,BD2 306
$1.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,BD2 307
$2.4000E 04,2,6000E 04,2.8000E 04,3.0000E 04/                            BD2 308
 DATA TL3/                                                               BD2 309
11.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,BD2 310
27.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,BD2 311
31,6000E 04,1.8000E 04,2.0000E 04,2,2000E 04,2,4000E 04,2,6000E 04,BD2 312
42.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,BD2 313
55.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,BD2 314
61,2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,2,0000E 04,2.2000E 04,BD2 315
72.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,BD2 316
83.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,BD2 317
99.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,BD2 318
$2.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04,BD2 319
$1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,BD2 320
$7,0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,BD2 321
$1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,BD2 322
$2.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,BD2 323
$5,0000E 03,6,0000E 03,7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,BD2 324
$1.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,BD2 325
$2.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04/                            BD2 326
 DATA TL4/                                                               BD2 327
11.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,BD2 328
27.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,BD2 329
31.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,BD2 330
```

```
42.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,BD2  331
55.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,BD2  332
61.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,BD2  333
72.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,BD2  334
83.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,BD2  335
99.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,BD2  336
$2.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04,BD2  337
$1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,BD2  338
$7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,BD2  339
$1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,BD2  340
$2.8000E 04,3.0000E 04,1.0000E 03,2.0000E 03,3.0000E 03,4.0000E 03,BD2  341
$5.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,9.0000E 03,1.0000E 04,BD2  342
$1.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,2.0000E 04,2.2000E 04,BD2  343
$2.4000E 04,2.6000E 04,2.8000F 04,3.0000E 04/                           BD2  344
 DATA TL5/                                                              BD2  345
1  70*0.0,                                          1.0000E 03,2.0000E 03,BD2  346
23.0000E 03,4.0000E 03,5.0000E 03,6.0000E 03,7.0000E 03,8.0000E 03,BD2  347
39.0000E 03,1.0000E 04,1.2000E 04,1.4000E 04,1.6000E 04,1.8000E 04,BD2  348
42.0000E 04,2.2000E 04,2.4000E 04,2.6000E 04,2.8000E 04,3.0000E 04,BD2  349
5 10*0./                                                                BD2  350
 DATA TL6/                                                              BD2  351
11.0000E-01,2.0000E-01,3.0000E-01,4.0000E-01,5.0000E-01,6.0000E-01,BD2  352
27.0000E-01,8.0000F-01,9.0000E-01,1.0000E 00,1.2000E 00,1.4000E 00,BD2  353
31.6000E 00,1.8000E 00,2.0000E 00,2.5000E 00,3.0000E 00,3.5000E 00,BD2  354
44.0000E 00,5.0000E 00,6.0000E 00,7.0000E 00,8.0000E 00,9.0000E 00,BD2  355
51.0000E 01,1.2000E 01,1.4000E 01,1.6000E 01,1.8000E 01,2.0000E 01,BD2  356
62.5000E 01,3.0000E 01,3.5000E 01,4.0000E 01,5.0000E 01,7.5000E 01,BD2  357
71.0000E 02. 63*0.0 /                                                   BD2  358
 DATA TL11/                                                             BD2  359
11.8756E-01,1.9016E-01,1.9259E-01,1.9484E-01,1.9693E-01,1.9888E-01,BD2  360
22.0070E-01,2.0239E-01,2.0397E-01,2.0545E-01,2.0684E-01,2.0815E-01,BD2  361
32.0937E-01,2.1053E-01,2.1162E-01,2.1265E-01,2.1362E-01,2.1455E-01,BD2  362
42.1543E-01,2.1626E-01,2.1705E-01,2.1781E-01,2.1853E-01,2.1988E-01,BD2  363
52.2111E-01,2.2224E-01,2.2329E-01,2.2425E-01,2.2515E-01,2.2598E-01,BD2  364
62.2676E-01,2.2749E-01,2.2817E-01,2.2881E-01,2.2941E-01,2.2998E-01,BD2  365
72.3052E-01,2.3151E-01,2.3240E-01,2.3320E-01,2.3394E-01,2.3461E-01,BD2  366
82.3522E-01,2.3579E-01,2.3631E-01,2.3680E-01,2.3725E-01,2.3767E-01,BD2  367
92.3806E-01,2.3825E-01, .67, .67, 28*5.0,                               BD2  368
$                          1.9590E 01,1.6040E 01,1.4260E 01,1.3090E 01,BD2  369
$1.2240E 01,1.1570E 01,1.1020E 01,1.0570E 01,1.0180E 01,9.8400E 00,BD2  370
$9.2600E 00,8.8000E 00,8.4100E 00,8.0800E 00,7.7900E 00,7.5200E 00,BD2  371
$7.2900E 00,7.0900E 00,6.9000E 00,6.7300E 00/                           BD2  372
 DATA TL12/                                                             BD2  373
12.5810E 01,2.1490E 01,1.9160E 01,1.7580E 01,1.6410E 01,1.5480E 01,BD2  374
21.4720E 01,1.4070E 01,1.3510E 01,1.3030E 01,1.2200E 01,1.1530E 01,BD2  375
31.0930E 01,1.0500E 01,1.0130E 01,9.7800E 00,9.4800E 00,9.2200E 00,BD2  376
48.9700E 00,8.7500E 00,2.2480E 01,1.7340E 01,1.4980E 01,1.3660E 01,BD2  377
51.2740E 01,1.2030E 01,1.1450E 01,1.0970E 01,1.0560E 01,1.0200E 01,BD2  378
69.6000E 00,9.1100E 00,8.7200E 00,8.4000E 00,8.0900E 00,7.8400E 00,BD2  379
77.6100E 00,7.4200E 00,7.2200E 00,7.0400E 00,3.1640E 01,2.7630E 01,BD2  380
82.4650E 01,2.2420E 01,2.0730E 01,1.9440E 01,1.8420E 01,1.7580E 01,BD2  381
91.6980E 01,1.6330E 01,1.5420E 01,1.4850E 01,1.4300E 01,1.3740E 01,BD2  382
$1.3240E 01,1.2780E 01,1.2390E 01,1.2050E 01,1.1730E 01,1.1440E 01,BD2  383
$2.7860E 01,2.3160E 01,2.0610E 01,1.8900E 01,1.7610E 01,1.6610E 01,BD2  384
$1.5780E 01,1.5080E 01,1.4470E 01,1.3940E 01,1.3050E 01,1.2320E 01,BD2  385
```

```
 1.1790E 01,1.1360E 01,1.0940E 01,1.0600. 01,1.0290E 01,1.0030E 01,BD2  386
$9.7700E 00,9.5100E 00,2.9080E 01,2.6300E 01,2.4780E 01,2.3670E 01,BD2  387
$2.2720E 01,2.1900E 01,2.1180E 01,2.0510E 01,1.9780E 01,1.9130E 01,BD2  388
$1.8050E 01,1.7150E 01,1.6420E 01,1.5820E 01,1.5240E 01,1.4770E 01,BD2  389
$1.4340E 01,1.3970E 01,1.3600E 01,1.3250E 01/                          BD2  390
 DATA TL13/                                                            BD2  391
11.9460E 01,1.6560E 01,1.4910E 01,1.3780E 01,1.2930E 01,1.2270E 01,BD2  392
21.1710E 01,1.1230E 01,1.0810E 01,1.0480E 01,9.8900E 00,9.4100E 00,BD2  393
39.0100E 00,8.6800E 00,8.3600E 00,8.1000E 00,7.8700E 00,7.6700E 00,BD2  394
47.4600E 00,7.2700E 00,2.3590E 01,2.0320E 01,1.8520E 01,1.7290E 01,BD2  395
51.6370E 01,1.5630E 01,1.5030E 01,1.4510E 01,1.4060E 01,1.3670E 01,BD2  396
61.3000E 01,1.2440E 01,1.1910E 01,1.1480E 01,1.1050E 01,1.0710E 01,BD2  397
71.0400E 01,1.0130E 01,9.8700E 00,9.6100E 00,2.7880E 01,2.5270E 01,BD2  398
82.3900E 01,2.2870E 01,2.2030E 01,2.1200E 01,2.0440E 01,1.9810E 01,BD2  399
91.9060E 01,1.8780E 01,1.7960E 01,1.7300E 01,1.6560E 01,1.5950E 01,BD2  400
$1.5360E 01,1.4890E 01,1.4460E 01,1.4090E 01,1.3720E 01,1.3360E 01,BD2  401
$6.5000E 00,8.2000E 00,9.6000E 00,1.0700E 01,1.1900E 01,1.3100E 01,BD2  402
$1.4300E 01,1.5200E 01,1.6400E 01,1.7600E 01,1.9700E 01,2.1000E 01,BD2  403
$2.0000E 01,1.5100E 01,1.3000E 01,1.1900E 01,1.1500E 01,1.1200E 01,BD2  404
$1.1000E 01,1.1000E 01,2.0000E 01,1.3700E 01,1.0900E 01,9.1000E 00,BD2  405
$7.7000E 00,6.7000E 00,6.0000E 00,5.4000E 00,4.9000E 00,4.7000E 00,BD2  406
$4.7000E 00,5.2000E 00,5.7000E 00,6.1000E 00,6.4000E 00,6.6000E 00,BD2  407
$6.8000E 00,6.9000E 00,7.0000E 00,7.0000E 00/                          BD2  408
 DATA TL14/                                                            BD2  409
13.6000E 00,5.0000E 00,6.6000E 00,8.4000E 00,1.0000E 01,1.1000E 01,BD2  410
21.1400E 01,1.1000E 01,1.0300E 01,1.0100E 01,1.0100E 01,1.0100E 01,BD2  411
31.0100E 01,1.0100E 01,1.0100E 01,1.0100E 01,1.0100E 01,1.0100E 01,BD2  412
41.0100E 01,1.0100E 01,5.3000E 00,7.7000E 00,8.8000E 00,9.5000E 00,BD2  413
59.9000E 00,1.0300E 01,1.0600E 01,1.0900E 01,1.1200E 01,1.1500E 01,BD2  414
61.2300E 01,1.3400E 01,1.4600E 01,1.6100E 01,1.8100E 01,2.0000E 01,BD2  415
72.2000E 01,2.4000E 01,2.6000E 01,2.8000E 01,1.0100E 00,1.1400E 00,BD2  416
81.3200E 00,1.5000E 00,1.6700E 00,1.8500E 00,2.0200E 00,2.2400E 00,BD2  417
92.4600E 00,2.6400E 00,3.0500E 00,3.4300E 00,3.7800E 00,4.0900E 00,BD2  418
$4.4000E 00,4.5800E 00,4.7500E 00,4.8800E 00,4.9700E 00,5.0600E 00,BD2  419
$2.7000E 00,3.0000E 00,3.5000E 00,4.1000E 00,4.8000E 00,5.7000E 00,BD2  420
$6.7000E 00,7.6000E 00,8.3000E 00,8.5000E 00,8.5000E 00,8.5000E 00,BD2  421
$8.5000E 00,8.5000E 00,8.5000E 00,8.5000E 00,8.5000E 00,8.5000E 00,BD2  422
$8.5000E 00,8.5000E 00,4.2000E-01,3.5000E-01,3.0000E-01,4.0000E-01,BD2  423
$6.0000E-01,8.0000E-01,1.0000E 00,1.1500E 00,1.3000E 00,1.5000E 00,BD2  424
$1.7000E 00,2.0000E 00,2.3000E 00,2.6000E 00,2.9000E 00,3.2000E 00,BD2  425
$3.5000E 00,3.8000E 00,4.1000E 00,4.4000E 00/                          BD2  426
 DATA TL15/                                                            BD2  427
1 70*0.0,                                                                   428
                                                 5.8800E 00,6.0800E 00,BD2  428
26.2000E 00,6.2700E 00,6.3300E 00,6.3700E 00,6.4000E 00,6.4200E 00,BD2  429
36.4400E 00,6.4600E 00,6.4900E 00,6.5100E 00,6.5300E 00,6.5500E 00,BD2  430
46.5700E 00,6.5900E 00,6.6100E 00,6.6400E 00,6.6700E 00,6.7000E 00,BD2  431
5 10*0./                                                               BD2  432
 DATA TL16/                                                            BD2  433
14.0079E 00,3.1300E 00,2.6494E 00,2.3144E 00,2.0661E 00,1.8767E 00,BD2  434
21.7293E 00,1.6122E 00,1.5175E 00,1.4398E 00,1.3204E 00,1.2336E 00,BD2  435
31.1679E 00,1.1166E 00,1.0753E 00,1.0006E 00,9.5003E-01,9.1311E-01,BD2  436
48.8453E-01,8.4277E-01,8.1287E-01,7.8976E-01,7.7111E-01,7.5553E-01,BD2  437
57.4220E-01,7.2022E-01,7.0254E-01,6.8776E-01,6.7510E-01,6.6405E-01,BD2  438
66.4136E-01,6.2350E-01,6.0882E-01,5.9640E-01,5.7626E-01,5.4146E-01,BD2  439
75.1803E-01, 63*0.0 /                                                  BD2  440
```

```
 DATA TL20/ 95*0.. 948.. 879.. 1000.. 1.. 0. /                                  BD2 441
 DATA TL21/                                                                     BD2 442
11.3417E 00,1.3311E 00,1.3206E 00,1.3104E 00,1.3006E 00,1.2912E 00,BD2 443
21.2822E 00,1.2736E 00,1.2655E 00,1.2578E 00,1.2504E 00,1.2435E 00,BD2 444
31.2368E 00,1.2305E 00,1.2246E 00,1.2189E 00,1.2134E 00,1.2083E 00,BD2 445
41.2033E 00,1.1986E 00,1.1941E 00,1.1898E 00,1.1857E 00,1.1779E 00,BD2 446
51.1708E 00,1.1642E 00,1.1581E 00,1.1524E 00,1.1472E 00,1.1422E 00,BD2 447
61.1376E 00,1.1333E 00,1.1292E 00,1.1254E 00,1.1218E 00,1.1184E 00,BD2 448
71.1152E 00,1.1093E 00,1.1039E 00,1.0990E 00,1.0946E 00,1.0906E 00,BD2 449
81.0868E 00,1.0834E 00,1.0802E 00,1.0773E 00,1.0746E 00,1.0720E 00,BD2 450
91.0697E 00,1.0685E 00, .67, .67, 28*1.0,                                       BD2 451
$                      2.2110E 01,1.8260E 01,1.6310E 01,1.5040E 01,BD2 452
$1.4100E 01,1.3360E 01,1.2760E 01,1.2260E 01,1.1820E 01,1.1440E 01,BD2 453
$1.0810E 01,1.0290E 01,9.8500E 00,9.4900E 00,9.1400E 00,8.8600E 00,BD2 454
$8.6000E 00,8.3800E 00,8.1600E 00,7.9500E 00/                                  BD2 455
 DATA TL22/                                                                     BD2 456
13.0860E 01,2.5990E 01,2.3330E 01,2.1520E 01,2.0170E 01,1.9100E 01,BD2 457
21.8220E 01,1.7460E 01,1.6820E 01,1.6240E 01,1.5290E 01,1.4490E 01,BD2 458
31.3790E 01,1.3290E 01,1.2800E 01,1.2400E 01,1.2040E 01,1.1730E 01,BD2 459
41.1420E 01,1.1130E 01,2.6450E 01,2.0400E 01,1.7610E 01,1.5940E 01,BD2 460
51.4910E 01,1.4110E 01,1.3450E 01,1.2910E 01,1.2440E 01,1.2030E 01,BD2 461
61.1340E 01,1.0790E 01,1.0320E 01,9.9500E 00,9.5800E 00,9.2900E 00,BD2 462
79.0100E 00,8.7800E 00,8.5500E 00,8.3300E 00,3.5090E 01,3.1450E 01,BD2 463
82.8500E 01,2.6230E 01,2.4520E 01,2.3230E 01,2.2100E 01,2.1110E 01,BD2 464
92.0340E 01,1.9730E 01,1.8700E 01,1.7920E 01,1.7240E 01,1.6610E 01,BD2 465
$1.6000E 01,1.5510E 01,1.5050E 01,1.4670E 01,1.4280E 01,1.3910E 01,BD2 466
$3.3330E 01,2.8040E 01,2.5130E 01,2.3170E 01,2.1670E 01,2.0530E 01,BD2 467
$1.9570E 01,1.8760E 01,1.8050E 01,1.7430E 01,1.6380E 01,1.5520E 01,BD2 468
$1.4860E 01,1.4310E 01,1.3790E 01,1.3360E 01,1.2970E 01,1.2640E 01,BD2 469
$1.2310E 01,1.1990E 01,3.2920E 01,3.0790E 01,2.9570E 01,2.8530E 01,BD2 470
$2.7540E 01,2.6650E 01,2.5860E 01,2.5100E 01,2.4270E 01,2.3520E 01,BD2 471
$2.2270E 01,2.1240E 01,2.0330E 01,1.9590E 01,1.8860E 01,1.8290E 01,BD2 472
$1.7700E 01,1.7300E 01,1.6840E 01,1.6410E 01/                                  BD2 473
 DATA TL23/                                                                     BD2 474
12.2560E 01,1.9290E 01,1.7490E 01,1.6250E 01,1.5320E 01,1.4580E 01,BD2 475
21.3940E 01,1.3420E 01,1.2940E 01,1.2550E 01,1.1880E 01,1.1340E 01,BD2 476
31.0850E 01,1.0460E 01,1.0070E 01,9.7600E 00,9.4800E 00,9.2300E 00,BD2 477
48.9900E 00,8.7600E 00,2.7580E 01,2.3950E 01,2.1940E 01,2.0570E 01,BD2 478
51.9540E 01,1.8710E 01,1.8030E 01,1.7450E 01,1.6940E 01,1.6490E 01,BD2 479
61.5730E 01,1.5100E 01,1.4460E 01,1.3930E 01,1.3420E 01,1.3010E 01,BD2 480
71.2620E 01,1.2300E 01,1.1980E 01,1.1670E 01,3.1090E 01,2.8980E 01,BD2 481
82.7770E 01,2.6750E 01,2.5910E 01,2.5020E 01,2.4200E 01,2.3500E 01,BD2 482
92.2890E 01,2.2350E 01,2.1430E 01,2.0670E 01,1.9790E 01,1.9060E 01,BD2 483
$1.8360E 01,1.7800E 01,1.7280E 01,1.6830E 01,1.6390E 01,1.5970E 01,BD2 484
$ 40*1./                                                                       BD2 485
 DATA TL24/                                                                     BD2 486
1 80*1.                    ,3.6000E-01,3.0000E-01,3.0000E-01,4.0000E-01,BD2 487
26.0000E-01,8.0000E-01,1.0000E 00,1.2000E 00,1.4000E 00,1.6000E 00,BD2 488
32.0000E 00,2.4000E 00,2.8000E 00,3.2000E 00,3.6000E 00,4.0000E 00,BD2 489
44.4000E 00,4.8000E 00,5.2000E 00,5.6000E 00/                                  BD2 490
 DATA TL25/                                                                     BD2 491
1 70*0.0,                                          5.7000E 00,5.8400E 00,BD2 492
25.9200E 00,5.9600E 00,5.9900E 00,6.0100E 00,6.0200E 00,6.0300E 00,BD2 493
36.0400E 00,6.0500E 00,6.0600E 00,6.0700E 00,6.0800E 00,6.1000E 00,BD2 494
46.1200E 00,6.1400E 00,6.1600E 00,6.1900E 00,6.2300E 00,6.2600E 00,BD2 495
```

```
5 10*0./                                                                BD2 496
 DATA TL26/                                                             BD2 497
11.0231E 00,1.0424E 00,1.0719E 00,1.0936E 00,1.1053E 00,1.1104E 00,BD2 498
21.1114E 00,1.1104E 00,1.1086E 00,1.1063E 00,1.1020E 00,1.0985E 00,BD2 499
31.0960E 00,1.0943E 00,1.0934E 00,1.0926E 00,1.0934E 00,1.0948E 00,BD2 500
41.0965E 00,1.0997E 00,1.1025E 00,1.1050E 00,1.1072E 00,1.1091E 00,BD2 501
51.1107E 00,1.1133E 00,1.1154E 00,1.1172E 00,1.1186E 00,1.1199E 00,BD2 502
61.1223E 00,1.1243E 00,1.1259E 00,1.1273E 00,1.1297E 00,1.1339E 00,BD2 503
71.1364E 00, 63*0.0 /                                                  BD2 504
 DATA TL30/ 95*0.0, .9247, 1., 1., 1., 0. /                             BD2 505
 DATA TL31/                                                             BD2 506
11.3482E 00,1.3296E 00,1.3126E 00,1.2969E 00,1.2825E 00,1.2693E 00,BD2 507
21.2572E 00,1.2460E 00,1.2358E 00,1.2262E 00,1.2174E 00,1.2093E 00,BD2 508
31.2016E 00,1.1945E 00,1.1879E 00,1.1817E 00,1.1759E 00,1.1704E 00,BD2 509
41.1653E 00,1.1604E 00,1.1558E 00,1.1515E 00,1.1474E 00,1.1398E 00,BD2 510
51.1330E 00,1.1267E 00,1.1211E 00,1.1159E 00,1.1111E 00,1.1067E 00,BD2 511
61.1027E 00,1.0989E 00,1.0954E 00,1.0921E 00,1.0891E 00,1.0862E 00,BD2 512
71.0835E 00,1.0786E 00,1.0743E 00,1.0704E 00,1.0669E 00,1.0637E 00,BD2 513
81.0608E 00,1.0582E 00,1.0557E 00,1.0535E 00,1.0515E 00,1.0496E 00,BD2 514
91.0478E 00,1.0469E 00,30*1.,                                          BD2 515
$                    1.1440E 00,1.1440E 00,1.1430E 00,1.1410E 00,BD2 516
$1.1410E 00,1.1420E 00,1.1410E 00,1.1410E 00,1.1410E 00,1.1410E 00,BD2 517
$1.1420E 00,1.1420E 00,1.1420E 00,1.1420E 00,1.1420E 00,1.1420E 00,BD2 518
$1.1420E 00,1.1420E 00,1.1420E 00,1.1420E 00,1.1420E 00/               BD2 519
 DATA TL32/                                                             BD2 520
11.1570E 00,1.1710E 00,1.1800E 00,1.1870E 00,1.1930E 00,1.1980E 00,BD2 521
21.2030E 00,1.2070E 00,1.2100E 00,1.2140E 00,1.2200E 00,1.2250E 00,BD2 522
31.2300E 00,1.2350E 00,1.2400E 00,1.2450E 00,1.2500E 00,1.2550E 00,BD2 523
41.2600E 00,1.2650E 00,1.1920E 00,1.1780E 00,1.1380E 00,1.1260E 00,BD2 524
51.1290E 00,1.1290E 00,1.1330E 00,1.1340E 00,1.1350E 00,1.1370E 00,BD2 525
61.1370E 00,1.1390E 00,1.1420E 00,1.1450E 00,1.1480E 00,1.1510E 00,BD2 526
71.1540E 00,1.1570E 00,1.1600E 00,1.1640E 00,1.0960E 00,1.1150E 00,BD2 527
81.1250E 00,1.1330E 00,1.1420E 00,1.1470E 00,1.1520E 00,1.1550E 00,BD2 528
91.1580E 00,1.1580E 00,1.1570E 00,1.1580E 00,1.1580E 00,1.1580E 00,BD2 529
$1.1580E 00,1.1570E 00,1.1570E 00,1.1560E 00,1.1550E 00,1.1540E 00,BD2 530
$1.1590E 00,1.1730E 00,1.1820E 00,1.1890E 00,1.1950E 00,1.2010E 00,BD2 531
$1.2050E 00,1.2090E 00,1.2130E 00,1.2170E 00,1.2230E 00,1.2290E 00,BD2 532
$1.2350E 00,1.2400E 00,1.2450E 00,1.2500E 00,1.2540E 00,1.2580E 00,BD2 533
$1.2620E 00,1.2660E 00,1.1130E 00,1.1400E 00,1.1590E 00,1.1690E 00,BD2 534
$1.1760E 00,1.1800E 00,1.1840E 00,1.1880E 00,1.1910E 00,1.1940E 00,BD2 535
$1.1990E 00,1.2030E 00,1.2070E 00,1.2110E 00,1.2150E 00,1.2180E 00,BD2 536
$1.2210E 00,1.2240E 00,1.2270E 00,1.2300E 00/                          BD2 537
 DATA TL33/                                                             BD2 538
11.1410E 00,1.1600E 00,1.1680E 00,1.1800E 00,1.1850E 00,1.1820E 00,BD2 539
21.1820E 00,1.1820E 00,1.1810E 00,1.1810E 00,1.1840E 00,1.1860E 00,BD2 540
31.1880E 00,1.1900E 00,1.1920E 00,1.1940E 00,1.1960E 00,1.1980E 00,BD2 541
41.2000E 00,1.2020E 00,1.1310E 00,1.1410E 00,1.1470E 00,1.1520E 00,BD2 542
51.1560E 00,1.1590E 00,1.1620E 00,1.1640E 00,1.1670E 00,1.1690E 00,BD2 543
61.1730E 00,1.1760E 00,1.1790E 00,1.1820E 00,1.1850E 00,1.1880E 00,BD2 544
71.1910E 00,1.1940E 00,1.1970E 00,1.2000E 00,1.0990E 00,1.1170E 00,BD2 545
81.1280E 00,1.1340E 00,1.1390E 00,1.1430E 00,1.1450E 00,1.1470E 00,BD2 546
91.1490E 00,1.1510E 00,1.1540E 00,1.1570E 00,1.1600E 00,1.1630E 00,BD2 547
$1.1660E 00,1.1690E 00,1.1720E 00,1.1750E 00,1.1780E 00,1.1800E 00,BD2 548
$ 40*1./                                                                BD2 549
 DATA TL34/100*1./                                                      BD2 550
```

```
 DATA TL35/100*1./                                                        BD2 551
 DATA TL36/                                                               BD2 552
11.1960E 00.1.2451E 00.1.2900E 00.1.2986E 00.1.2865E 00.1.2665E 00.BD2 553
21.2455E 00.1.2253E 00.1.2078E 00.1.1919E 00.1.1678E 00.1.1496E 00.BD2 554
31.1366E 00.1.1270E 00.1.1197E 00.1.1080E 00.1.1016E 00.1.0980E 00.BD2 555
41.0958E 00.1.0935E 00.1.0925E 00.1.0922E 00.1.0922E 00.1.0923E 00.BD2 556
51.0923E 00.1.0927E 00.1.0930E 00.1.0933E 00.1.0937E 00.1.0939E 00.BD2 557
61.0943E 00.1.0944E 00.1.0944E 00.1.0943E 00.1.0941E 00.1.0947E 00.BD2 558
71.0957E 00.63*1./                                                       BD2 559
 DATA TL40/ 95*0.0, 1.2133, 1., 1., 1.165, 0. /                          BD2 560
 END                                                                     BD2 561-
```

```
      SUBROUTINE AESOLN (X)                                                  AES    1
      LOGICAL ERR                                                            AES    2
C     ROUTINE FOR CALCULATING EFFECTIVE AREA RATIO AND ITS DERIVATIVE        AES    3
C     CALL TO AESOLN(X) MUST BE PRECEDED BY CALL GEOMAR(X,S1,S2)             AES    4
C           X       = AXIAL COORDINATE ALONG NOZZLE                          AES    5
C           DELBL   = BOUNDARY LAYER DISPLACEMENT THICKNESS DIVIDED BY R0    AES    6
C           S1      = GEOMETRIC AREA RATIO (PROVIDED BY CALLING ROUTINE)     AES    7
C                     S1 IS RESET TO EQUAL THE EFFECTIVE AREA RATIO          AES    8
C           S2      = DERIVATIVE OF S1 (PROVIDED BY THE CALLING ROUTINE)     AES    9
C                     S2 IS RESET TO THE DERIVATIVE OF THE EFFECTIVE AREA     AES   10
C                          RATIO                                             AES   11
C                                                                            AES   12
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),       AES   13
     1      SW,R0,JDIM,IPCINT                                                AES   14
      DIMENSION DELP(2),DELSTP(2)                                            AES   15
      COMMON /AREA/ ATP1(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),              AES   16
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                      AES   17
      COMMON /AEGEOM/ SQRTA,S1,S2                                            AES   18
      COMMON /ERROR/ ERR                                                     AES   19
      COMMON /NEQ/ OMDST(2),DDELBL(2)                                        AES   20
      IF (NPRFLS.EQ.2) GO TO 20                                              AES   21
      IF (JDIM.EQ.0) GO TO 10                                                AES   22
      SQRTAE=(SQRTA-DELBL(1))/OMDST(1)                                       AES   23
      S1=SQRTAE**2                                                           AES   24
      S2=(S2/SQRTA-2.*DDELBL(1))/OMDST(1)*SQRTAE                             AES   25
      GO TO 40                                                               AES   26
   10 S1=(S1-DELBL(1))/OMDST(1)                                              AES   27
      S2=(S2-DDELBL(1))/OMDST(1)                                             AES   28
      GO TO 40                                                               AES   29
   20 CALL GMAR3 (X,DYDX,DZDX,Y,Z)                                           AES   30
      IF (ERR) RETURN                                                        AES   31
      DO 30 K=1,2                                                            AES   32
      DELP(K)=R0*DELBL(K)                                                    AES   33
   30 DELSTP(K)=R0*(1.-OMDST(K))                                             AES   34
      DENOM=(RTHCM(1)-DELSTP(1))*(RTHCM(2)-DELSTP(2))                        AES   35
      FAC1=Y-DELP(1)                                                         AES   36
      FAC2  -DELP(2)                                                         AES   37
      S1=F  1*FAC2/DENOM                                                     AES   38
      S2=(FAC2*(DYDX-DDELBL(1)*R0)+FAC1*(DZDX-DDELBL(2)*R0))/DENOM           AES   39
   40 RETURN                                                                 AFS   40
      END                                                                    AES   41
```

```
      SUBROUTINE AGSOLN (AE,DEL,UPDOWN,AG,X)                                AGS    1
      LOGICAL ERR,FIX                                                       AGS    2
C     SOLUTION FOR THE GEOMETRIC AREA RATIO WHEN THE EFFECTIVE AREA         AGS    3
C          RATIO AND BOUNDARY LAYER DISPLACEMENT THICKNESS ARE GIVEN        AGS    4
C                                                                           AGS    5
C           AE     = EFFECTIVE AREA RATIO                                   AGS    6
C           DEL    = DISPLACEMENT THICKNESS                                 AGS    7
C           UPDOWN=  1. IF DOWNSTREAM SOLUTION IS DESIRED                   AGS    8
C                   -1. IF UPSTREAM SOLUTION IS DESIRED                     AGS    9
C           AG     = COMPUTED GEOMETRIC AREA RATIO                          AGS   10
C                                                                           AGS   11
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),             AGS   12
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                     AGS   13
      COMMON /NEQ/ CMDST(2),DDELBL(2)                                       AGS   14
      COMMON /ERROR/ ERR                                                    AGS   15
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),      AGS   16
     1   SW,R0,JDIM,IPCINT                                                  AGS   17
      DIMENSION DEL(2),DELP(2),DSTP(2)                                      AGS   18
      DATA RNAME /6HAGSOLN/                                                 AGS   19
      IF (NPRFLS.EQ.2) GO TO 20                                             AGS   20
C     NOT A CHANNEL                                                         AGS   21
      IF (JDIM.NE.0) GO TO 10                                               AGS   22
C     TWO-DIMENSIONAL FLOW                                                  AGS   23
      AG=DEL(1)+CMDST(1)*AE                                                 AGS   24
      GO TO 100                                                             AGS   25
C     AXISYMMETRIC FLOW                                                     AGS   26
10    SQRTA=DEL(1)+CMDST(1)*SQRT(AE)                                        AGS   27
      AG=SQRTA**2                                                           AGS   28
      GO TO 100                                                             AGS   29
C     CHANNEL                                                               AGS   30
20    DO 30 L=1,2                                                           AGS   31
      DSTP(L)=R0*(1.-CMDST(L))                                              AGS   32
30    DELP(L)=R0*DEL(L)                                                     AGS   33
      FIX=.FALSE.                                                           AGS   34
      YOZO=RTHCM(1)*RTHCM(2)                                                AGS   35
40    AEC=(1.-DSTP(1)/RTHCM(1))*(1.-DSTP(2)/RTHCM(2))*AE-DELP(1)*DELP(2)    AGS   36
     1/YOZO                                                                 AGS   37
      A=AE                                                                  AGS   38
      ICOUNT=0                                                              AGS   39
50    CALL FINDX (A,UPDOWN,X)                                               AGS   40
      IF (ERR) RETURN                                                       AGS   41
60    ICOUNT=ICOUNT+1                                                       AGS   42
      IF (ICOUNT.GT.20) GO TO 90                                            AGS   43
      CALL GMAR2 (X,Y,Z)                                                    AGS   44
      IF (ICOUNT.GT.1) GO TO 70                                             AGS   45
      A=AEC+(Y*DELP(2)+Z*DELP(1))/YOZO                                      AGS   46
      XO=X                                                                  AGS   47
      FO=A-AE                                                               AGS   48
      IF (A.GE.1.) GO TO 50                                                 AGS   49
      X=0.                                                                  AGS   50
      GO TO 60                                                              AGS   51
70    CALL GEOMAR (X,A,DA)                                                  AGS   52
      IF (ERR) RETURN                                                       AGS   53
      F=AEC+(Y*DELP(2)+Z*DELP(1))/YOZO-A                                    AGS   54
      IF (ABS(F)/A.LE.1.E-5) GO TO 120                                      AGS   55
```

```
      IF (F.EQ.FO) GO TO 120                                                AGS   56
      XL=X                                                                  AGS   57
      X=X-F*(X-XO)/(F-FO)                                                   AGS   58
      XO=XL                                                                 AGS   59
      FO=F                                                                  AGS   60
      IF (X*UPDOWN.GE.0.) GO TO 60                                          AGS   61
C     SIGN OF X IS INCONSISTENT WITH UPDOWN                                 AGS   62
      IF (FIX) GO TO 90                                                     AGS   63
      IF (AE.GT.1.001) GO TO 90                                             AGS   64
C     TROUBLE IS DIAGNOSED AS DUE TO SONIC POINT TOO FAR DOWNSTREAM         AGS   65
C     OF GEOMETRIC THROAT.  TRY FIX -- RESET DISPLACEMENT THICKNESSES       AGS   66
C     TO THEIR VALUES AT THE THROAT.                                        AGS   67
      DO 80 L=1,2                                                           AGS   68
80    DELP(L)=DSTP(L)                                                       AGS   69
      FIX=.TRUE.                                                            AGS   70
      WRITE (6,140)                                                         AGS   71
      GO TO 40                                                              AGS   72
90    WRITE (6,150) AE,DEL,UPDOWN,X                                         AGS   73
      NAMELIST /AGDMP/ OMDST,RO,DSTP,DELP,YOZO,RTHCM,AEC,A,  AO,XO,F,FO,AGS   74
     1     Y,Z,ICOUNT                                                       AGS   75
      WRITE (6,AGDMP)                                                       AGS   76
      CALL DUMP (RNAME)                                                     AGS   77
      GO TO 130                                                             AGS   78
100   IF (AG.GT.1.) GO TO 110                                               AGS   79
      AG=1.                                                                 AGS   80
      X=0.                                                                  AGS   81
      GO TO 130                                                             AGS   82
110   CALL FINDX (AG,UPDOWN,X)                                              AGS   83
      GO TO 130                                                             AGS   84
120   AG=A                                                                  AGS   85
130   RETURN                                                                AGS   86
C                                                                           AGS   87
C                                                                           AGS   88
C                                                                           AGS   89
140   FORMAT (29H0+++++ FIX REQUIRED IN AGSOLN)                             AGS   90
150   FORMAT (30H0CONVERGENCE FAILURE IN AGSOLN,8X,3HAE=,1PE12.5,8X,4HDEAGS   91
     1L=,2E12.5,8X,7HUPDOWN=,0PF4.1,8X,2HX=,1PE12.5)                        AGS   92
      END                                                                   AGS   93-
```

```
      SUBROUTINE AXFIT                                                      AXF    1
      REAL ACOM(30),ELMENT(10),HP(20)                                       AXF    2
C     DETERMINES AXIAL POSITION DURING PERTURBATION SOLUTION                AXF    3
      DOUBLE PRECISION AA,CAPX,CDIJ                                         AXF    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                                 AXF    5
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,   AXF    6
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,   AXF    7
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,   AXF    8
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,   AXF    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,   AXF   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,   AXF   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,   AXF   12
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,   AXF   13
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  AXF   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),AXF   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),AXF   16
     2CLN1MC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),AXF   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),AXF   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),AXF   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),AXF   20
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) AXF   21
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          AXF   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                        AXF   23
      COMMON       IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,   AXF   24
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,   AXF   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,   AXF   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    MI,  NFIT,   AXF   27
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                         AXF   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) AXF   29
      COMMON ACOM,ELMENT,HP                                              AXF   30
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),   AXF   31
     1     SW,R0,JDIM,IPCINT                                            AXF   32
      IF (CT-CTMAX) 20,20,10                                             AXF   33
10    UPDOWN=-1.                                                         AXF   34
      GO TO 30                                                           AXF   35
20    UPDOWN=1.                                                          AXF   36
30    IF (ISW3B.NE.0.AND.IUPD.EQ.0) GO TO 40                             AXF   37
      CALL FINDX (AFNTS,UPDOWN,CX)                                       AXF   38
      GO TO 50                                                           AXF   39
40    CALL AGSOLN (AFNTS,DELBL,UPDOWN,AG,CX)                             AXF   40
50    RETURN                                                             AXF   41
      END                                                                AXF   42-
```

```
      SUBROUTINE AXSECT (LL,NN)                                          AXS    1
C     SUBROUTINE AXSECT ADDS DATA FOR UNSPECIFIED SPECIES PAIRS TO STEP  AXS    2
CC    LL OF THE EDITED CROSS SECTION DATA.  SPECIES TO BE CONSIDERED ARE AXS    3
C     GIVEN IN NUMERICAL ORDER IN THE FIRST NN ENTRIES OF THE ARRAY I.   AXS    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                               AXS    5
      COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,I(50)             AXS    6
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ         AXS    7
      IF (NN.LE.0) GO TO 40                                              AXS    8
      LQ1=NQ(LL)+1                                                       AXS    9
      N=0                                                                AXS   10
      DO 20 IV=1,NN                                                      AXS   11
      II=I(IV)                                                           AXS   12
      DO 20 JV=IV,NN                                                     AXS   13
      JJ=I(JV)                                                           AXS   14
      IF (Q(1,II,JJ).NE.0.) GO TO 20                                     AXS   15
      Q(1,II,JJ)=1.                                                      AXS   16
      N=N+1                                                              AXS   17
      LQ=400                                                             AXS   18
      DO 10 L=LQ1,399                                                    AXS   19
      LQ=LQ-1                                                            AXS   20
      IQ(LQ+1)=IQ(LQ)                                                    AXS   21
10    JQ(LQ+1)=JQ(LQ)                                                    AXS   22
      IQ(LQ1)=II                                                         AXS   23
      JQ(LQ1)=JJ                                                         AXS   24
20    CONTINUE                                                           AXS   25
      DO 30 L=LL,NKQ                                                     AXS   26
30    NQ(L)=NQ(L)+N                                                      AXS   27
40    RETURN                                                             AXS   28
      END                                                                AXS   29-
```

```
      SUBROUTINE BLAYER (FINAL)                                          BLA    1
C     LAMINAR BOUNDARY LAYER ROUTINE                                     BLA    2
      LOGICAL FINAL,OLDMF,TRCALC,COUPLD,MODLPT,ERR                       BLA    3
      REAL ACOM(30),ELMENT(10),HP(20)                                    BLA    4
      DOUBLE PRECISION AA,CAPX,CDIJ                                      BLA    5
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                              BLA    6
      COMMON       A, AFNTS,   AFNX, AMACH,     AR,   ARBA,   ARBB, BZERO,  BLA    7
     1       C,   CARB,     CH,    CHA,   CLNT,     CM,    CMA,    CRA,    CRP,  BLA    8
     2   CRRB,    CRS,   CSTA,     CT,   CTAP,    CTB,    CTC,  CTMAX,  CTMXX,  BLA    9
     3    CTP,   CTPL,    CTT,     CX,    CXB,  CXMAX,DATEST,DBTEST,  DELT1,   BLA   10
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,    ENT,   FLUX,  HDELX,    PCT,    BLA   11
     5PCTEST,   PRES,  PRESA, PRESB,PRESTH,   PRHC,   RHAP,    RHO,   RHOB,   BLA   12
     6RHOBAR,   RHOC,   RHOP,  RHPL,  RHTH,ROBARA,ROBARP,   SCPG,    SDT,    BLA   13
     7    SEN,   SHPG,     SC,    SL,   SL64,     SM,    SU,    SU2,   SUMG,   BLA   14
     8   TEST,  TESTB,TPRINT, TSTOP,     UP,     ZP,    ZPA                  BLA   15
      COMMON          BE(64),    BET(20),   BLBK(31),    CAI(64),CAPXTH(20),BLA   16
     1   CCPJ(20),  CEACT(64),    CGI(20),   CGMU(20),    CHI(64),  CHII(20),BLA   17
     2CLN1MC(64),  CLNPI(64),    CMW(20),   ETAI(64),   ETAJ(20),    GJA(20),BLA   18
     3    GJB(20),PERTGJ(20),    PGJ(20),     PI(64),  PICHI(64),     QM(20),BLA   19
     4     QQ(64),    SAJ(20),  SDCHI(64),   SENT(20),    SHJ(20),   SHJA(20),BLA   20
     5   SKIL(20),     SS(20),     TB(30),    TFA(20),    TFB(20),    TFC(20),BLA   21
     6    TFD(20),    TFE(20),    TFK(20),  THEV(20), XMJAT(20),   XNUI(64)  BLA   22
      COMMON          BETA(64,20),    ELJ(10,20),   GELJ(10,20),             BLA   23
     1 XNUIJ(64,20),XNUIJP(64,20)                                        BLA   24
      COMMON       IC,     IM,   INEQ,  INEQV,     IP,   IRUN,    ISC,  ISCP1,  BLA   25
     1   ISMC,ISMCNR,    ISP,    ISS,  ISSNR,  ISSP1,  ISSP2,  ISSP3,  ISSP4,  BLA   26
     2  ISW1A,  ISW1B,  ISW2A,  ISW2B,  ISW3A,  ISW3B,  ISW4A,  ISW4B,  ISW5A,  BLA   27
     3  ISW5B,  ISW6A,  ISW6B,   IUPD,  IZERO,    JJK,     LC,     M1,   NFIT,  BLA   28
     4    NIT,    NNN,    NNS,    NOS,    NQT, NTEST                       BLA   29
      COMMON    IGJ(20),    IGM(20),    ITB(5),    KUR(64,20),   LPIJ(20,10) BLA   30
      COMMON ACOM,ELMENT,HP                                               BLA   31
      DIMENSION GJ(20)                                                    BLA   32
      EQUIVALENCE (GJ(1),CAPX(1))                                         BLA   33
      COMMON /TNERK/ SDTE,CTEB,DCHA,CHB,SDCHA,DQMAX,IFAIL                 BLA   34
      COMMON /TEMPRY/ SAVEC(20)                                           BLA   35
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),    BLA   36
     1      SW,RO,JDIM,IPCINT                                            BLA   37
      COMMON /TRPROP/ VISC,PR,SIGMA,FLEWIS                               BLA   38
      COMMON /ERROR/ ERR                                                  BLA   39
      DIMENSION TRP(4),TRPSV(4)                                           BLA   40
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),           BLA   41
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                  BLA   42
      COMMON /OUTPUT/ FVOUT(35),GJMF(20)                                  BLA   43
      EQUIVALENCE (T,CT),(P,PRES),(RH,RHO),(U,SU),(AE,AFNX),(X,CX),       BLA   44
     1      (H,CH)                                                       BLA   45
      DIMENSION ORDIN(2),DLM(2),COSB(2),LPRIME(2),HTRP1(2)                BLA   46
      DATA LPRIME /2,1/                                                   BLA   47
      COMMON /BLNE/ XI(2),ORDINP(2),DELBLP(2),DDBLP(2),AMPP,AMP,          BLA   48
     1      COUPLD,ISMD                                                  BLA   49
      DIMENSION XTEST(20),XSAVE(20)                                       BLA   50
      DIMENSION YZ(2),DYZDX(2),HTR(2)                                     BLA   51
      EQUIVALENCE (TRP(1),VISC)                                           BLA   52
      COMMON /BLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOT(2),        BLA   53
     1      TAUW(2)                                                      BLA   54
      COMMON /NEQ/ OMDST(2),DDELBL(2)                                     BLA   55
```

```
      COMMON /SS/ CAS,US                                                BLA   56
      COMMON /SWITCH/ XPB,DXPB,W                                        BLA   57
      EQUIVALENCE (XP,XPB), (DXP,DXPB)                                  BLA   58
      COMMON /DRV/ SHJDGJ                                               BLA   59
      COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE                               BLA   59A
      COMMON /BLRAD/ Y0Z0                                               BLA   60
      COMMON /MODP2/ MODLPT                                             BLA   61
      COMMON /CCLDSP/ CGMW,QPJ(10),IJCS(10),NCS                         BLA   62
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),       BLA   63
     1     KTR(25),ICH(20),IPA(25)                                      BLA   64
      DATA ONE /1./, ZERO /0./                                          BLA   65
      NAMELIST /DMP/ IPOINT,X,T,P,RH,U,AM,H,AE,AG,AJ,UPRIME,AS,DX,HE,HR,BLA   66
     1     HREF,HW,TE,TREF,COSB,DADX,YZ,DYZDX,VISC,VISCR,VISCW,         BLA   67
     2     PR,PRREF,PRW,ALPR,     ORDIN,XI,OMEGA,SIGH,DMDX,DLM,US,R0,   BLA   68
     3     SW,NBL,ITYPE,RER,BLINT,ENERIM,SN,XSN,THETA1,THETA,HTR,       BLA   69
     4     DELBL,DDELBL,XL,RAF,TAUW,QWDOT,XTEST,GAMMAE,WW,AVCON,B,EK1,  BLA   70
     5     OLDMF,TRCALC,TEP,TREFP,RMR,F0,F2                             BLA   71
C     DELBL=DISPLACEMENT THICKNESS                                      BLA   72
C     THETA=MOMENTUM THICKNESS                                          BLA   73
C     THETA1=MOMENTUM THICKNESS FOR TWO-DIMENSIONAL FLOW                BLA   74
C     THE BOUNDARY LAYER THICKNESSES DELBL, THETA, THETA1 ARE ALL       BLA   75
C          NONDIMENSIONALIZED USING THE GEOMETRIC THROAT RADIUS R0. THEYBLA   76
C          ARE CONVERTED TO INCH UNITS FOR OUTPUT, IN SUBROUTINE OUT1.  BLA   77
      IF (X.GT.XP.OR.IPOINT.EQ.1) GO TO 10                              BLA   78
      IF (X.EQ.CXB) GO TO 550                                           BLA   79
      WRITE (6,560) X,XP                                                BLA   80
      WRITE (6,DMP)                                                     BLA   81
      GO TO 550                                                         BLA   82
10    DO 540 L=1,NPRFLS                                                 BLA   83
      IF (IPOINT.NE.1.OR.L.EQ.2) GO TO 90                               BLA   84
      SWP1=SW+1.                                                        BLA   85
      B=2.14+1.28*SWP1+0.93*(SWP1**2)**2                                BLA   86
      BM1=B-1.                                                          BLA   87
C     EK1 AND B ARE THE COEFFICIENTS IN THE COHEN-RESHOTKO LINEAR FIT TOBLA   88
C      THE MOMENTUM PARAMETER (CAP. N) AS A FUNCTION OF THE CORRELATION BLA   89
C          NUMBER SN (SMALL N).  THE FORMULA IS  N=EK1+B*SN.             BLA   90
      SWFAC=SWP1*(7.+3.4*SWP1)                                          BLA   91
      SWFAC2=1.+4.5*SWP1**0.9                                           BLA   92
      HCON=CRP/CMA                                                      BLA   93
      HW=CPWALL*TWALL                                                   BLA   94
      TE=0.                                                             BLA   95
      TREFP=0.                                                          BLA   96
      H0=CHA*HCON                                                       BLA   97
      DO 20 I=1,ISS                                                     BLA   98
      XSAVE(I)=SAVEC(I)                                                 BLA   99
20    XTEST(I)=-1.                                                      BLA  100
      DO 40 I=1,ISS                                                     BLA  101
      DO 30 K=1,NCS                                                     BLA  102
      IF (IJCS(K).NE.I) GO TO 30                                        BLA  103
      SAVEC(I)=QPJ(K)                                                   BLA  104
      GO TO 40                                                          BLA  105
30    CONTINUE                                                          BLA  106
      SAVEC(I)=0.                                                       BLA  107
40    CONTINUE                                                          BLA  108
      CALL TRANSP (TWALL,PRESA)                                         BLA  109
```

```
      PRW=PR                                                            BLA 110
      VISCW=VISC                                                        BLA 111
      RWMW=VISCW*PRESA*CGMW/82.05/TWALL/RHAP                            BLA 112
      DO 50 I=1,ISS                                                     BLA 113
50    SAVEC(I)=XSAVE(I)                                                 BLA 114
      IF (NPRFLS.EQ.1) GO TO 60                                         BLA 115
      YDZ0=RTHCM(1)*RTHCM(2)                                            BLA 116
      ITYPE=3                                                           BLA 117
      GO TO 80                                                          BLA 118
60    IF (JDIM.EQ.0) GO TO 70                                           BLA 119
      ITYPE=2                                                           BLA 120
      GO TO 80                                                          BLA 121
70    ITYPE=1                                                           BLA 122
C     ITYPE IS A FLAG FOR TYPE OF NOZZLE GEOMETRY                       BLA 123
C          ITYPE=1   TWO-DIMENSIONAL                                    BLA 124
C          ITYPE=2   AXISYMMETRIC                                       BLA 125
C          ITYPE=3   RECTANGULAR CHANNEL                                BLA 126
80    AVCON=W/RO                                                        BLA 127
C     RO IS SET IN MAIN                                                 BLA 128
      COUPLD=.FALSE.                                                    BLA 129
      ISMD=30                                                           BLA 130
      DDELBL(1)=0.                                                      BLA 131
      DDELBL(2)=0.                                                      BLA 132
90    IF (L.EQ.2) GO TO 100                                             BLA 133
      IF (NT.EQ.2) HO=CHA*HCON                                          BLA 134
      AS=SQRT(CAS*CT)                                                   BLA 135
100   IF (FINAL) GO TO 110                                              BLA 136
      B_INTI=BLINT(L)                                                   BLA 137
      XII=XI(L)                                                         BLA 138
110   IF (L.EQ.2) GO TO 220                                             BLA 139
      TE=CTAP*T                                                         BLA 140
      DO 120 I=1,ISS                                                    BLA 141
      IF (ABS(XTEST(I)-SAVEC(I)).GT.0.01) GO TO 130                     BLA 142
120   CONTINUE                                                          BLA 143
      OLDMF=.TRUE.                                                      BLA 144
      TRCALC=.FALSE.                                                    BLA 145
      GO TO 150                                                         BLA 146
130   OLDMF=.FALSE.                                                     BLA 147
      DO 140 I=1,ISS                                                    BLA 148
140   XTEST(I)=SAVEC(I)                                                 BLA 149
      GO TO 160                                                         BLA 150
150   IF (ABS(TE-TEP)/TE.LT.0.005) GO TO 180                            BLA 151
160   CALL TRANSP (TE,P*PRESA)                                          BLA 152
      TEP=TE                                                            BLA 153
      IF (ERR) RETURN                                                   BLA 154
      IF (IPOINT.LE.1) PRREF=PR                                         BLA 155
      DO 170 I=1,4                                                      BLA 156
170   TRPSV(I)=TRP(I)                                                   BLA 157
180   HE=HCON*H                                                         BLA 158
      PRPWR=PRREF**0.56                                                 BLA 159
      HR=HE+PRPWR*(HO-HE)                                               BLA 160
C     HR=RECOVERY ENTHALPY                                              BLA 161
      HREF=0.5*(HE+HW)+0.22*(HR-HE)                                     BLA 162
      TREF=CTAP*HREF/HO                                                 BLA 163
      CALL GEOMAP (X,AG,DADX)                                           BLA 164
```

```
      IF (ERR) RETURN                                                    BLA 165
C     AG=GEOMETRIC AREA RATIO (AREA DIVIDED BY AREA AT THROAT)           BLA 166
C     AE=EFFECTIVE AREA RATIO ALLOWING FOR BOUNDARY LAYER DISPLACEMENT   BLA 167
C           THICKNESS                                                    BLA 168
      GO TO (190,200,210), ITYPE                                         BLA 169
  190 AJ=1.                                                              BLA 170
      COSB(1)=1./SQRT(1.+(R0*DADX)**2)                                   BLA 171
      GO TO 230                                                          BLA 172
  200 AJ=AG                                                              BLA 173
      COSB(1)=1./SQRT(1.+0.25*(R0*DADX)**2/AG)                           BLA 174
      GO TO 230                                                          BLA 175
  210 CALL GMAR3 (X,DYZDX(1),DYZDX(2),YZ(1),YZ(2))                       BLA 176
  220 LPR=LPRIME(L)                                                      BLA 177
      AJ=(YZ(LPR))**2                                                    BLA 178
      COSB(L)=1./SQRT(1.+(DYZDX(L))**2)                                  BLA 179
C     COSB=COSINE OF THE ANGLE BETWEEN THE NOZZLE WALL AND THE AXIS OR   BLA 180
C           CENTER PLANE.                                                BLA 181
  230 IF (L.EQ.2) GO TO 270                                              BLA 182
      UPRIME=U*US                                                        BLA 183
      IF (OLDMF.AND.ABS(TREF-TREFP)/TREF.LT.0.005) GO TO 240             BLA 184
      CALL TRANSP (TREF,P*PRESA)                                         BLA 185
      IF (ERR) RETURN                                                    BLA 186
      TREFP=TREF                                                         BLA 187
      TRCALC=.TRUE.                                                      BLA 188
      PRREF=PR                                                           BLA 189
      VISCR=VISC                                                         BLA 190
  240 IF (.NOT.TRCALC) GO TO 250                                         BLA 191
      VR=VISCR/VISCW                                                     BLA 192
      TR=TREF/TWALL                                                      BLA 193
      OMEGA=ALOG(VR)/ALOG(TR)                                            BLA 194
      IF (IPOINT.NE.1) GO TO 250                                         BLA 195
      EK1=0.38-0.76*PRREF*(1.-OMEGA)*EXP(-6.67*SWP1)                     BLA 196
      IF (EK1.GE.0.01) GO TO 250                                         BLA 197
      WRITE (6,580)                                                      BLA 198
      WRITE (6,DMP)                                                      BLA 199
      EK1=0.01                                                           BLA 200
  250 DO 260 I=1,4                                                       BLA 201
  260 TRP(I)=TRPSV(I)                                                    BLA 202
      AM=UPRIME/AS                                                       BLA 203
  270 ORDIN(L)=AJ*AM**BM1*AS*RH*T                                        BLA 204
      IF (ITYPE.EQ.2) R=SQRT(AG)                                         BLA 205
C     R=RATIO OF LOCAL NOZZLE RADIUS TO THROAT RADIUS.                   BLA 206
      IF (IPOINT.NE.1) GO TO 290                                         BLA 207
C     FIRST POINT                                                        BLA 208
      DX=(X-XZERO)/10.                                                   BLA 209
      DXSQ=DX**2                                                         BLA 210
      XX=XZERO                                                           BLA 211
      XI(L)=0.                                                           BLA 212
      BLINT(L)=0.                                                        BLA 213
      CC=AM*AG                                                           BLA 214
      CALL RADIUS (ITYPE,XZERO,R1,AG1,AGJ,L)                             BLA 215
      AM1=CC/AG1                                                         BLA 216
      ORD1=AGJ*AM1**BM1                                                  BLA 217
      DO 280 I=1,10                                                      BLA 218
      XX=XX+DX                                                           BLA 219
```

```
      CALL RADIUS (ITYPE,XX,R2,AG1,AGJ,L)                                BLA 220
      DXI=SQRT(DXSQ+(R2-R1)**2)                                          BLA 221
      XI(L)=XI(L)+DXI                                                    BLA 222
      AM1=CC/AG1                                                         BLA 223
      ORD2=AGJ*AM1**BM1                                                  BLA 224
      BLINT(L)=BLINT(L)+0.5*(ORD1+ORD2)*DXI                              BLA 225
      ORD1=ORD2                                                          BLA 226
280   R1=R2                                                              BLA 227
      BLINT(L)=BLINT(L)*AS                                               BLA 228
      DLM(L)=-DADX/AG                                                    BLA 229
C     X IS THE AXIAL COORDINATE DOWN THE NOZZLE, ZERO AT THE GEOMETRIC   BLA 230
C          THROAT.                                                       BLA 231
C     XI IS THE STREAMWISE COORDINATE IN THE BOUNDARY LAYER, ZERO AT THEBLA 232
C          NOZZLE ENTRANCE (WHERE X=XZERO).                              BLA 233
      GO TO 370                                                          BLA 234
290   DX=X-XP                                                            BLA 235
      IF (INEQ.EQ.0) GO TO 320                                           BLA 236
C     COMPUTE DERIVATIVES FROM RATE DATA                                 BLA 237
      IF (NT.EQ.1) GO TO 300                                             BLA 238
      DUDX=GJ(1)*CCPJ(1)*DTE+SCPGH*DT                                    BLA 239
      GO TO 310                                                          BLA 240
300   DUDX=SCPG*DT                                                       BLA 241
310   DUDX=-CMA/U*(DUDX+SHJDGJ)                                          BLA 242
      IF (NT.EQ.2) DUDX=DUDX+DCHA/SU                                     BLA 243
      DLM(L)=DUDX/U-0.5*DT/T                                             BLA 244
      GO TO 360                                                          BLA 245
320   IF (IPOINT.GT.2) GO TO 340                                         BLA 246
330   DMDX=(AM-AMP)/DX                                                   BLA 247
      GO TO 350                                                          BLA 248
C     CALCULATE DERIVATIVE FROM QUADRATIC FIT                            BLA 249
340   IF (DX.LE.0.) GO TO 370                                            BLA 250
      DX2=DX+DXP                                                         BLA 251
      XPXP=X+XP                                                          BLA 252
      DMDX=(AM-AMP)/DX                                                   BLA 253
      C2=(DMDX-(AM-AMPP)/DX2)/DXP                                        BLA 254
      C1=DMDX-C2*XPXP                                                    BLA 255
      DMDX=C1+2.*C2*X                                                    BLA 256
      IF (DMDX.LT.0.) GO TO 330                                          BLA 257
350   DLM(L)=DMDX/AM                                                     BLA 258
C     TRAPEZOIDAL RULE INTEGRATION                                       BLA 259
360   DXI=DX/COSB(L)                                                     BLA 260
      XI(L)=XI(L)+DXI                                                    BLA 261
      BLINT(L)=BLINT(L)+0.5*(ORDIN(L)+ORDINP(L))*DXI                     BLA 262
370   DLM(L)=DLM(L)*COSB(L)                                              BLA 263
      RER=RH*RHAP*UPRIME*R0/VISC                                         BLA 264
C     RER=REYNOLDS NUMBER BASED ON THROAT RADIUS AND FREE-STREAM         BLA 265
C          CONDITIONS.                                                   BLA 266
      ENPRIM=BLINT(L)/ORDIN(L)/R0*EK1                                    BLA 267
      RMR=RWMW*P/VISC/RH                                                 BLA 268
      THETA1=SQRT(ENPRIM/RER*RMR)                                        BLA 269
      GO TO (380,390,380), ITYPE                                         BLA 270
380   THETA(L)=THETA1                                                    BLA 271
      GO TO 400                                                          BLA 272
C     TRANSVERSE CURVATURE CORRECTION FOR AXISYMMETRIC NOZZLES           BLA 273
390   SQT=SQRT(1.-2.*COSB(L)*THETA1/R)                                   BLA 274
```

```
      THETA(L)=R/COSB(L)*(1.-SQT)                                          BLA 275
400   IF (.NOT.FINAL.AND.IPOINT.NE.1) GO TO 430                            BLA 276
      SN(L)=-ENPRIM*RO*DLM(L)                                              BLA 277
      IF (IPOINT.NE.1) GO TO 410                                           BLA 278
      XSN(L)=SN(L)                                                         BLA 279
      GO TO 420                                                            BLA 280
410   WW=EXP(-AVCON*DX/AG)                                                 BLA 281
      XSN(L)=(1.-WW)*SN(L)+WW*XSN(L)                                       BLA 282
420   XSNV=AMIN1(XSN(L),ZERO)                                              BLA 283
      ALPR=ALOG(PRREF)                                                     BLA 284
      GAMMAE=AS**2/(8.3144E7/CM*TE)                                        BLA 285
      IF (GAMMAE.LE.1.) GAMMAE=1.01                                        BLA 286
      SIGH=1./(1.+2./(GAMMAE-1.)/AM**2)                                    BLA 287
      HTR(L)=0.75*(EXP(4.6*XSNV)*(1.+(3.-15.6*XSNV)*SWP1)-1.)              BLA 288
      HTRP1(L)=(HTR(L)+1.)*(SWP1**(-0.4*ALPR)-0.47*SIGH*ALPR)              BLA 289
      HTR(L)=HTRP1(L)-1.                                                   BLA 290
430   HFP1=HTRP1(L)/T                                                      BLA 291
      HF=HFP1-1.                                                           BLA 292
      DELBL(L)=HF*THETA(L)                                                 BLA 293
      IF (IPOINT.LE.1.OR..NOT.FINAL) GO TO 470                             BLA 294
      IF (COUPLD) GO TO 440                                                BLA 295
      IF (IUPD.EQ.1) GO TO 470                                             BLA 296
      COUPLD=.TRUE.                                                        BLA 297
440   DDELBL(L)=(DELBL(L)-DELBLP(L))/(X-XP)                                BLA 298
      IF (ISMD.EQ.1.OR.L.EQ.2) GO TO 450                                   BLA 299
      WD=1./ISMD                                                           BLA 300
      GO TO 460                                                            BLA 301
450   IF (L.EQ.2) GO TO 460                                                BLA 302
      WD=0.5*AMIN1(ONE,(DX/DXO)**2)                                        BLA 303
460   DDELBL(L)=WD*DDELBL(L)+(1.-WD)*DDBLP(L)                              BLA 304
      IF (ISMD.EQ.1.OR.L.LT.NPRFLS) GO TO 470                              BLA 305
      ISMD=ISMD-1                                                          BLA 306
470   CONTINUE                                                             BLA 307
      IF (FINAL) GO TO 480                                                 BLA 308
      BLINT(L)=BLINTI                                                      BLA 309
      XI(L)=XII                                                            BLA 310
      GO TO 540                                                            BLA 311
480   DELBLP(L)=DELBL(L)                                                   BLA 312
      ORDINP(L)=ORDIN(L)                                                   BLA 313
      OM1=AMAX1(1.-OMEGA,ZERO)                                             BLA 314
      F0=EXP(-0.427*(SWP1**(-OM1)-1.))                                     BLA 315
      AX=15.5*EXP(-29.*SWP1)+0.67*EXP(4.37*SWP1)                           BLA 316
      BX=0.7+0.47*SWP1                                                     BLA 317
      F1=0.76*OM1**1.2*(1.-SQRT(1.-SIGH))                                  BLA 318
      IF (SN(L).LT.-1.E-20) F1=F1/(1.+AX*(-SN(L))**BX)                     BLA 319
      IF (F0) 490,490,500                                                  BLA 320
490   F3=0.                                                                BLA 321
      GO TO 510                                                            BLA 322
500   F3=EXP(3.5*AMIN1(SN(L),ZERO)/F0**1.57)                               BLA 323
510   F2=F0+0.45*OM1*(1.-F3)+F1                                            BLA 324
      IF (SN(L).LE.-1.E-20) GO TO 520                                      BLA 325
      XL=0.2205                                                            BLA 326
      GO TO 530                                                            BLA 327
520   XL=0.2205+SWFAC2/(-2.7/SN(L)+5.06/(-SN(L))**0.224)                   BLA 328
530   XL=F2*XL                                                             BLA 329
```

```
      TAUW(L)=0.00209*XL*RMR*VISC*UPRIME/THETA(L)/R0                    BLA 330
      RA=SWP1**(-0.86*OMEGA)*(1./(6.2+2.35/SWP1**0.86)+(1.-PRREF)**2*(0.BLA 331
     1185*(1.-2.*SWP1)+0.29*SWP1**0.38*SIGH/OMEGA))                    BLA 332
      RB=0.94-1.31*SWP1**0.486-ALPR*(0.62*SWP1+(3.5*OM1-0.49-3.61*SWP1)*BLA 333
     1SIGH)                                                            BLA 334
      RAF=PRREF**(-0.35)*(1.-SN(L)/(RA-RB*SN(L)))                       BLA 335
      QWDOT(L)=1760.*TAUW(L)*(HR-HW)/UPRIME/PRW/RAF                     BLA 336
      DDBLP(L)=DDELBL(L)                                                BLA 337
      IF (L.LT.NPRFLS) GO TO 540                                        BLA 338
      IPOINT=IPOINT+1                                                   BLA 339
      AMPP=AMP                                                          BLA 340
      AMP=AM                                                            BLA 341
      XP=X                                                              BLA 342
      DXP=DX                                                            BLA 343
      IF (.NOT.MODLPT) DX0=DX                                           BLA 344
      REPF=RER/R0*30.48                                                 BLA 345
  540 CONTINUE                                                          BLA 346
      IF (ISW4B.GT.0) WRITE (6,DMP)                                     BLA 347
      IF (ISW4B.LT.0) WRITE (6,570) X,IPOINT,ISMD,WD,(DDELBL(J),J=1,NPRFBLA 348
     1LS)                                                               BLA 349
  550 RETURN                                                            BLA 350
C                                                                       BLA 351
C                                                                       BLA 352
C                                                                       BLA 353
  560 FORMAT (36H0ZERO OR NEGATIVE STEP IN BLAYER, X=,1PE15.8,10X,3HXP=,BLA 354
     1E15.8)                                                            BLA 355
  570 FORMAT (12H BLAYER   X=,F10.5,5X,7HIPOINT=,I4,5X,5HISMD=,I2,5X,3HWBLA 356
     1D=,F6.4,5X,7HDDELBL=,2F15.6)                                      BLA 357
  580 FORMAT (56H0***** CALCULATED EK1 LESS THAN 0.01. EK1 RESET TO 0.01BLA 358
     1.)                                                                BLA 359
      END                                                               BLA 360-
```

```
      SUBROUTINE BLCALL (FINAL)                                                 BLC    1
C     CALLING ROUTINE FOR BOUNDARY LAYER IN NONEQUILIBRIUM CALCULATIONS         BLC    2
      LOGICAL FINAL                                                             BLC    3
      REAL ACOM(30),ELMENT(10),HP(20)                                           BLC    4
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                                      BLC    5
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                                     BLC    6
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,       BLC    7
     1       C,  CARB,    CH,   CHA,  CLNT,     CM,   CMA,   CRA,   CRP,      BLC    8
     2    CRRB,   CRS,  CSTA,    CT,  CTAP,    CTB,   CTC, CTMAX, CTMXX,      BLC    9
     3     CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,       BLC   10
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,         BLC   11
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHC,  RHAP,   RHO,  RHOB,         BLC   12
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,         BLC   13
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,       BLC   14
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                       BLC   15
      COMMON           BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),BLC   16
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),BLC   17
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),BLC   18
     3    GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),BLC   19
     4     QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),BLC   20
     5   SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),BLC   21
     6    TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),  XNUI(64) BLC   22
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),                BLC   23
     1 XNUIJ(64,20),XNUIJP(64,20)                                               BLC   24
      COMMON       IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,      BLC   25
     1   ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,       BLC   26
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,          BLC   27
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,        BLC   28
     4    NIT,   NNN,   NNS,   NQS,   NQT, NTEST                             BLC   29
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10)      BLC   30
      COMMON ACOM,ELMENT,HP                                                     BLC   31
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBBLC   32
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                        BLC   33
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),                      BLC   34
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),              BLC   35
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),                   BLC   36
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                              BLC   37
      COMMON /TEMPRY/ SAVEC(20)                                                 BLC   38
      IF (INEQ.EQ.0) GO TO 20                                                   BLC   39
      DO 10 J=1,ISS                                                             BLC   40
10    SAVEC(J)=GJ(J)*CM                                                         BLC   41
      GO TO 40                                                                  BLC   42
20    CT=CT+PCT                                                                 BLC   43
      RHO=RHO+PRHO                                                              BLC   44
      DO 30 J=1,ISS                                                             BLC   45
30    SAVEC(J)=(GJ(J)+PERTGJ(J))*CM                                             BLC   46
40    IF (ISW3B.EQ.0) GO TO 50                                                  BLC   47
      CALL BLAYER (FINAL)                                                       BLC   48
50    IF (INEQ.NE.0) GO TO 60                                                   BLC   49
      CT=CT-PCT                                                                 BLC   50
      RHO=RHO-PRHO                                                              BLC   51
60    RETURN                                                                    BLC   52
      END                                                                       BLC   53
```

```
      SUBROUTINE BXSECT (L,MV,I,J)                                          BXS    1
C     SUBROUTINE BXSECT DETERMINES THE CORRESPONDENCE BETWEEN THE KQ ANDBXS    2
C     V ARRAYS AND SEARCHES THE V ARRAY FOR REFERENCES TO SPECIES PAIRS.BXS    3
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ            BXS    4
      COMMON /TRANS8/ NV(15),N                                              BXS    5
      IF (L.NE.0) GO TO 40                                                  BXS    6
      MV=1                                                                  BXS    7
      MADD=0                                                                BXS    8
10    L=L+1                                                                 BXS    9
      K=KQ(L)                                                               BXS   10
      IF (K.EQ.9) GO TO 30                                                  BXS   11
      IF (K.NE.12) GO TO 50                                                 BXS   12
      MADD=4                                                                BXS   13
20    MADD=MADD-2                                                           BXS   14
30    MM=MV+MADD                                                            BXS   15
      I=V(MM+1)                                                             BXS   16
      J=V(MM+2)                                                             BXS   17
      RETURN                                                                BXS   18
40    V(MM+1)=I                                                             BXS   19
      V(MM+2)=J                                                             BXS   20
      IF (MADD.NE.0) GO TO 20                                               BXS   21
50    MV=MV+NV(K)                                                           BXS   22
      IF (L.LT.NKQ) GO TO 10                                                BXS   23
      L=0                                                                   BXS   24
      RETURN                                                                BXS   25
      END                                                                   BXS   26
```

```
      SUBROUTINE COMM                                                   COM    1
      LOGICAL ERR,FAILED,LOGC,NOREAC,RCALC                              COM    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   COM    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              COM    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             COM    5
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,  COM    6
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  COM    7
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  COM    8
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  COM    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  COM   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  COM   11
     6RHOBAR,  RHOC,  RHOP,  RHPL, RHTH,ROBARA,ROBARP,  SCPG,   SDT,   COM   12
     7   SEN,  SHPG,    SC,    SL,  SL54,    SM,    SU,   SU2,  SUMG,  COM   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  COM   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),COM   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),COM   16
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),COM   17
     3   GJJ(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),   QM(20),COM   18
     4    QQ(64),   SAJ(20), SDCHI(64),   SENT(20),   SHJ(20), SHJA(20),COM   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),COM   20
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),   XNUI(64) COM   21
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          COM   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       COM   23
      COMMON      IC,    IM,   INEQ, INFOV,    IP,  IRUN,   ISC, ISCP1,  COM   24
     1   ISMC,ISMCNR,   ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  COM   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    COM   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,   NFIT,  COM   27
     4    NIT,   NNN,   NNS,   NQS,   NQT, NTEST                       COM   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) COM   29
      COMMON ACOM,ELMENT,HP                                             COM   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBCOM   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                COM   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              COM   33
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       COM   34
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            COM   35
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       COM   36
      COMMON /ERROR/ ERR                                                COM   37
      COMMON /CERROR/ FAILED                                            COM   38
      COMMON /TNONEQ/ CTE,DTE,EPAR,EPAR(2,25),NT,ITR(25),KTF(25),       COM   39
     1      KTR(25),ICH(20),IPA(25)                                     COM   40
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         COM   41
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                 COM   42
      COMMON /TNDC/ XMJATD(20,2),CLNTD(2),CTD(2)                        COM   43
      COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE                               COM   44
      COMMON /GLIM2/ NOREAC(64)                                         COM   45
      DATA VIC1,VIC2 /1.094E24,1.1328E-16/, VNC /4.99E29/               COM   46
      DIMENSION ALGJ(20)                                                COM   47
      EQUIVALENCE (ALGJ(1),AA(1,24))                                    COM   48
      NAMELIST /COMDMP/ CX,AFNX,DLOGA,RHO,DLOGR,SUMG,SHPG,RHAP,CRP,     COM   49
     1      SU2,CSTA,Z1,Z2F,  CLNTD,GJ,CHI,PI,XMJATD,CTD,SUMGH,SCPGH,   COM   50
     2      QDPR,QDPE,RKR,RKF,GELAS,CLNPI,CLNIMC                        COM   51
      DATA ICOUNT /0/, ONE /1./                                         COM   52
      DATA ICYCLE /90000000/                                            COM   53
      ICOUNT=ICOUNT+1                                                   COM   54
      IF (ISW5B.LT.0) ICYCLE=-ISW5B                                     COM   55
```

```
      IF (ICYCLE.EQ.90000000) GO TO 30                                  COM   56
      IF (MOD(ICOUNT,ICYCLE)) 10,20,10                                  COM   57
10    I WSB=0                                                           COM   58
      GO TO 30                                                          COM   59
20    ISWSB=1                                                           COM   60
      WRITE (6,550) ICOUNT                                              COM   61
30    CONTINUE                                                          COM   62
      FAILED=.FALSE.                                                    COM   63
      SUMG=0.                                                           COM   64
      SCPGH=0.                                                          COM   65
      QDPR=0.                                                           COM   66
      QDPE=0.                                                           COM   67
      DO 70 J=1,ISS                                                     COM   68
      IF (GJ(J)) 80,40,50                                               COM   69
40    ALGJ(J)=-1000.                                                    COM   70
      GO TO 60                                                          COM   71
50    ALGJ(J)=ALOG(GJ(J))                                               COM   72
60    IF (J.LT.NT) GO TO 70                                             COM   73
      SUMG=SUMG+GJ(J)                                                   COM   74
      SCPGH=SCPGH+GJ(J)*CCPJ(J)                                         COM   75
70    CONTINUE                                                          COM   76
      SUMGH=SUMG                                                        COM   77
      SCPG=SCPGH                                                        COM   78
      IF (NT.EQ.1) GO TO 100                                            COM   79
      SUMG=SUMG+GJ(1)                                                   COM   80
      SCPG=SCPG+GJ(1)*CCPJ(1)                                           COM   81
      TEP=CTE*CTAP                                                      COM   82
      GO TO 100                                                         COM   83
80    WRITE (6,560)                                                     COM   84
90    FAILED=.TRUE.                                                     COM   85
      RETURN                                                            COM   86
100   IF (INEQ) 110,150,110                                             COM   87
110   SEN=0.0                                                           COM   88
      SHPG=0.                                                           COM   89
      DO 130 J=1,ISS                                                    COM   90
      IF (GJ(J)) 80,130,120                                             COM   91
120   SEN=SEN+CJ(J)*(SENT(J)-ALGJ(J))                                   COM   92
130   SHPG=SHPG+GJ(J)*SHJ(J)                                            COM   93
      CM=1./SUMG                                                        COM   94
      CH=CMA*SHPG                                                       COM   95
      SU2=2.*(CHA-CH)                                                   COM   96
      IF (SU2.GE.0.) GO TO 140                                          COM   97
      WRITE (6,570)                                                     COM   98
      GO TO 90                                                          COM   99
140   SU=SQRT(SU2)                                                      COM  100
150   CALL GECM                                                         COM  101
      IF (ERR) RETURN                                                   COM  102
      RHOP=RHO*RHAP                                                     COM  103
      RHPL=ALOG(RHOP)                                                   COM  104
      Z1=ALOG(SL64/SU)+CSTA                                             COM  105
      Z2F=41.2918*CRP                                                   COM  106
      RCALC=.FALSE.                                                     COM  107
C     MAIN LOOP OVER THE REACTIONS                                      COM  108
      DO 460 I=1,ISR                                                    COM  109
      IF (NCREAC(I)) GO TO 450                                          COM  110
```

```
C       COMPUTATION OF PICHI                                                    COM 111
        LOGC=.FALSE.                                                            COM 112
        IF (NT.EQ.1) GO TO 160                                                  COM 113
        KF=KTF(I)                                                               COM 114
        KR=KTR(I)                                                               COM 115
        GO TO 170                                                               COM 116
160     KF=1                                                                    COM 117
        KR=1                                                                    COM 118
170     IF (KF-3) 180,190,200                                                   COM 119
180     CLKF=ETAI(I)*CLNTD(KF)+CAI(I)-CEACT(I)/(CTD(KF)*CRP)                    COM 120
        GO TO 240                                                               COM 121
190     CLKF=ETAI(I)*CLNTD(2)+CAI(I)+ALOG(1.-EXP(-CEACT(I)/(CTD(1)*CRP)))       COM 122
        GO TO 240                                                               COM 123
200     IF (RCALC) GO TO 220                                                    COM 124
        RCALC=.TRUE.                                                            COM 125
        IF (NPRFLS.EQ.2) GO TO 210                                              COM 126
        CALL GMAR (CX,R)                                                        COM 127
        GO TO 220                                                               COM 128
210     CALL GMAR2 (CX,Y,Z)                                                     COM 129
        R=1./SQRT(0.5*(1./Y**2+1./Z**2))                                        COM 130
220     IF (KF.EQ.4) GO TO 230                                                  COM 131
        CLKF=CAI(I)-0.5*ALOG(R)                                                 COM 132
        GO TO 240                                                               COM 133
230     IPAI=IPA(I)                                                             COM 134
        TAU=SPAR*RHOP*GJ(IPAI)*R                                                COM 135
        CLKF=ETAI(I)*CLNTD(2)-ALOG(AMAX1(ONE,TAU))+CAI(I)                       COM 136
240     CLNPI(I)=CLKF+Z1                                                        COM 137
        CLKR=CLKF                                                               COM 138
        CLN1MC(I)=0.0                                                           COM 139
        T6=0.0                                                                  COM 140
        T7=0.0                                                                  COM 141
        IF (KR.EQ.0) GO TO 250                                                  COM 142
        Z2=Z2F*CTD(KR)                                                          COM 143
        Z3=Z2*RHOP                                                              COM 144
        CLZ2=ALOG(Z2)                                                           COM 145
        CLZ3=ALOG(Z3)                                                           COM 146
250     DO 280 J=1,ISS                                                          COM 147
        IF (XNUIJ(I,J).EQ.0.0) GO TO 260                                        COM 148
        T6=T6+XNUIJ(I,J)*ALGJ(J)                                                COM 149
260     IF (KUR(I,J).EQ.0) GO TO 270                                            COM 150
        T7=T7+GJ(J)                                                             COM 151
270     IF (BETA(I,J).EQ.0..OR.KR.EQ.0) GO TO 280                               COM 152
        CLN1MC(I)=CLN1MC(I)+BETA(I,J)*(ALGJ(J)+CLZ3+XMJATD(J,KR))               COM 153
280     CONTINUE                                                                COM 154
        IF (KR.EQ.0) GO TO 300                                                  COM 155
        IF (CLN1MC(I).LT.20.) GO TO 290                                         COM 156
        LOGC=.TRUE.                                                             COM 157
        GO TO 310                                                               COM 158
290     CHI(I)=1.0-EXP(CLN1MC(I))                                               COM 159
        GO TO 310                                                               COM 160
300     CHI(I)=1.0                                                              COM 161
310     CLNPI(I)=CLNPI(I)+(XNUI(I)-1.0)*RHPL+T6                                 COM 162
        IF (QQ(I)) 320,350,320                                                  COM 163
320     IF (T7) 330,340,330                                                     COM 164
330     CLT8F=ALOG(RHOP*T7)                                                     COM 165
```

```
      GO TO 350                                                          COM 166
340   PI(I)=0.0                                                          COM 167
      CLTBF=-10000000.                                                   COM 168
      GO TO 370                                                          COM 169
350   CLTBF=0.                                                           COM 170
360   PI(I)=EXP(CLNPI(I)+CLTBF)                                          COM 171
370   IF (LOGC) GO TO 380                                                COM 172
      PICHI(I)=PI(I)*CHI(I)                                              COM 173
      GO TO 390                                                          COM 174
380   PICHI(I)=-EXP(CLN1MC(I)+CLNPI(I)+CLTBF)                            COM 175
390   IF (NT.EQ.1) GO TO 460                                             COM 176
C     COMPUTATION OF RADIATIVE LOSS QDPR AND ENERGY TRANSFER TO THE      COM 177
C          ELECTRON GAS QDPE                                             COM 178
      SUMF=0.                                                            COM 179
      RKF=EXP(CLKF)                                                      COM 180
      SUMR=0.                                                            COM 181
      CALL EPART (I,TEP,EF,ER,QF,QR)                                     COM 182
      DO 410 J=1,ISS                                                     COM 183
      IF (KR.EQ.0) GO TO 400                                             COM 184
      CLKR=CLKR+BETA(I,J)*(CLZ2+XMJATD(J,KR))                            COM 185
400   SUMF=SUMF+XNUIJ(I,J)*(RHPL+ALGJ(J))                                COM 186
410   SUMR=SUMR+XNUIJP(I,J)*(RHPL+ALGJ(J))                               COM 187
      IF (KR) 420,420,430                                                COM 188
420   RKR=0.                                                             COM 189
      GO TO 440                                                          COM 190
430   RKR=EXP(CLKR)                                                      COM 191
440   SUMF=EXP(SUMF+CLTBF)                                               COM 192
      SUMR=EXP(SUMR+CLTBF)                                               COM 193
      QDPR=QDPR+QF*RKF*SUMF-QR*RKR*SUMR                                  COM 194
      QDPE=QDPE+EF*RKF*SUMF-ER*RKR*SUMR                                  COM 195
      NAMELIST /CDP/ I,KF,KR,CLKF,CLKR,T6,T7,Z2,Z3,CLZ2,CLZ3,            COM 196
     1    CLTBF,RKF,RKR,EF,ER,QF,QR,SUMF,SUMR,QDPR,QDPE                  COM 197
      IF (INEQ.NE.0.AND.ISWSB.NE.0) WRITE (6,CDP)                        COM 198
      GO TO 460                                                          COM 199
450   PICHI(I)=0.                                                        COM 200
460   CONTINUE                                                           COM 201
      IF (INEQ) 470,540,470                                              COM 202
470   GO TO (480,490), NT                                                COM 203
480   PRES=CT*SUMG                                                       COM 204
      GO TO 530                                                          COM 205
490   PRES=CT*SUMGH+CTE*GJ(1)                                            COM 206
C     COMPUTATION OF ENERGY TRANSFER TO THE ELECTRON GAS BY ELASTIC      COM 207
C          COLLISIONS                                                    COM 208
      TEQ=TEP**3                                                         COM 209
      VIC=VIC1*RHOP/SQRT(TEQ)*ALOG(VIC2*TEQ/RHOP/GJ(1))                  COM 210
      QELAS=0.                                                           COM 211
      VNCC=VNC*SQRT(TEP)*RHOP*PIOMEG(TEP)                                COM 212
      DO 520 J=2,ISS                                                     COM 213
      ICHG=ICH(J)                                                        COM 214
      GO TO (500,510), ICHG                                              COM 215
500   VEJ=VNCC*GJ(J)                                                     COM 216
      GO TO 520                                                          COM 217
510   VEJ=VIC*GJ(J)                                                      COM 218
520   QELAS=QELAS+VEJ/CGI(J)                                             COM 219
      QELAS=3.*CRA*RHOP*CGI(1)*CTAP*(CT-CTE)*GJ(1)*QELAS                 COM 220
```

```
      QDPE=QDPE+QELAS                                                    COM 221
530   PRES=PRES*RHO*CMA/ROBARA                                           COM 222
      SEN=CRA*(SEN-ALOG(PRES*CM*PRESA)/CM)                               COM 223
540   CONTINUE                                                           COM 224
      IF (INEQ.NE.0.AND.ISW5B.NE.0) WRITE (6,COMDMP)                     COM 225
      RETURN                                                             COM 226
C                                                                        COM 227
CC                                                                       COM 228
C                                                                        COM 229
550   FORMAT (///11H ***ICOUNT=,I5)                                      COM 230
560   FORMAT (43HONEGATIVE CONCENTRATION ENCOUNTERED IN COMM)            COM 231
570   FORMAT (24HOSU2 LESS THAN 0 IN COMM)                               COM 232
      END                                                                COM 233-
```

```
      SUBROUTINE CXSECT (L,J)                                          CXS    1
C     SUBROUTINE CXSECT DETERMINES THE CORRESPONDENCE OF SPECIES PAIRS  CXS    2
C     BETWEEN THE MASTER SPECIES LIST AND THE TRANSPORT CALCULATIONS.  CXS    3
      COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,I(50)           CXS    4
      M=L-J                                                            CXS    5
      L=I(L)                                                           CXS    6
      J=I(J)                                                           CXS    7
      IF (ISIGN(M,L-J).EQ.M) GO TO 10                                  CXS    8
      M=L                                                              CXS    9
      L=J                                                              CXS   10
      J=M                                                              CXS   11
10    RETURN                                                           CXS   12
      END                                                              CXS   13-
```

```
      SUBROUTINE DERIVS                                                 DER    1
      LOGICAL ERR,FAILED                                                DER    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   DER    3
      DOUBLE PRECISION AA,CAPX,CDIJ                                     DER    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             DER    5
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  DER    6
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  DER    7
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  DER    8
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  DER    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  DER   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  DER   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  DER   12
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  DER   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  DER   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),DER   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),DER   16
     2CLN1MC(64), CLNEI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),DER   17
     3   GJB(20),PERTGJ(20),   PGJ(20),   PI(64),  PICHI(64),   QM(20),DER   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),DER   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),DER   20
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) DER   21
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),         DER   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       DER   23
      COMMON      IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,  DER   24
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  DER   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,  DER   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,  DER   27
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        DER   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) DER   29
      COMMON ACOM,ELMENT,HP                                             DER   30
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  DER   31
     1      SW,RO,JDIM,IPCINT                                           DER   32
      COMMON /ERROR/ ERR                                                DER   33
      COMMON /CERROR/ FAILED                                            DER   34
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         DER   35
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                 DER   36
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),       DER   37
     1      KTR(25),ICH(20),IPA(25)                                     DER   38
      COMMON /TNDC/ XMJATD(20,2),CLNTD(2),CTD(2)                        DER   39
      DIMENSION DELI1(2),DELI2(2),DELO1(2),DELO2(2)                     DER   40
      NAMELIST /DRVDMP/ ITRY,DELBL,DELI1,DELI2,DELO1,DELO2,DEN          DER   41
      ITRY=0                                                            DER   42
   10 IF (NT.EQ.1) GO TO 30                                             DER   43
      CTSAVE=CT                                                         DER   44
      CT=CTE                                                            DER   45
      CALL THERM                                                        DER   46
      CLNTD(2)=CLNT                                                     DER   47
      HJE=SHJ(1)                                                        DER   48
      CPJE=CCPJ(1)                                                      DER   49
      SENTE=SENT(1)                                                     DER   50
      DO 20 I=1,ISS                                                     DER   51
   20 XMJATD(I,2)=XMJAT(I)                                              DER   52
      CT=CTSAVE                                                         DER   53
   30 CALL THERM                                                        DER   54
      CLNTD(1)=CLNT                                                     DER   55
```

```
      CTD(1)=CT                                                          DER   56
      DO 40 I=1,ISS                                                      DER   57
   40 XMJATD(I,1)=XMJAT(I)                                               DER   58
      IF (NT.EQ.1) GO TO 50                                              DER   59
      SHJ(1)=HJE                                                         DER   60
      CCPJ(1)=CPJE                                                       DER   61
      SENT(1)=SENTE                                                      DER   62
      CTD(2)=CTE                                                         DER   63
   50 CALL COMM                                                          DER   64
      IF (ERR.OR.FAILED) RETURN                                          DER   65
      CALL EXACT                                                         DER   66
      IF (ERR) RETURN                                                    DER   67
      IF (DLCGA*CX.GE.-0.01) GO TO 60                                    DER   68
      FAILED=.TRUE.                                                      DER   69
      RETURN                                                             DER   70
   60 IF (ISW38.EQ.0.OR.IUPD.EQ.1) GO TO 140                             DER   71
      DO 70 K=1,NPRFLS                                                   DER   72
      DELI2(K)=DELI1(K)                                                  DER   73
   70 DELI1(K)=DELBL(K)                                                  DER   74
      CALL BLCALL (,FALSE.)                                              DER   75
      IF (ERR) RETURN                                                    DER   76
      IF (ABS(1.-DELBL(1)/DELI1(1)).GT.0.01) GO TO 80                    DER   77
      IF (NPRFLS.EQ.1) GO TO 130                                         DER   78
      IF (ABS(1.-DELBL(2)/DELI1(2)).LE.0.01) GO TO 130                   DER   79
   80 DO 90 K=1,NPRFLS                                                   DER   80
      DELO2(K)=DELO1(K)                                                  DER   81
   90 DELO1(K)=DELBL(K)                                                  DER   82
      ITRY=ITRY+1                                                        DER   83
      IF (ITRY.EQ.1) GO TO 10                                            DER   84
      IF (ITRY.GT.6) GO TO 110                                           DER   85
      DO 100 K=1,NPRFLS                                                  DER   86
      DEN=DELO2(K)-DELO1(K)-DELI2(K)+DELI1(K)                            DER   87
      IF (DEN.EQ.0.) GO TO 120                                           DER   88
  100 DELBL(K)=(DELI1(K)*DELO2(K)-DELO1(K)*DELI2(K))/DEN                 DER   89
      GO TO 10                                                           DER   90
  110 WRITE (6,150)                                                      DER   91
  120 WRITE (6,DRVDMP)                                                   DER   92
  130 CONTINUE                                                           DER   93
  140 RETURN                                                             DER   94
C                                                                        DER   95
C                                                                        DER   96
C                                                                        DER   97
  150 FORMAT (39H BOUNDARY LAYER ITERATION NOT CONVERGED)                DER   98
      END                                                                DER   99-
```

```
      SUBROUTINE DSMSCL (A,I,N)                                          DSM    1
      DOUBLE PRECISION A                                                 DSM    2
C     BASED ON SUBROUTINE SIMQ FROM IBM SCIENTIFIC SUBROUTINE PACKAGE    DSM    3
      DATA RNAME /6HDSMSOL/                                              DSM    4
      DIMENSION A(22,24)                                                 DSM    5
      .1=N+1                                                             DSM    6
      M1=I+1                                                             DSM    7
      L=0                                                                DSM    8
      DO 20 K=1,I                                                        DSM    9
      DO 10 J=1,I                                                        DSM   10
      LN=L/N                                                             DSM   11
      L=L+1                                                              DSM   12
      LM=L-N*LN                                                          DSM   13
      LN1=LN+1                                                           DSM   14
      A(LM,LN1)=A(J,K)                                                   DSM   15
      IF (K.EQ.1) A(J,N1)=A(J,M1)                                        DSM   16
10    CONTINUE                                                           DSM   17
20    CONTINUE                                                           DSM   18
      CALL SIMQ (A(1,1),A(1,N1),I,KS)                                    DSM   19
      IF (KS.EQ.0) GO TO 30                                              DSM   20
      WRITE (6,50)                                                       DSM   21
      CALL DUMP (RNAME)                                                  DSM   22
      RETURN                                                             DSM   23
30    DO 40 J=1,I                                                        DSM   24
40    A(J,M1)=A(J,N1)                                                    DSM   25
      RETURN                                                             DSM   26
C                                                                        DSM   27
CC                                                                       DSM   28
50    FORMAT (36H0MATRIX OF COEFFICIENTS IS SINGULAR.)                   DSM   29
      END                                                                DSM   30-
```

```
      SUBROUTINE DUMP (RNAME)                                          DUM    1
      LOGICAL ERR                                                      DUM    2
      COMMON /ERROR/ ERR                                               DUM    3
      ERR=.TRUE.                                                       DUM    4
      WRITE (6,10) RNAME                                               DUM    5
      RETURN                                                           DUM    6
C                                                                      DUM    7
C                                                                      DUM    8
10    FORMAT (1H1,24H DUMP ROUTINE CALLED BY ,A6)                      DUM    9
      END                                                              DUM   10-
```

```
      SUBROUTINE DUMPEX                                                 DPX    1
      LOGICAL DATAPE,FAILED,COUPLD,NOREAC,MODLPT                        DPX    2
      DOUBLE PRECISION AA,CAPX,CDIJ,AIN                                 DPX    3
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             DPX    4
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  DPX    5
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  DPX    6
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  DPX    7
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELTI,  DPX    8
     4 DELTP,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  DPX    9
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHC,  RHAP,   RHO,  RHOB,  DPX   10
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  DPX   11
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  DPX   12
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  DPX   13
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),DPX   14
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),DPX   15
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),DPX   16
     3   GJB(20),PERTGJ(20),   PGJ(20),     PI(64), PICHI(64),    QM(20),DPX   17
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),DPX   18
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),DPX   19
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),   XNUI(64) DPX   20
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          DPX   21
     1 XNUIJ(64,20),XNUIJP(64,20)                                       DPX   22
      COMMON       IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,  DPX   23
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,   DPX   24
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    DPX   25
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,   DPX   26
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        DPX   27
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) DPX   28
      COMMON /TEMPRY/ SAVEC(20)                                         DPX   29
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  DPX   30
     1     SW,RO,JDIM,IPCINT                                            DPX   31
      COMMON /TRPROP/ VISC,PR,SIGMA,FLEWIS                             DPX   32
      COMMON /NEQ/ OMDST(2),DDELBL(2)                                   DPX   33
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         DPX   34
     1     NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                  DPX   35
      COMMON /EQC/ ZCAP(20),ZSEN,ZCH,ZCM,ZRBP,ZRHO                      DPX   36
      COMMON /STAG/ TS,EPSLON,TSF,EPSF                                  DPX   37
      COMMON /MODP2/ MODLPT                                             DPX
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,   CATFAC,TMODEL,XMODP1,DXMODP,   DPX
     1     TPLATE,KDIM                                                  DPX
      COMMON /NEWMP/ FACMP,NMODPT                                       DPX
      COMMON /MODPT/ TSOJAM(20),TSAR(20),NTS,NBL                        DPX
      COMMON /CERROR/ FAILED                                            DPX
      COMMON /GLIM2/ NOREAC(64)                                         DPX
      COMMON /BLNF/ XI(2),ORDINP(2),DELBLP(2),DDBLP(2),AMPP,AMP,        DPX   45
     1     COUPLD,ISMD                                                  DPX   46
      COMMON /DRV/ SHJDGJ                                               DPX   47
      COMMON /BLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,CFDOT(2),      DPX   48
     1     TAUW(2)                                                      DPX   49
      COMMON /SWITCH/ XPB,DXPB,W                                        DPX   50
      COMMON /CONVRT/ CF(6)                                             DPX   51
      COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAPE     DPX   52
      COMMON /AEGEOM/ SQRTA,S1,S2                                       DPX   53
      COMMON /THRT/ RSA                                                 DPX   54
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),       DPX   55
```

```
     1      KTR(25),ICH(20),IPA(25)                                          DPX   56
      COMMON /TNDC/ XMJATD(20,2),CLNTD(2),CTD(2)                             DPX   57
      COMMON /TNERK/ SDTE,CTEB,QEMAX                                         DPX   58
      COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE                                    DPX   59
      COMMON /INGNE/ AIN(10,10)                                              DPX   60
      NAMELIST /DMP1/                                                        DPX   61
     1          A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,       DPX   62
     2      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,      DPX   63
     3   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,      DPX   64
     4    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,      DPX   65
     5 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,      DPX   66
     6PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,      DPX   67
     7RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,      DPX   68
     8    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,      DPX   69
     9   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                     DPX   70
      NAMELIST /DMP2/                                                        DPX   71
     1      BE,BET,BLBK,CAI,CAPX,CAPXTH,CCPJ,                               DPX   72
     2      CEACT,CGI,CGMU,CHI,CHII,CLN1MC,CLNFI,CMW,ETAI,ETAJ,GJA,         DPX   73
     3      GJB,PERTGJ,PGJ,PI,PICHI,QM,QQ,SAJ,SCCHI,SENT,SHJ,               DPX   74
     4     SHJA,SKIL,SS,TB,TFA,TFB,TFC,TFD,TFE,TFK,THEV,XMJAT,XNUI          DPX   75
      NAMELIST /DMP3/                                                        DPX   76
     1          IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,      DPX   77
     2  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,      DPX   78
     3 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,      DPX   79
     4 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,      DPX   80
     5   NIT,   NNN,   NNS,   NQS,   NQT, NTEST,IGJ,IGM,ITB                DPX   81
      NAMELIST /DMP4/ DELBL,BLINT,TWALL,CPWALL,VISROT,RO,                    DPX   82
     1      DIAM,SW,JDIM,IPOINT,VISC,SAVEC,XZERO,PRREF,HR,                  DPX   83
     2      OMDST,DDELBL,ZCAP,ZSEN,ZCH,ZCM,ZRBP,ZRHO,TS,EPSLON,             DPX   84
     3      TSF,EPSF,THETA,SN,XSN,    REPF QWDOT,TAUW,SHJDGJ,               DPX   85
     4      XPB,DXPB,CF,XMP1,FSTAG,CATFAC,.MODEL,XMODP1,                    DPX   86
     5      KDIM,ATPI,PARAM,RTHCM,NSECT,ISHAPE,PR,SIGMA,                    DPX   87
     6      FLEXIS,FACMP,NMODPT,TSDIAM,NTS,TSAR,NPROFL,NPRFLS,NBL,SQRTA,    DPX   88
     7      S1,S2,RSA,W,SUMGH,SCPGH,QDPR,QDPE,SDTE,CTEB,CTE,DTE,BPAR,       DPX   89
     8      EPAR,NT,ITR,KTF,KTR,ICH,IPA,XMJATD,CLNTD,CTD,QEMAX,AIN,         DPX   90
     9      FAILED,NOREAC,XI,ORDINP,DELBLP,DDBLP,AMPP,AMP,COUPLD,ISMD       DPX   91
      WRITE (6,DMP1)                                                         DPX   92
      WRITE (6,DMP2)                                                         DPX   93
      WRITE (6,DMP3)                                                         DPX   94
      WRITE (6,DMP4)                                                         DPX   95
      IF (.NOT.DATAPE) GO TO 20                                              DPX   96
      DO 10 I=1,NRCOUT                                                       DPX   97
10    BACKSPACE ITPOUT                                                       DPX   98
      WRITE (6,30) ITPOUT,NRCOUT                                             DPX   99
      NRCOUT=0                                                               DPX  100
20    WRITE (6,40)                                                           DPX  101
      RETURN                                                                 DPX  102
C                                                                            DPX  103
C                                                                            DPX  104
C                                                                            DPX  105
30    FORMAT (11H0******TAPE,I3,14H BACKSPACED BY,I4,41H RECORDS TO DELE     DPX  106
     1TE DATA FROM FAILED CASE.)                                             DPX  107
40    FORMAT (1H1)                                                           DPX  108
      END                                                                    DPX  109-
```

```
      SUBROUTINE ELCOND (SIGMA,SSIG)                                    ELC    1
C     SUBROUTINE ELCOND COMPUTES ELECTRICAL CONDUCTIVITY IN MHOS/CM.    ELC    2
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              ELC    3
      COMMON /TRANS3/ B(20,2),BR(20,2),A(2),X(20),DH(20)                ELC    4
      COMMON /TRANS5/ N,IELEC,ID1,ID2                                   ELC    5
      SIGMA=0.                                                          ELC    6
      IF (IELEC.EQ.0) GO TO 20                                          ELC    7
      DO 10 I=2,N                                                       ELC    8
10    SIGMA=SIGMA+X(I)*Q(1,1,I)                                         ELC    9
      SIGMA=SSIG*X(1)/SIGMA                                             ELC   10
20    RETURN                                                            ELC   11
      END                                                               ELC   12-
```

```
      SUBROUTINE E  VE (ET,IP)                                          ELT    1
C     TIMING ROUTI.                                                     ELT    2
C                                                                       ELT    3
C     ET=ELAPSED TIME FROM BEGINNING OF EXECUTION IN SECONDS            ELT    4
C     IP=CONTROL FOR PRINTING THE TIME                                  ELT    5
C          IP = 0       SUPPRESS PRINT                                 ELT    6
C                                                                       ELT    7
      LOGICAL CALLED                                                    ELT    8
      DATA CALLED /.FALSE./                                             ELT    9
      ETP=ETO                                                           ELT   10
      ETO=ET                                                            ELT   11
      IF (CALLED) GO TO 10                                              ELT   12
      ETO=0.                                                            ELT   13
      CALL RESET                                                        ELT   14
      CALLED=.TRUE.                                                     ELT   15
      ET=0.                                                             ELT   16
      GO TO 30                                                          ELT   17
10    CALL TIME (I)                                                     ELT   18
      ET=0.001*I                                                        ELT   19
      IF (IP.EQ.0) GO TO 20                                             ELT   20
      ETM=ET/60.                                                        ELT   21
      DET=ET-ETO                                                        ELT   22
      WRITE (6,40) ETM,DET                                              ELT   23
      GO TO 30                                                          ELT   24
20    ETO=ETP                                                           ELT   25
30    RETURN                                                            ELT   26
C                                                                       ELT   27
C                                                                       ELT   28
40    FORMAT (24H0**********ELAPSED TIME=,F6.2,28H MINUTES SINCE START OELT   29
     1F RUN,,F6.0,32H SECONDS SINCE LAST PRINTED TIME/)                 ELT   30
      END                                                               ELT   31-
```

```
      SUBROUTINE EPART (IR,TE,EF,ER,QF,QR)                                  EPA    1
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),           EPA    2
     1     KTR(25),ICH(20),IPA(25)                                          EPA    3
C     DEFINITIONS OF ARGUMENTS                                              EPA    4
C     INPUTS                                                                EPA    5
C           IR = INDE  OF REACTIONS                                         EPA    6
C           TE = ELEC RON TEMPERATURE (DEG. K)                              EPA    7
C                                                                           EPA    8
C     OUTPUTS                                                               EPA    9
C           EF = ENERGY GAINED BY THE ELECTRON GAS IN THE FORWARD           EPA   10
C                REACTION (CAL/MOLE)                                        EPA   11
C          -ER = ENERGY GAINED BY THE ELECTRON GAS IN THE REVERSE           EPA   12
C                REACTION (CAL/MOLE)                                        EPA   13
C           QF = ENERGY LOST BY RADIATION IN THE FORWARD REACTION (CAL/      EPA   14
C                MOLE)                                                      EPA   15
C          -QR = ENERGY LOST BY RADIATION IN THE REVERSE REACTION (CAL/      EPA   16
C                MOLE)                                                      EPA   17
C                                                                           EPA   18
      DATA THR/2.9808/, R0 /1.9872/                                         EPA   19
      IT=ITR(IR)                                                            EPA   20
      E0=EPAR(1,IR)                                                         EPA   21
      GO TO (10,20,50,20,80,90), IT                                         EPA   22
   10 EF=-EPAR(2,IR)*R0*TE                                                  EPA   23
      GO TO 30                                                              EPA   24
   20 EF=-THR*TE                                                            EPA   25
      IF (IT.EQ.4) GO TO 70                                                 EPA   26
   30 ER=0.                                                                 EPA   27
      QF=E0-EF                                                              EPA   28
   40 QR=0.                                                                 EPA   29
      GO TO 100                                                             EPA   30
   50 EF=0.                                                                 EPA   31
      ER=0.                                                                 EPA   32
   60 QF=0.                                                                 EPA   33
      GO TO 40                                                              EPA   34
   70 ER=EF                                                                 EPA   35
      GO TO 60                                                              EPA   36
   80 EF=E0                                                                 EPA   37
      GO TO 70                                                              EPA   38
   90 QF=E0                                                                 EPA   39
      EF=0.                                                                 EPA   40
      ER=0.                                                                 EPA   41
      GO TO 40                                                              EPA   42
  100 RETURN                                                                EPA   43
      END                                                                   EPA   44
```

```
      SUBROUTINE EQCALC (T,P)                                                EQC    1
      LOGICAL ERR                                                            EQC    2
C     CALCULATES THERMOCHEMICAL EQUILIBRIUM FOR SPECIFIED T AND P.           EQC    3
CC          T=TEMPERATURE IN DEG. K                                          EQC    4
C           P=PRESSURE IN ATM.                                               EQC    5
      REAL ACOM(30),ELMENT(10),HP(20)                                        EQC    6
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                                   EQC    7
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                                  EQC    8
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,    EQC    9
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,    EQC   10
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,    EQC   11
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,    EQC   12
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,    EQC   13
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHC,  RHAP,   RHO,  RHOB,    EQC   14
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,    EQC   15
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,    EQC   16
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                      EQC   17
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),  EQC   18
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),   EQC   19
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),   EQC   20
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),   EQC   21
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),   EQC   22
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),   EQC   23
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64)    EQC   24
      COMMON            BETA(64,20),   ELJ(10,20),  GELJ(10,20),           EQC   25
     1 XNUIJ(64,20),XNUIJP(64,20)                                           EQC   26
      COMMON      IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,    EQC   27
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,     EQC   28
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,      EQC   29
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,     EQC   30
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                            EQC   31
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10)    EQC   32
      COMMON ACOM,ELMENT,HP                                                 EQC   33
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBE  EQC   34
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                    EQC   35
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),                  EQC   36
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),          EQC   37
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),               EQC   38
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                          EQC   39
      COMMON /ERROR/ ERR                                                    EQC   40
      COMMON /EQC/ ZCAP(20),ZSEN,ZCH,ZCM,ZRBP,ZRHO                          EQC   41
      COMMON /EQC2/ ZPZ,ZGMU(20)                                            EQC   42
      DATA RNAME /6HEQCALC /                                                EQC   43
      CTSAVE=CT                                                             EQC   44
      CT=T/CTAP                                                             EQC   45
      DO 10 I=1,ISC                                                         EQC   46
      ZCAP(I)=QM(I)                                                         EQC   47
      IF (QM(I).LE.0.) ZCAP(I)=1.E-3                                        EQC   48
10    ZGMU(I)=ALOG(ZCAP(I))                                                 EQC   49
      ZPZ=ALOG(P)                                                           EQC   50
      CALL THERM                                                            EQC   51
      IF (ISMC.EQ.0) GO TO 60                                               EQC   52
      DO 30 I=1,ISMC                                                        EQC   53
      IL=I+ISC                                                              EQC   54
      CHII(I)=-XMJAT(IL)+ZPZ*BET(I)                                         EQC   55
```

```
      DO 20 J=1,ISC                                                     EQC  56
2     CHII(I)=CHII(I)+XMJAT(J)*CDIJ(I,J)                                EQC  57
30    CONTINUE                                                          EQC  58
      DO 50 I=1,ISMC                                                    EQC  59
      SKIL(I)=CHII(I)                                                   EQC  60
      DO 40 J=1,ISC                                                     EQC  61
40    SKIL(I)=SKIL(I)+CDIJ(I,J)*ZGMU(J)                                 EQC  62
      L=I+ISC                                                           EQC  63
      ZGMU(L)=SKIL(I)                                                   EQC  64
      SKIL(I)=EXP(SKIL(I))                                              EQC  65
50    ZCAP(L)=SKIL(I)                                                   EQC  66
60    M1=ISC+1                                                          EQC  67
      N=0                                                               EQC  68
70    N=N+1                                                             EQC  69
      DO 90 J=1,ISC                                                     EQC  70
      AA(J,M1)=QM(J)-ZCAP(J)                                            EQC  71
      IF (ISMC.EQ.0) GO TO 90                                           EQC  72
      DO 80 I=1,ISMC                                                    EQC  73
80    AA(J,M1)=-(CDIJ(I,J)-QM(J)*BET(I))*SKIL(I)+AA(J,M1)               EQC  74
90    CONTINUE                                                          EQC  75
      DO 140 J=1,ISC                                                    EQC  76
      DO 140 K=1,ISC                                                    EQC  77
      IF (J-K) 110,100,110                                              EQC  78
100   AA(J,K)=ZCAP(J)                                                   EQC  79
      GO TO 120                                                         EQC  80
110   AA(J,K)=0.0                                                       EQC  81
120   IF (ISMC.EQ.0) GO TO 140                                          EQC  82
      DO 130 I=1,ISMC                                                   EQC  83
130   AA(J,K)=AA(J,K)+(CDIJ(I,K)*(CDIJ(I,J)-QM(J)*BET(I))*SKIL(I))      EQC  84
140   CONTINUE                                                          EQC  85
      CALL DSMSOL (AA,ISC,22)                                           EQC  86
      IF (ERR) RETURN                                                   EQC  87
      DO 180 K=1,ISC                                                    EQC  88
      ZA=1.+AA(K,M1)                                                    EQC  89
      IF (ZA) 150,150,160                                               EQC  90
150   ZCAP(K)=ZCAP(K)/2.                                                EQC  91
      GO TO 170                                                         EQC  92
160   ZCAP(K)=ZCAP(K)*ZA                                                EQC  93
170   ZGMU(K)=ALOG(ZCAP(K))                                             EQC  94
180   CONTINUE                                                          EQC  95
      IF (ISMC.EQ.0) GO TO 210                                          EQC  96
      DO 200 I=1,ISMC                                                   EQC  97
      SKIL(I)=CHII(I)                                                   EQC  98
      DO 190 L=1,ISC                                                    EQC  99
190   SKIL(I)=SKIL(I)+CDIJ(I,L)*ZGMU(L)                                 EQC 100
      J=I+ISC                                                           EQC 101
      ZGMU(J)=SKIL(I)                                                   EQC 102
      SKIL(I)=EXP(SKIL(I))                                              EQC 103
      ZCAP(J)=SKIL(I)                                                   EQC 104
200   CONTINUE                                                          EQC 105
210   DO 220 K=1,ISC                                                    EQC 106
      IF (ABS(AA(K,M1)).LE.TEST) GO TO 220                              EQC 107
      IF (N-NTEST) 70,70,240                                            EQC 108
220   CONTINUE                                                          EQC 109
      ZCH=0.                                                            EQC 110
```

```
      ZSEN=-ZPZ                                                          EQC 111
      ZCM=0.                                                             EQC 112
      DO 230 I=1,ISS                                                     EQC 113
      ZSEN=ZSEN+ZCAP(I)*(SENT(I)-ZGMU(I))                                EQC 114
      ZCH=ZCH+ZCAP(I)*SHJ(I)                                             EQC 115
230   ZCM=ZCM+ZCAP(I)*CGI(I)                                             EQC 116
      ZRBP=P*ZCM/T/82.055                                                EQC 117
      ZRHO=ZRBP/(1.+ZRBP*BZERO)                                          EQC 118
      ZCH=ZCH+(ZRBP/ZRHO-1.)                                             EQC 119
      CT=CTSAVE                                                          EQC 120
      RETURN                                                             EQC 121
240   WRITE (6,250)                                                      EQC 122
      CALL DUMP (RNAME)                                                  EQC 123
      RETURN                                                             EQC 124
C                                                                        EQC 125
CC                                                                       EQC 126
C                                                                        EQC 127
250   FORMAT (35H1TOO MANY NEWTON-RAPHSON ITERATIONS)                    EQC 128
      END                                                                EQC 129-
```

```
      SUBROUTINE EXACT                                                      EXA    1
      LOGICAL ERR                                                           EXA    2
      REAL ACOM(30),ELMENT(10),HP(20)                                       EXA    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                                  EXA    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                                 EXA    5
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,   EXA    6
     1      C,  CARB,    CH,   CHA,  CLNT,     CM,   CMA,   CRA,   CRP,   EXA    7
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,    CTB,   CTC, CTMAX, CTMXX,   EXA    8
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,    EXA    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,     EXA   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,     EXA   11
     6RHOBAR,  RHOC,  RHOP,  RHPL, RHTH,ROBARA,ROBARP,  SCPG,   SDT,     EXA   12
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,   SU,    SU2,  SUMG,    EXA   13
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                    EXA   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),EXA   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),EXA   16
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),EXA   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(30),EXA   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),EXA   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),EXA   20
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),  XNU1(64)  EXA   21
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),           EXA   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                         EXA   23
      COMMON      IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,   EXA   24
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,    EXA   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,     EXA   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,    EXA   27
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                          EXA   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10)  EXA   29
      COMMON ACOM,ELMENT,HP                                               EXA   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBEXA   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                  EXA   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),                EXA   33
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),        EXA   34
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),             EXA   35
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                        EXA   36
      COMMON /DRV/ SHJDGJ                                                 EXA   37
      COMMON /ERROR/ ERR                                                  EXA   38
      COMMON /SS/ CAS,US                                                  EXA   39
      COMMON /TNERK/ SDTE,CTEB,DCHA,CHB,SDCHA,DQMAX,IFAIL                 EXA   40
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),         EXA   41
     1      KTR(25),ICH(20),IPA(25)                                      EXA   42
      COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE                                 EXA   43
      NAMELIST /EXTDMP/ CT,CMA,SU2,DLOGA,DLOGR,SUMG,SHJDGJ,SUMTDG,        EXA   44
     1      IUPD,NT,DT,DTE,DGJ,PICHI,SHJ                                 EXA   45
      Z1=SU2/CMA                                                          EXA   46
      IF (INEQ.NE.0) GO TO 100                                            EXA   47
      DO 10 I=1,ISSP2                                                     EXA   48
      DO 10 J=1,ISSP4                                                     EXA   49
   10 AAA(I,J)=0                                                          EXA   50
      DO 20 I=1,ISC                                                       EXA   51
      DO 20 J=1,ISS                                                       EXA   52
   20 AAA(I,J)=LPIJ(J,I)                                                  EXA   53
      DO 50 I=ISCP1,ISS                                                   EXA   54
      K=I-ISC                                                             EXA   55
```

```
      S1=BET(K)+SHJ(I)/CT                                                EXA   56
      DO 30 J=1,ISC                                                      EXA   57
      AAA(I,J)=CDIJ(K,J)/GJ(J)                                           EXA   58
30    S1=S1-CDIJ(K,J)*SHJ(J)/CT                                          EXA   59
      DO 50 J=ISCP1,ISS                                                  EXA   60
      IF (I-J) 50,40,50                                                  EXA   61
40    AAA(I,J)=-1.0/GJ(J)                                                EXA   62
50    CONTINUE                                                           EXA   63
      AAA(I,ISSP1)=S1/CT                                                 EXA   64
      AAA(I,ISSP2)=BET(K)                                                EXA   65
60    CONTINUE                                                           EXA   66
      DO 70 J=1,ISS                                                      EXA   67
      AAA(ISSP1,J)=1.0                                                   EXA   68
70    AAA(ISSP2,J)=SHJ(J)                                                EXA   69
      AAA(ISSP1,ISSP1)=SUMG/CT                                           EXA   70
      AAA(ISSP1,ISSP2)=SUMG-Z1/CT                                        EXA   71
      AAA(ISSP1,ISSP3)=-Z1/CT                                            EXA   72
      AAA(ISSP2,ISSP1)=SCPG                                              EXA   73
      AAA(ISSP2,ISSP2)=-Z1                                               EXA   74
      AAA(ISSP2,ISSP3)=-Z1                                               EXA   75
      DO 80 I=ISSP1,ISSP2                                                EXA   76
80    AAA(I,ISSP3)=-DLOGA*AAA(I,ISSP3)                                   EXA   77
      CALL DSMSOL (AAA,ISSP2,22)                                         EXA   78
      IF (ERR) RETURN                                                    EXA   79
      DO 90 I=1,ISS                                                      EXA   80
90    DGJ(I)=AAA(I,ISSP3)                                                EXA   81
      DT=AAA(ISSP1,ISSP3)                                                EXA   82
      DLOGR=AAA(ISSP2,ISSP3)                                             EXA   83
      GO TO 180                                                          EXA   84
100   NE=NT+1                                                            EXA   85
      IF (NT.EQ.2) DCHA=-(CMA/CRP/RHAP/US)*QDPR/RHO/SU                   EXA   86
      SHJDGJ=0.                                                          EXA   87
      SUMTDG=0.                                                          EXA   88
      DO 120 J=1,ISS                                                     EXA   89
      DGJ(J)=0.                                                          EXA   90
      DO 110 I=1,ISR                                                     EXA   91
110   DGJ(J)=DGJ(J)+BETA(I,J)*PICHI(I)                                   EXA   92
      SHJDGJ=SHJDGJ+SHJ(J)*DGJ(J)                                        EXA   93
      IF (J.LT.NT) GO TO 120                                             EXA   94
      SUMTDG=SUMTDG+DGJ(J)                                               EXA   95
120   CONTINUE                                                           EXA   96
      SUMTDG=CT*SUMTDG                                                   EXA   97
      IF (NT.EQ.2) SUMTDG=SUMTDG+CTE*DGJ(1)                              EXA   98
      ECOEF=SQRT(CMA/8.31434E7/CTAP)/RHOP/SU/CRP                         EXA   99
      AAA(1,1)=SCPGH                                                     EXA  100
      IF (NT.EQ.2) AAA(1,3)=CCPJ(1)*GJ(1)                                EXA  101
      AAA(1,2)=-Z1                                                       EXA  102
      IF (IUPD.EQ.1) GO TO 130                                           EXA  103
      AAA(1,NE+1)=Z1*DLOGA-SHJDGJ                                        EXA  104
      IF (NT.EQ.1) GO TO 130                                             EXA  105
      AAA(1,4)=AAA(1,4)-ECOEF*QDPR                                       EXA  106
130   AAA(2,1)=SUMGH                                                     EXA  107
      IF (NT.EQ.2) AAA(2,3)=GJ(1)                                        EXA  108
      IF (IUPD.EQ.1) GO TO 140                                           EXA  109
      AAA(2,2)=CT*SUMGH-Z1+(NT-1)*GJ(1)*CTE                              EXA  110
```

```
      AAA(2,NE+1)=-SUMTDG+Z1*DLOGA                                                    EXA 111
140   IF (NT.EQ.1) GO TO 150                                                          EXA 112
      AAA(3,1)=0.                                                                     EXA 113
      AAA(3,3)=1.5*GJ(1)                                                              EXA 114
      IF (IUPD.EQ.1) GO TO 150                                                        EXA 115
      AAA(3,2)=-GJ(1)*CTE                                                             EXA 116
      AAA(3,4)=-1.5*CTE*DGJ(1)+ECOEF*QDPE                                             EXA 117
150   IF (IUPD.EQ.0) GO TO 170                                                        EXA 118
      AAA(1,NE+1)=Z1*DLOGR-SHJDGJ                                                     EXA 119
      IF (NT.EQ.1) GO TO 160                                                          EXA 120
      AAA(1,4)=AAA(1,4)-ECOEF*QDPR                                                    EXA 121
160   AAA(2,2)=-Z1                                                                    EXA 122
      AAA(2,NE+1)=-SUMTDG-DLOGR*(CT*SUMGH-Z1+(NT-1)*GJ(1)*CTE)                        EXA 123
      AAA(3,2)=0.                                                                     EXA 124
      IF (NT.EQ.1) GO TO 170                                                          EXA 125
      AAA(3,4)=GJ(1)*CTE*DLOGR-1.5*CTE*DGJ(1)+ECOEF*QDPE                              EXA 126
170   CALL DSMSOL (AAA,NE,22)                                                         EXA 127
      IF (ERR) RETURN                                                                 EXA 128
      DT=AAA(1,NE+1)                                                                  EXA 129
      IF (IUPD.EQ.0) DLOGR=AAA(2,NE+1)                                                EXA 130
      IF (IUPD.EQ.1) DLOGA=AAA(2,NE+1)                                                EXA 131
      IF (NT.EQ.2) DTE=AAA(3,NE+1)                                                    EXA 132
180   IF (DLOGA.NE.0.) GO TO 190                                                      EXA 133
      AMACH=1.                                                                        EXA 134
      GO TO 200                                                                       EXA 135
190   AMACH=1./SQRT(ABS(1.+DLOGA/DLOGR))                                              EXA 136
200   IF (INEQ.NE.0.AND.ISW5B.NE.0) WRITE (6,EXTDMP)                                  EXA 137
      RETURN                                                                          EXA 138
      END                                                                             EXA 139-
```

```
EXP  1
EXP  2
EXP  3
EXP  4
EXP  5
EXP  6
EXP  7
EXP  8-
```

```
      FUNCTION EXP(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      IF (X.GT.-180.) GO TO 10
      EXP=0.DD
      GO TO 20
   10 EXP=DEXP(X)
   20 RETURN
      END
```

```
      SUBROUTINE FINDX (A,UPDOWN,X)                                          FDX    1
      LOGICAL ERR,UPPER                                                      FDX    2
C     SOLVES GEOMETRIC AREA RELATION FOR X, GIVEN THE VALUE A(X).            FDX    3
C           A=GEOMETRIC AREA RATIO (=1.0 AT GEOMETRIC THROAT)                FDX    4
C           UPDOWN=1. IF DOWNSTREAM SOLUTION IS DESIRED                      FDX    5
C                 =-1. IF UPSTREAM SOLUTION IS DESIRED                       FDX    6
C           X=VALUE OF X OBTAINED                                            FDX    7
C           ENTRY FINDXC SOLVES FOR X AT WHICH THE MBL-TH PROFILE HAS A      FDX    8
C           HALF-WIDTH OF A.                                                 FDX    9
      COMMON /ERROR/ ERR                                                     FDX   10
      DIMENSION Z(2),DZDX(2),RNAME(2)                                        FDX   11
      DATA RNAME /5HFINDX,6HFINDXC/                                          FDX   12
      NAMELIST /DMP/ V,AR,DADX,VO,VL,VU,ERRL,ERRU,ERR,UPPER                  FDX   13
      IENTRY=1                                                               FDX   14
      GO TO 10                                                               FDX   15
      ENTRY FINDXC(A,MBL,X)                                                  FDX   16
      IENTRY=2                                                               FDX   17
      UPDOWN=1.                                                              FDX   18
      N=0                                                                    FDX   19
      V=100.                                                                 FDX   20
      GO TO 60                                                               FDX   21
10    N=0                                                                    FDX   22
      V=1.                                                                   FDX   23
      IF (ABS(A-1.0).LT.1.E-5) GO TO 40                                      FDX   24
      IF (A-1.) 20,40,50                                                     FDX   25
20    WRITE (6,170) A                                                        FDX   26
30    CALL DUMP (RNAME(IENTRY))                                              FDX   27
40    X=0.                                                                   FDX   28
      RETURN                                                                 FDX   29
50    VL=0.                                                                  FDX   30
      ERRL=1.-A                                                              FDX   31
      UPPER=.FALSE.                                                          FDX   32
      ERRU=1.E30                                                             FDX   33
C     NEWTON-RAPHSON ITERATION TO SOLVE FOR X                                FDX   34
60    N=N+1                                                                  FDX   35
      IF (N.LE.50) GO TO 70                                                  FDX   36
      WRITE (6,180) A,UPDOWN,IENTRY,MBL                                      FDX   37
      WRITE (6,DMP)                                                          FDX   38
      GO TO 30                                                               FDX   39
70    GO TO (80,130), IENTRY                                                 FDX   40
80    CALL GEOMAR (SIGN(V,UPDOWN),AR,DADX)                                   FDX   41
      IF (ERR) RETURN                                                        FDX   42
      ERA=AR-A                                                               FDX   43
      IF (ERA) 90,160,110                                                    FDX   44
90    IF (ERA-ERRL) 140,140,100                                              FDX   45
100   ERRL=ERA                                                               FDX   46
      VL=V                                                                   FDX   47
      GO TO 140                                                              FDX   48
110   UPPER=.TRUE.                                                           FDX   49
      IF (ERA-ERRU) 120,140,140                                              FDX   50
120   ERRU=ERA                                                               FDX   51
      VU=V                                                                   FDX   52
      GO TO 140                                                              FDX   53
130   CALL GMAR3 (V,DZDX(1),DZDX(2),Z(1),Z(2))                               FDX   54
      AR=Z(MBL)                                                              FDX   55
```

```
      DADX=DZDX(MBL)                                                    FDX   56
140   IF (ABS(1.-AR/A).LE.1.E-5) GO TO 160                              FDX   57
      IF (N.LE.10) GO TO 150                                            FDX   58
      IF (IENTRY.EQ.2.OR..NOT.UPPER) GO TO 150                          FDX   59
      V=VL-(VU-VL)*ERRL/(ERRU-ERRL)                                     FDX   60
      GO TO 60                                                          FDX   61
150   VO=V                                                              FDX   62
      V=V+(A-AR)/ABS(DADX)                                              FDX   63
      V=AMIN1(V,2.*VO)                                                  FDX   64
      GO TO 60                                                          FDX   65
160   X=SIGN(V,UPDOWN)                                                  FDX   66
      RETURN                                                            FDX   67
C                                                                       FDX   68
170   FORMAT (49H0FINDX CALLED WITH AN AREA RATIO LESS THAN UNITY,,10X,2FDX   69
     1HA=,1PE15.8)                                                      FDX   70
180   FORMAT (33H0MORE THAN 50 ITERATIONS IN FINDX,10X,2HA=,1PD15.8,10X,FDX   71
     17HUPDOWN=,0PF3.0,10X,7HIENTRY=,I2,10X,4HMBL=,I2)                  FDX   72
      END                                                               FDX   73-
```

```
      SUBROUTINE FROZEQ                                                 FRO    1
      REAL ACOM(30),ELMENT(10),HP(20)                                   FRO    2
      LOGICAL SWTCH,SKIP,MODLPT,TSTEPC,AXISYM,WEDGEM,ERR,LASTPT         FRO    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              FRO    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             FRO    5
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  FRO    6
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  FRO    7
     2   CRR9,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  FRO    8
     3    CTP,  CTPL,   CTT,    CX,   CX8, CXMAX,DATFST,DBTEST, DELT1,  FRO    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  FRO   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHO8,  FRO   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  FRO   12
     7   SEN,  SHPG,   SC,     SL,  SL64,    SM,    SU,   SU2,  SUMG,  FRO   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  FRO   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),FRO   15
     1  CCPJ(20), CFACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),FRO   16
     2CLNMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),FRO   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),FRO   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),FRO   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),FRO   20
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),  XNUI(64) FRO   21
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          FRO   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       FRO   23
      COMMON      IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  FRO   24
     1  ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  FRO   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    FRO   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,  FRO   27
     4    NIT,   NNN,   NNS,   NQS,   NQT, NTEST                       FRO   28
      COMMON    IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) FRO   29
      COMMON ACOM,ELMENT,HP                                             FRO   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBFRO   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                FRO   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              FRO   33
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       FRO   34
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            FRO   35
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       FRO   36
      COMMON /TEMPRY/ SAVEC(20)                                         FRO   37
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         FRO   38
     1      NSECTU(2),ISHAPE(12,2),NFROFL(2),NPFFLS,NBL                 FRO   39
      COMMON /BLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOT(2),      FRO   40
     1      TAUW(2)                                                     FRO   41
      COMMON /RESPRP/ VISCR,PRR,SIGR,FLEWR                              FRO   42
      COMMON /TRPROP/ VISC,PR,SIGMA,FLEWIS                              FRO   43
      COMMON /ERROR/ ERR                                                FRO   44
      DIMENSION DELBLO(2),           DELBLP(2)                          FRO   45
      COMMON /OUTPUT/ FVOUT(35),GJMF(20)                                FRO   46
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,   CATFAC,TMODEL,XMODP1,DXMODP,   FRO   47
     1      TPLATE,KDIM                                                 FRO   48
      COMMON /STAG/ T2E,EPSLON,T2F,EPSF                                 FRO   49
      COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,MBL                        FRO   50
      DIMENSION BLVAR(13),BLVARO(13),BLVARS(13)                         FRO   51
      EQUIVALENCE (BLVAR(1),REPF)                                       FRO   52
      DIMENSION SOLN(3,2)                                               FRO   53
      DATA SOLN /4HFROZ,4HEN F,3HLOW,4HEQUI,4HLIBR,3HIUM/               FRO   54
      DATA ONE /1./                                                     FRO   55
```

```
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,PWALL,CPWALL,VISROT,DIAM(2),    FRO   56
     1     SW,RO,JDIM,IPCINT                                             FRO   57
      COMMON /SS/ CAS,US                                                 FRO   58
      COMMON /NEQ/ OMDST(2),DDELBL(2)                                    FRO   59
      COMMON /SWITCH/ XPB,DXPB,W                                         FRO   60
      COMMON /AVG/ WSAVE                                                 FRO   61
      COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,      FRO   62
     1     NWX,NANGLE,NRADLE,WEDGEM,AXISYM,ISW98                         FRO   63
      NAMELIST /FRDMP/ CT,ISONIC,FLAG,PRES,RHO,SU,AMACH,AFNX,XLAST,      FRO   64
     1     DELBL,DELBLO,XO,DELBLP,XP,BLVAR,IRSTRT,W                      FRO   65
      NAMELIST /FRDMP2/ XMODEL,X1,X2,CT,CT1,CT2,NMPI,XFRAC,DELBL,        FRO   66
     1     DELBLO,ERRX                                                   FRO   67
C     FROZEN FLOW SOLUTION                                               FRO   68
      ENTRY FROZEN                                                       FRO   69
      ISOLN=1                                                            FRO   70
C     DETERMINE SONIC CONDITIONS IN FROZEN FLOW                          FRO   71
      DELT=DELT1                                                         FRO   72
      F1=0.0                                                             FRO   73
      T1=1.0                                                             FRO   74
      CT=T1-DELT                                                         FRO   75
      S1=0.0                                                             FRO   76
      CLNMA=ALOG(CMA)                                                    FRO   77
      DO 10 J=1,ISS                                                      FRO   78
      SS(J)=ALOG(GJA(J))                                                 FRO   79
10    S1=S1-GJA(J)*(SS(J)+CLNMA)                                         FRO   80
      S2=-(ZPA/CMA)                                                      FRO   81
      ENT=S1+S2-CRRB                                                     FRO   82
20    T2=CT                                                              FRO   83
      CALL THERM                                                         FRO   84
      CALL PROP                                                          FRO   85
      F2=FLUX                                                            FRO   86
30    CT=CT-DELT                                                         FRO   87
      T3=CT                                                              FRO   88
      CALL THERM                                                         FRO   89
      CALL PROP                                                          FRO   90
      F3=FLUX                                                            FRO   91
      IF (F3-F2) 40,60,80                                                FRO   92
40    IF (DELT-TESTB) 90,50,50                                           FRO   93
50    DELT=DELT/2.0                                                      FRO   94
      CT=T1-DELT                                                         FRO   95
      GO TO 20                                                           FRO   96
60    IF (DELT-TESTB) 90,70,70                                           FRO   97
70    T1=T2                                                              FRO   98
      F1=F2                                                              FRO   99
      DELT=DELT/2.0                                                      FRO  100
      CT=T1-DELT                                                         FRO  101
      GO TO 20                                                           FRO  102
80    T1=T2                                                              FRO  103
      T2=T3                                                              FRO  104
      F1=F2                                                              FRO  105
      F2=F3                                                              FRO  106
      GO TO 30                                                           FRO  107
90    Z=F1-F2+F3-F2                                                      FRO  108
      IF (ABS(Z/F2).GT..00001) GO TO 100                                 FRO  109
      CT=T2                                                              FRO  110
```

```
      GO TO 110                                                          FRO 111
  100 CT=T2+DELT*(F3-F1)/(2.0*Z)                                         FRO 112
  110 CONTINUE                                                           FRO 113
      CALL THERM                                                         FRO 114
      CALL PROP                                                          FRO 115
      SM=FLUX                                                            FRO 116
      CTMAX=CT                                                           FRO 117
      PRESTH=PRES                                                        FRO 118
      RHTH=RHO                                                           FRO 119
      IPASS=2                                                            FRO 120
      GO TO 120                                                          FRO 121
C     EQUILIBRIUM FLOW SOLUTION                                          FRO 122
      ENTRY EQUIL(IPASS)                                                 FRO 123
      ISCLN=2                                                            FRO 124
  120 IRSTRT=1                                                           FRO 125
      W=WSAVE                                                            FRO 126
  130 DO 140 J=1,ISS                                                     FRO 127
  140 GJ(J)=GJA(J)                                                       FRO 128
      ISONIC=-1                                                          FRO 129
      LASTPT=.FALSE.                                                     FRO 130
      SKIP=.FALSE.                                                       FRO 131
      DELTSV=DELT1                                                       FRO 132
      IF (ISW3H.NE.0) DELT1=0.049*(1.-CTMAX)                             FRO 133
      XMOD1=XMP1                                                         FRO 134
      ITS=1                                                              FRO 135
      TSTEPC=.FALSE.                                                     FRO 136
      MODLPT=.FALSE.                                                     FRO 137
      CALL NEXTMP (ITS,XMOD1,XMODEL)                                     FRO 138
      INEQ=0                                                             FRO 139
      T2E=CTAP                                                           FRO 140
      DO 150 I=1,13                                                      FRO 141
  150 BLVAR(I)=0.                                                        FRO 142
      EPSLON=0.                                                          FRO 143
      T2F=CTAP                                                           FRO 144
      EPSF=0.                                                            FRO 145
      DO 160 I=1,NPRFLS                                                  FRO 146
      DELBL(I)=0.                                                        FRO 147
  160 OMDST(I)=1.                                                        FRO 148
      ISWTCH=0                                                           FRO 149
      SWTCH=.FALSE.                                                      FRO 150
      IPOINT=1                                                           FRO 151
      PRES=1.0                                                           FRO 152
      CT=1.0                                                             FRO 153
      RHO=1.0                                                            FRO 154
      RHOBAR=ROBARA                                                      FRO 155
      SU=0.0                                                             FRO 156
      AMACH=0.0                                                          FRO 157
      AFNX=0.0                                                           FRO 158
      RHOB=1.0                                                           FRO 159
      PRESB=1.0                                                          FRO 160
      DELTV=DELT1                                                        FRO 161
      XO=0.                                                              FRO 162
      XP=0.                                                              FRO 163
      FLAG=-1.                                                           FRO 164
      IF (IPASS.EQ.1) GO TO 180                                          FRO 165
```

```
      WRITE (6,630) (SOLN(N,ISOLN),N=1,3)                                FRO 166
      DO 170 J=1,ISS                                                     FRO 167
  170 SAVEC(J)=GJA(J)*CMA                                                FRO 168
      AMACH=0.                                                           FRO 169
      AFNX=1.E38                                                         FRO 170
      VICC=VISCR                                                         FRO 171
      SIGMA=SIGR                                                         FRO 172
      CX=-1.E10                                                          FRO 173
      CH=CHA                                                             FRO 174
      CM=CMA                                                             FRO 175
      CALL OUT2 (ISOLN)                                                  FRO 176
      IF (ERR) RETURN                                                    FRO 177
C     DECREMENT TEMPERATURE TO GENERATE SOLUTION                         FRO 178
  180 IF (.NOT.TSTEPC.AND.(XO-XP).LT.10.) GO TO 190                      FRO 179
      TSTEPC=.TRUE.                                                      FRO 180
      DELTV=AMIN1(DELT1,10.*DELTV/(XO-XP))                               FRO 181
      DELTV=AMAX1(DELTV,0.02*DELT1)                                      FRO 182
  190 DELTV=AMIN1(DELTV,0.1*CT)                                          FRO 183
      CT=CT-DELTV                                                        FRO 184
      IF (ISONIC.EQ.-1.AND.CT.LE.CTMAX) GO TO 200                        FRO 185
      IF (CT.LT.TSTOP) GO TO 610                                         FRO 186
      GO TO 220                                                          FRO 187
C     THROAT HAS BEEN REACHED.  COMPUTE CONDITIONS AT THROAT.            FRO 188
  200 IF (ISW3B.NE.0) GO TO 210                                          FRO 189
      FLAG=1.                                                            FRO 190
      ISONIC=1                                                           FRO 191
      GO TO 220                                                          FRO 192
  210 CTSTAR=CT+DELTV                                                    FRO 193
      CT=CTMAX                                                           FRO 194
      ISONIC=0                                                           FRO 195
C     COMPUTE THE FLOW CONDITIONS FOR TEMPERATURE CT.                    FRO 196
  220 GO TO (230,240), ISOLN                                             FRO 197
  230 CALL THERM                                                         FRO 198
      CALL PROP                                                          FRO 199
      GO TO 250                                                          FRO 200
  240 CALL NEWRAP                                                        FRO 201
      IF (ERR) RETURN                                                    FRO 202
  250 AFNX=AMAX1(SM/FLUX,ONE)                                            FRO 203
      IF (PRESB.EQ.PRES) GO TO 260                                       FRO 204
      AMACH=SU*SQRT(RHO*ABS(ALOG(RHO/RHOB)/ALOG(PRES/PRESB))/PRES)       FRO 205
      PRESB=PRES                                                         FRO 206
  260 RHOB=RHO                                                           FRO 207
      AS=SU*US/AMACH                                                     FRO 208
      CAS=AS*AS/CT                                                       FRO 209
      DO 290 J=1,ISS                                                     FRO 210
      GO TO (270,280), ISOLN                                             FRO 211
  270 SAVEC(J)=GJA(J)*CMA                                                FRO 212
      GO TO 290                                                          FRO 213
  280 SAVEC(J)=GJ(J)*CM                                                  FRO 214
  290 CONTINUE                                                           FRO 215
      IF (SKIP) GO TO 360                                                FRO 216
C     DETERMINE X FROM EFFECTIVE AREA RATIO                              FRO 217
      IF (ISW3B.EQ.0) GO TO 310                                          FRO 218
      IF (ISONIC) 310,320,300                                            FRO 219
  300 IF (SWITCH) GO TO 340                                              FRO 220
```

```
      IS .CH=ISWTCH+1                                                    FRO 221
      IF (ISWTCH.EQ.3) GO TO 330                                         FRO 222
C     X BASED ON APPROXIMATING GEOMETRIC AREA RATIO BY AFNX              FRO 223
310   CALL FINDX (AFNX,FLAG,CX)                                          FRO 224
      IF (ERR) RETURN                                                    FRO 225
      IF (ISW3B.EQ.0) GO TO 410                                          FRO 226
      IF (CX.LT.XZERO) GO TO 180                                         FRO 227
      GO TO 370                                                          FRO 228
320   CX=0.                                                              FRO 229
      DELT1=DELTSV                                                       FRO 230
      DELTV=DELT1                                                        FRO 231
      GO TO 370                                                          FRO 232
330   SWTCH=.TRUE.                                                       FRO 233
      XPB=XO                                                             FRO 234
      DXPB=XO-XP                                                         FRO 235
C     X CORRECTED FOR DISPLACEMENT THICKNESS                             FRO 236
340   CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)                                FRO 237
      IF (ERR) RETURN                                                    FRO 238
      IF (MODLPT) GO TO 360                                              FRO 239
      IF (CX.GT.XLAST) GO TO 370                                         FRO 240
      WRITE (6,640)                                                      FRO 241
      IF (ISW3B.EQ.0.OR.AR.GT.4.*AFNX) GO TO 350                         FRO 242
      IRSTRT=IRSTRT+1                                                    FRO  43
      IF (IRSTRT.GT.3) GO TO 350                                         FRO 244
      W=W/2.                                                             FRO 245
      WRITE (6,650) W                                                    FRO 246
      GO TO 130                                                          FRO 247
350   WRITE (6,FRDMP)                                                    FRO 248
      RETURN                                                             FRO 249
360   CALL TRANSP (CT*CTAP,PRES*PRESA)                                   FRO 250
      GO TO 410                                                          FRO 251
370   XLAST=CX                                                           FRO 252
      CALL BLAYER (.TRUE.)                                               FRO 253
      IF (ERR) RETURN                                                    FRO 254
      IF (IPASS.EQ.1.AND.ISONIC.NE.0) GO TO 180                          FRO 255
      IF (IS ONIC) 410,380,400                                           FRO 256
380   DO 390 M=1,NPRFLS                                                  FRO 257
390   DMDST( M)=DELBL(M)-DELBL(M)                                        FRO 258
      IF (IP .EQ.1) RETURN                                               FRO 259
400   CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)                                FRO 260
410   IF (MODLPT) GO TO 500                                              FRO 261
      IF (.NOT.SKIP) GO TO 420                                           FRO 262
      SKIP=.FALSE.                                                       FRO 263
      GO TO 550                                                          FRO 264
420   XP=XO                                                              FRO 265
      XO=CX                                                              FRO 266
      IF (ISW3B.EQ.0) GO TO 440                                          FRO 267
      DO 430 M=1,NPRFLS                                                  FRO 268
      DELBLP(M)=DELBLO(M)                                                FRO 269
430   DELBLO(M)=DELBL(M)                                                 FRO 270
440   IF (CX.LT. (MAX) GO TO 450                                         FRO 271
      XMODEL=CX .X                                                       FRO 272
      LASTPT=.TRUE.                                                      FRO 273
450   IF (CX.GE.XMODEL) GO TO 460                                        FRO 274
      IF (SKIP) GO TO 220                                                FRO 275
```

```
      GO TO 550                                                          FRO 276
  460 X1=XP                                                              FRO 277
      X2=XO                                                              FRO 278
      MODLPT=.TRUE.                                                      FRO 279
      SKIP=.FALSE.                                                       FRO 280
      NMPI=0                                                             FRO 281
      CT1=CT+DELTV                                                       FRO 282
      CT2=CT                                                             FRO 283
      CTSAVE=CT                                                          FRO 284
      IF (ISW3B.EQ.0) GO TO 490                                          FRO 285
      CXSAVE=CX                                                          FRO 286
      XFRAC=(XMODEL-XP)/(XO-XP)                                          FRO 287
      DO 470 M=1,NPRFLS                                                  FRO 288
  470 DELBL(M)=(DELBLC(M)-DELBLP(M))*XFRAC+DELBLP(M)                     FRO 289
      DO 480 K=1,13                                                      FRO 290
  480 BLVARS(K)=BLVAR(K)                                                 FRO 291
  490 CT=CT1+(CT2-CT1)*(XMODEL-X1)/(X2-X1)                               FRO 292
      GO TO 220                                                          FRO 293
  500 NMPI=NMPI+1                                                        FRO 294
      ERRX=CX-XMODEL                                                     FRO 295
      IF (ABS(ERRX).LE.0.0025) GO TO 520                                 FRO 296
      IF (NMPI.LE.9) GO TO 510                                           FRO 297
      WRITE (6,660) CX,XMODEL                                            FRO 298
      WRITE (6,FRDMP2)                                                   FRO 299
      GO TO 520                                                          FRO 300
  510 CT1=CT2                                                            FRO 301
      CT2=CT                                                             FRO 302
      X1=X2                                                              FRO 303
      X2=CX                                                              FRO 304
      GO TO 490                                                          FRO 305
  520 IF (LASTPT) GO TO 530                                              FRO 306
      CALL NEXTMP (ITS,XMOD1,XMODEL)                                     FRO 307
  530 SKIP=.TRUE.                                                        FRO 308
      IF (ISW3B.EQ.0) GO TO 550                                          FRO 309
      DO 540 K=1,13                                                      FRO 310
  540 BLVAR(K)=BLVARO(K)+(BLVARS(K)-BLVARO(K))*XFRAC                     FRO 311
  550 CALL OUT2 (ISOLN)                                                  FRO 312
      IF (ERR) RETURN                                                    FRO 313
      IF (.NOT.MODLPT) GO TO 580                                         FRO 314
      MODLPT=.FALSE.                                                     FRO 315
      CALL MODEL                                                         FRO 316
      IF (LASTPT) GO TO 610                                              FRO 317
      IF (ERR) RETURN                                                    FRO 318
      CT=CTSAVE                                                          FRO 319
      CX=CXSAVE                                                          FRO 320
      IF (ISW3B.EQ.0) GO TO 440                                          FRO 321
      DO 560 K=1,13                                                      FRO 322
  560 BLVAR(K)=BLVARS(K)                                                 FRO 323
      DO 570 M=1,NPRFLS                                                  FRO 324
  570 DELBL(M)=DELBLO(M)                                                 FRO 325
      GO TO 440                                                          FRO 326
  580 IF (ISW3B.EQ.0) GO TO 600                                          FRO 327
      DO 590 K=1,13                                                      FRO 328
  590 BLVARO(K)=BLVAR(K)                                                 FRO 329
  600 IF (CX.GT.CXMAX) GO TO 610                                         FRO 330
```

```
      IF (ISONIC.NE.0) GO TO 180                                        FRO 331
      CT=CTSTAR                                                         FRO 332
      ISONIC=1                                                          FRO 333
      FLAG=1.                                                           FRO 334
      GO TO 180                                                         FRO 335
610   IF (NPRFLS.EQ.1) GO TO 620                                        FRO 336
      AXISYM=.TRUE.                                                     FRO 337
      CALL MODEL                                                        FRO 338
      AXISYM=.FALSE.                                                    FRO 339
620   RETURN                                                            FRO 340
C                                                                       FRO 341
C                                                                       FRO 342
C                                                                       FRO 343
630   FORMAT (1H1,3A4,8HSOLUTION//)                                     FRO 344
640   FORMAT (22H0X DECREASED IN FROZEQ)                                FRO 345
650   FORMAT (103H RESTART SOLUTION WITH A LARGER AVERAGING DISTANCE FORFRO 346
     1 THE BOUNDARY LAYER CORRELATION PARAMETER. NEW W=,F9.6/1H1)       FRO 347
660   FORMAT (73H ITERATION TO OBTAIN FREE-STREAM SOLUTION AT MODEL POINFRO 348
     1T DID NOT CONVERGE,5X,2HX=,1PE10.3,5X,7HXMODEL=,E10.3,3H CM)      FRO 349
      END                                                               FRO 350-
```

```
      SUBROUTINE GEOM                                                   GEO    1
      REAL ACOM(30),ELMENT(10),HP(20)                                   GEO    2
      LOGICAL ERR                                                       GEO    3
      DOUBLE PRECISION AA,CAPX,CDIJ                                     GEO    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             GEO    5
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,  GEO    6
     1      C,  CARB,     CH,    CHA,  CLNT,     CM,   CMA,   CRA,   CRP,  GEO    7
     2  CRRB,   CRS,   CSTA,     CT,  CTAP,    CTB,   CTC, CTMAX, CTMXX,  GEO    8
     3   CTP,  CTPL,    CTT,     CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  GEO    9
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,   ENT,  FLUX, HDELX,   PCT,  GEO   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  GEO   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCRG,   SDT,  GEO   12
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  GEO   13
     8  TEST, TFSTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  GEO   14
      COMMON         BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),GEO   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),GEO   16
     2CL NIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),GEO   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),GEO   18
     4    GQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),GEO   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),GEO   20
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),   XNUI(64) GEO   21
      COMMON          BETA(64,20),   FLJ(10,20),  GELJ(10,20),          GEO   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       GEO   23
      COMMON      IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  GEO   24
     1  ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  GEO   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,  GEO   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,    JJK,    LC,    M1,  NFIT,  GEO   27
     4    NIT,   NNN,   NNS,   NQS,   NQT, NTEST                       GEO   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) GEO   29
      COMMON ACOM,ELMENT,HP                                             GEO   30
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  GEO   31
     1     SA,RO,JDIM,IPCINT                                            GEO   32
      COMMON /NEQ/ OMOST(2),DDELBL(2)                                   GEO   33
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         GEO   34
     1     NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                  GEO   35
      COMMON /AEGEOM/ SQRTA,S1,S2                                       GEO   36
      COMMON /ERROR/ ERR                                                GEO   37
      DIMENSION DELSTP(2)                                               GEO   38
      DATA RNAME /4HGEOM/                                               GEO   39
      CALL GEOMAR (CX,S1,S2)                                            GEO   40
      IF (ERR) RETURN                                                   GEO   41
      IF (NPRFLS.EQ.1.AND.JDIM.EQ.1) SQRTA=SQRT(S1)                     GEO   42
      IF (ISW3B.EQ.0.OR.IUPD.EQ.1) GO TO 10                             GEO   43
      CALL AESCLN (CX)                                                  GEO   44
   10 IF (INEQ.EQ.0.OR.IUPD.EQ.0) GO TO 180                             GEO   45
      IF (S1.LE.1.D0) GO TO 50                                          GEO   46
      ARG=2.*C*(S1-1.D0)/A                                              GEO   47
      DELRHO=SQRT(ARG)                                                  GEO   48
      IF (S1.GT.1.001) GO TO 60                                         GEO   49
      IF (S2) 20,20,30                                                  GEO   50
   20 SGN=1.                                                            GEO   51
      GO TO 40                                                          GEO   52
   30 SGN=-1.                                                           GEO   53
   40 RHO=RHTH+SGN*SQRT(ARG-SGN    1.)/3.*SQRT(A/C/(A+2.))*DELRHO**3)   GEO   54
      GO TO 130                                                         GEO   55
```

```
50      S1=1.                                                          GEO  56
        S2=0.                                                          GEO  57
        RHO=RHTH                                                       GEO  58
        GO TO 130                                                      GEO  59
60      A2=S1*S1                                                       GEO  60
        N=0                                                            GEO  61
        OPHA=1.+0.5*A                                                  GEO  62
        IF (S2) 70,70,80                                               GEO  63
70      RHO=AMIN1(RHTH+DELRHO,(1.-C/A2)**(1./A))                       GEO  64
        GO TO 90                                                       GEO  65
80      RHO=AMAX1(RHTH-DELRHO,SQRT(C)/S1)                              GEO  66
90      N=N+1                                                          GEO  67
        IF (N-50) 100,100,130                                          GEO  68
100     CONTINUE                                                       GEO  69
        IF (RHO.GE.0.) GO TO 110                                       GEO  70
        WRITE (6,200)                                                  GEO  71
        CALL DUMP (RNAME)                                              GEO  72
        RETURN                                                         GEO  73
110     Z=RH  **A                                                      GEO  74
        OMZ=1.-Z                                                       GEO  75
        AN=SQRT(C/OMZ)/RHO                                             GEO  76
        DADR=AN/RHO*(OPHA*Z-1.)/OMZ                                    GEO  77
        DD=(S1-AN)/DADR                                                GEO  78
        RHO=RHO+DD                                                     GEO  79
        IF (ABS(DD/RHO).GE.1.0E-6) GO TO 90                            GEO  80
        Y=SIGN(1.0D0,CX*(RHO-RHTH))                                    GEO  81
        IF (Y) 130,130,120                                             GEO  82
120     RHO=RHTH-SIGN(1.0D-02,CX)                                      GEO  83
        N=0                                                            GEO  84
        GO TO 90                                                       GEO  85
130     AFNX=SM/(SU*RHO)                                               GEO  86
        IF (S2.NE.0.) GO TO 170                                        GEO  87
        IF (NPRFLS.EQ.1) GO TO 150                                     GEO  88
        D2LA=0.                                                        GEO  89
        DO 140 J=1,2                                                   GEO  90
        DELSTP(J)=R0*(1.-OMDST(J))                                     GEO  91
        NSUJ=NSECTU(J)                                                 GEO  92
140     D2LA=D2LA+1./(RTHCM(J)-DELSTP(J))/PARAM(3,NSUJ,J)              GEO  93
        GO TO 160                                                      GEO  94
150     NSUJ=NSECTU(1)                                                 GEO  95
        D2LA=1./(RTHCM(1)*OMDST(1)*PARAM(3,NSUJ,1))                    GEO  96
        IF (JDIM.NE.0) D2LA=2.*D2LA/SQRTA                              GEO  97
160     DLOGR=-SQRT(C*D2LA/A/RHTH)                                     GEO  98
        GO TO 190                                                      GEO  99
170     DLOGR=((2.*C)/(A*(S1*RHO)**2-C*(A+2.)))*S2/S1                  GEO 100
        GO TO 190                                                      GEO 101
180     AFNX=S1                                                        GEO 102
        DLOGA=S2/S1                                                    GEO 103
        RHO=SM/(SU*AFNX)                                               GEO 104
190     RETURN                                                         GEO 105
C                                                                      GEO 106
C                                                                      GEO 107
C                                                                      GEO 108
200     FORMAT (21HONEGATIVE RHO IN GEOM)                              GEO 109
        END                                                            GEO 110-
```

```
      SUBROUTINE GEOMAR (X,ARATIO,DERIVA)                                GMR    1
C     COMPUTES GEOMETRIC AREA RATIO ARATIO AND ITS DERIVATIVE DERIVA.    GMR    2
      REAL ACOM(30),ELMENT(10),HP(20)                                    GMR    3
      DOUBLE PRECISION AA,CAPX,CDIJ                                      GMR    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                              GMR    5
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  GMR    6
     1       C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  GMR    7
     2  CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,    GMR    8
     3   CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,    GMR    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,    GMR   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,    GMR   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,    GMR   12
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,    GMR   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                    GMR   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),GMR   15
     1  CCPJ(20), CFACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),GMR   16
     2CLN1MC(64), CLNFI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),GMR   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),GMR   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),GMR   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),GMR   20
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) GMR   21
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          GMR   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                        GMR   23
      COMMON      IC,     IM,   INEQ,  INEQV,     IP,   IRUN,    ISC, ISCP1,  GMR   24
     1  ISMC,ISMCNR,    ISR,    ISS,  ISSNR,  ISSP1,  ISSP2,  ISSP3, ISSP4,  GMR   25
     2 ISW1A, ISW1B,  ISW2A,  ISW2B,  ISW3A,  ISW3B,  ISW4A,  ISW4B, ISW5A,  GMR   26
     3 ISW5B, ISW6A,  ISW6B,   IUPD,  IZERO,    JJK,     LC,     M1,  NFIT,  GMR   27
     4    NIT,    NNN,    NNS,    NOS,    NQT, NTEST                      GMR   28
      COMMON    IGJ(20),    IGM(20),    ITB(5),    KUR(64,20),  LPIJ(20,10) GMR   29
      COMMON ACOM,ELMENT,HP                                               GMR   30
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),           GMR   31
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                   GMR   32
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),    GMR   33
     1      SW,RO,JDIM,IPOINT                                             GMR   34
      DIMENSION SN(2),RAD(2),S1(2),S2(2),DRV(2)                           GMR   35
      DATA RNAME /6HGEOMAR/                                               GMR   36
      DATA SN /-1.,1./                                                    GMR   37
      COMMON /THRT/ PSA                                                   GMR   38
      IENTRY=1                                                            GMR   39
      GO TO 10                                                            GMR   40
      ENTRY GMAR(X,Y)                                                     GMR   41
      IENTRY=2                                                            GMR   42
      GO TO 10                                                            GMR   43
      ENTRY GMAR2(X,Y,Z)                                                  GMR   44
      IENTRY=3                                                            GMR   45
      GO TO 10                                                            GMR   46
      ENTRY GMAR3(X,DYDX,DZDX,Y,Z)                                        GMR   47
      IENTRY=4                                                            GMR   48
10    DO 100 I=1,NPRFLS                                                   GMR   49
      NSM1=NSECT(I)-1                                                     GMR   50
      DO 20 J=1,NSM1                                                      GMR   51
      K=J                                                                 GMR   52
      IF (X.LT.ATPI(J,I)) GO TO 30                                        GMR   53
20    CONTINUE                                                            GMR   54
      K=NSECT(I)                                                          GMR   55
```

```
      ISJ=ISHAPE(K,I)                                                    GMR  56
      IF (ISJ.GT.1) GO TO 70                                             GMR  57
      RAD(I)=PARAM(1,K,I)+PARAM(2,K,I)*X                                 GMR  58
      GO TO (50,100,100,40), IENTRY                                      GMR  59
40    DRV(I)=PARAM(2,K,I)                                                GMR  60
      GO TO 100                                                          GMR  61
50    RAT=RAD(I)/RTHCM(I)                                                GMR  62
      IF (JDIM.EQ.0.OR.NPRFLS.EQ.2) GO TO 60                             GMR  63
      S1(I)=RAT**2                                                       GMR  64
      S2(I)=2.*RAT/RTHCM(I)*PARAM(2,K,I)                                 GMR  65
      GO TO 140                                                          GMR  66
60    S1(I)=RAT                                                          GMR  67
      S2(I)=PARAM(2,K,I)/RTHCM(I)                                        GMR  68
      GO TO 100                                                          GMR  69
70    DAB=X-PARAM(2,K,I)                                                 GMR  70
      Q=SQRT(PARAM(3,K,I)**2-DAB**2)                                     GMR  71
      RAD(I)=PARAM(1,K,I)+SN(ISJ-1)*Q                                    GMR  72
      GO TO (80,100,100,80), IENTRY                                      GMR  73
80    DRV(I)=-SN(ISJ-1)*DAB/Q                                            GMR  74
      IF (IENTRY.EQ.4) GO TO 100                                         GMR  75
      RAT=RAD(I)/RTHCM(I)                                                GMR  76
      IF (JDIM.EQ.0.OR.NPRFLS.EQ.2) GO TO 90                             GMR  77
      S1(I)=RAT**2                                                       GMR  78
      S2(I)=2.*RAT/RTHCM(I)*DRV(I)                                       GMR  79
      GO TO 100                                                          GMR  80
90    S1(I)=RAT                                                          GMR  81
      S2(I)=DRV(I)/RTHCM(I)                                              GMR  82
100   CONTINUE                                                           GMR  83
      IF (NPRFLS.EQ.1) GO TO 130                                         GMR  84
      GO TO (120,130,110,110), IENTRY                                    GMR  85
110   SQTRSA=SQRT(RSA)                                                   GMR  86
      Y=RAD(1)*SQTRSA                                                    GMR  87
      Z=RAD(2)*SQTRSA                                                    GMR  88
      IF (IENTRY.EQ.3) GO TO 180                                         GMR  89
      DYDX=DRV(1)*SQTRSA                                                 GMR  90
      DZDX=DRV(2)*SQTRSA                                                 GMR  91
      GO TO 180                                                          GMR  92
120   ARATIO=S1(1)*S1(2)*RSA                                             GMR  93
      DERIVA=(S1(1)*S2(2)+S1(2)*S2(1))*RSA                               GMR  94
      GO TO 150                                                          GMR  95
130   IF (IENTRY.EQ.2) GO TO 160                                         GMR  96
140   ARATIO=S1(1)*RSA                                                   GMR  97
      DERIVA=S2(1)*RSA                                                   GMR  98
150   IF (ARATIO.LT.1..AND.ARATIO.GE.0.99999) ARATIO=1.                  GMR  99
      IF (ARATIO.GE.1..AND.X*DERIVA.GE.0.) GO TO 180                     GMR 100
      WRITE (6,190) X,ARATIO,DERIVA                                      GMR 101
      CALL DUMP (RNAME)                                                  GMR 102
      RETURN                                                             GMR 103
160   Y=RAD(1)                                                           GMR 104
      IF (JDIM.EQ.0) GO TO 170                                           GMR 105
      Y=Y*SQRT(RSA)                                                      GMR 106
      GO TO 180                                                          GMR 107
170   Y=Y*RSA                                                            GMR 108
180   RETURN                                                             GMR 109
C                                                                        GMR 110
```

```
C                                                              GMR 111
C                                                              GMR 112
190   FORMAT (40HOERROR IN INPUT DATA FOR NOZZLE GEOMETRY.4H   X=.1PE14.6GMR 113
     1.9H  ARATIO=.E14.6.9H  DERIVA=.E14.6)                    GMR 114
      END                                                      GMR 115-
```

```
      SUBROUTINE INGAS                                                  ING    1
      LOGICAL ERR                                                       ING    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   ING    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ,AIN                          ING    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             ING    5
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  ING    6
     1     C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  ING    7
     2  CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  ING    8
     3   CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  ING    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  ING   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  ING   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  ING   12
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,   SU,   SU2,  SUMG,  ING   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                 ING   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),ING   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),ING   16
     2CLN1MC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),ING   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),ING   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),ING   19
     5  SKIL(20),    SS(20),    TH(30),   TFA(20),   TFB(20),   TFC(20),ING   20
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) ING   21
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),         ING   22
     1 XNUI(64,20),XNUIJP(64,20)                                        ING   23
      COMMON       IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,  ING   24
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  ING   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,  ING   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,  ING   27
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        ING   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) ING   29
      COMMON ACOM,ELMENT,HP                                             ING   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBING   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                ING   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              ING   33
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       ING   34
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            ING   35
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       ING   36
      COMMON /EQC/ ZCAP(20),ZSEN,ZCP,ZCM,ZRBP,ZRHO                      ING   37
      COMMON /EQC2/ ZPZ,ZGMU(20)                                        ING   38
      COMMON /INGNE/ AIN(10,10)                                         ING   39
      COMMON /ERROR/ ERR                                                ING   40
      DO 10 I=1,ISC                                                     ING   41
      DO 10 J=1,ISC                                                     ING   42
   10 AIN(I,J)=LPIJ(I,J)                                                ING   43
      CALL MATINV (AIN,ISC,10)                                          ING   44
      IF (ERR) RETURN                                                   ING   45
      ISMC=ISS-ISC                                                      ING   46
      IF (ISMC.EQ.0) GO TO 60                                           ING   47
      DO 30 I=1,ISMC                                                    ING   48
      L=I+ISC                                                           ING   49
      DO 30 J=1,ISC                                                     ING   50
      CDIJ(I,J)=0.0                                                     ING   51
      DO 20 K=1,ISC                                                     ING   52
   20 CDIJ(I,J)=CDIJ(I,J)+AIN(K,J)*LPIJ(L,K)                            ING   53
   30 CONTINUE                                                          ING   54
      DO 50 I=1,ISMC                                                    ING   55
```

```
      BET(I)=-1.0                                                       ING   56
      DO 40 K=1,ISC                                                     ING   57
40    BET(I)=BET(I)+CDIJ(I,K)                                           ING   58
50    CONTINUE                                                          ING   59
60    SUM=0.0                                                           ING   60
      DO 80 I=1,ISC                                                     ING   61
      QM(I)=0.0                                                         ING   62
      DO 70 K=1,ISC                                                     ING   63
70    QM(I)=QM(I)+AIN(K,I)*CAPQ(K)                                      ING   64
      SUM=SUM+QM(I)                                                     ING   65
80    CONTINUE                                                          ING   66
      DO 90 I=1,ISC                                                     ING   67
      QM(I)=QM(I)/SUM                                                   ING   68
      CCI(I)=0.0                                                        ING   69
      DO 90 J=1,ISC                                                     ING   70
90    CCI(I)=CCI(I)+LPIJ(I,J)*CMW(J)                                    ING   71
      DO 100 I=1,ISC                                                    ING   72
100   CGI(I)=CCI(I)                                                     ING   73
      IF (ISMC.EQ.0) GO TO 130                                          ING   74
      DO 120 I=1,ISMC                                                   ING   75
      SUM=0.0                                                           ING   76
      DO 110 K=1,ISC                                                    ING   77
110   SUM=SUM+CDIJ(I,K)*CGI(K)                                          ING   78
      IL=I+ISC                                                          ING   79
      CGI(IL)=SUM                                                       ING   80
120   CONTINUE                                                          ING   81
130   RETURN                                                            ING   82
      ENTRY INTA                                                        ING   83
      CALL EQCALC (CTAP,PRESA)                                          ING   84
      IF (ERR) RETURN                                                   ING   85
      ZPA=ZPZ                                                           ING   86
      CM=ZCM                                                            ING   87
      SEN=ZSEN                                                          ING   88
      CHA=ZCH                                                           ING   89
      CMA=CM                                                            ING   90
      CRRB=SEN/CMA                                                      ING   91
      SEN=CRRB*CRA                                                      ING   92
      CH=CHA                                                            ING   93
      DO 140 I=1,ISS                                                    ING   94
      CGMU(I)=ZGMU(I)                                                   ING   95
      CAPX(I)=ZCAP(I)                                                   ING   96
140   GJA(I)=CAPX(I)/CMA                                                ING   97
      ROBARP=ZRHP                                                       ING   98
      RHAP=ZRHO                                                         ING   99
      ROBARA=ROPARP/RHAP                                                ING  100
      RETURN                                                            ING  101
      END                                                               ING  102-
```

```
      SUBROUTINE INIT                                                   INI    1
      REAL ACOM(30),ELMENT(10),HP(20)                                   INI    2
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              INI    3
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             INI    4
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO, INI    5
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP, INI    6
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX, INI    7
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1, INI    8
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT, INI    9
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB, INI   10
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT, INI   11
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG, INI   12
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                 INI   13
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),INI   14
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),INI   15
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),INI   16
     3   GJP(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),INI   17
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),INI   18
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),INI   19
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) INI   20
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),         INI   21
     1 XNUIJ(64,20),XNUIJP(64,20)                                       INI   22
      COMMON      IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1, INI   23
     1   ISMC,ISMCNR,   ISR,   ISS, ISSNP, ISSP1, ISSP2, ISSP3, ISSP4, INI   24
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    INI   25
     3 ISW5B, ISW6A, ISW6B,  IUPD,  IZERO,   JJK,    LC,    M1,  NFIT, INI   26
     4    NIT,   NNN,   NNS,   NQS,   NQT, NTEST                       INI   27
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) INI   28
      COMMON ACOV,ELMENT,HP                                             INI   29
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBINI   30
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                INI   31
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              INI   32
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       INI   33
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            INI   34
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       INI   35
      COMMON /MASSFL/ SMASS,CTMXXI,TSTOPI,IS(20)                        INI   36
      DIMENSION SPRP(43,30)                                             INI   37
      COMMON /POLYAT/ THLVE(4,20)                                       INI   38
      COMMON /SPEC/ SP1(43),SP2(43),SP3(43),SP4(43),SP5(43),SP6(43),    INI   39
     1      SP7(43),SP8(43),SP9(43),SP10(43),SP11(43),SP12(43),SP13(43),INI   40
     2      SP14(43),SP15(43),SP16(43),SP17(43),SP18(43),SP19(43),      INI   41
     3      SP20(43),SP21(43),SP22(43),SP23(43),SP24(43),SP25(43),      INI   42
     4      SP26(43),SP27(43),SP28(43),SP29(43),SP30(43)                INI   43
      EQUIVALENCE (SP1(1),SPRP(1,1))                                    INI   44
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),       INI   45
     1      KTR(25),ICH(20),IPA(25)                                     INI   46
      CTMXX=CTMXXI/CTAP                                                 INI   47
      IF (INEQV.NE.0) CTMXX=2.                                          INI   48
      TSTOP=TSTOPI/CTAP                                                 INI   49
      IF (TSTOP.EQ.0.) TSTOP=0.004                                      INI   50
      ARBA=5.0                                                          INI   51
      ARBB=1.0                                                          INI   52
      DATEST=1.01                                                       INI   53
      DBTEST=5.0                                                        INI   54
      PCTEST=1.2                                                        INI   55
```

```
      TEST=.000001                                                      INI   56
      TESTB=.00001                                                      INI   57
      CT=1.0                                                            INI   58
      CTE=1.0                                                           INI   59
      JJK=0                                                             INI   60
      NQT=0                                                             INI   61
      INEQ=0                                                            INI   62
      NTEST=100                                                         INI   63
      IM=1                                                              INI   64
      CRA=1.9872                                                        INI   65
      CRP=CRA CTAP                                                      INI   66
      CTPL=ALOG(CTAP)                                                   INI   67
      IF (NFIT) 10,30,10                                                INI   68
10    T1=CTAP                                                           INI   69
      T2=T1*T1                                                          INI   70
      T3=T1*T2                                                          INI   71
      T4=T1*T3                                                          INI   72
      DO 20 I=1,ISS                                                     INI   73
      IF (IGJ(I).EQ.0) GO TO 20                                         INI   74
      II=IS(I)                                                          INI   75
      TFB(I)=T1*SPRP(10,II)                                             INI   76
      TFC(I)=T2*SPRP(11,II)                                             INI   77
      TFD(I)=T3*SPRP(12,II)                                             INI   78
      TFE(I)=T4*SPRP(13,II)                                             INI   79
20    CONTINUE                                                          INI   80
30    DO 60 I=1,ISS                                                     INI   81
      IF (ETAJ(I).LE.0.) GO TO 60                                       INI   82
      II=IS(I)                                                          INI   83
      SAJ(I)=SPRP(17,II)+(1.5+ETAJ(I))*CTPL                             INI   84
      THEV(I)=SPRP(18,II)/CTAP                                          INI   85
      THEVE(1,I)=THEV(I)                                                INI   86
      DO 40 J=1,3                                                       INI   87
40    THEVE(J+1,I)=SPRP(J+40,II)/CTAP                                   INI   88
      K=IGM(I)                                                          INI   89
      DO 50 L=1,K                                                       INI   90
50    ELJ(L,I)=SPRP(L+3,II)/CRP                                         INI   91
60    CONTINUE                                                          INI   92
      DO 70 I=1,ISS                                                     INI   93
      II=IS(I)                                                          INI   94
70    SHJA(I)=SPRP(15,II)/CRP                                           INI   95
      ISMCNR=ISMC                                                       INI   96
      ISSNR=ISS                                                         INI   97
      RETURN                                                            INI   98
      END                                                               INI   99-
```

```
      SUBROUTINE KANDMU (ZK)                                            KAN    1
C     SUBROUTINE KANDMU COMPUTES VISCOSITY AND TRANSLATIONAL COMPONENT  KAN    2
C     OF THERMAL CONDUCTIVITY.  INDEX L=1 INDICATES TRANSLATIONAL       KAN    3
C     THERMAL CONDUCTIVITY. L=2 INDICATES VISCOSITY.                    KAN    4
      DIMENSION ZK(2)                                                   KAN    5
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              KAN    6
      COMMON /TRANS3/ B(20,2),BR(20,2),A(2),X(20),DH(20)                KAN    7
      COMMON /TRANS5/ N,IELEC,ID1,ID2                                   KAN    8
C     COMPUTE DIAGONAL MATRIX ELEMENTS CAP-A(I)                         KAN    9
      DO 20 I=1,N                                                       KAN   10
      B(I,1)=0.                                                         KAN   11
      DO 10 J=1,N                                                       KAN   12
10    B(I,1)=B(I,1)+X(J)*Q(3,I,J)                                       KAN   13
      B(I,2)=B(I,2)/ZM2(I)                                              KAN   14
      BR(I,1)=1./B(I,1)                                                 KAN   15
20    BR(I,2)=1./B(I,2)                                                 KAN   16
      IF (N.EQ.1) GO TO 70                                              KAN   17
C     COMPUTE AVERAGE OF NON-DIAGONAL MATRIX ELEMENTS SMALL-A(I,J)      KAN   18
      DO 60 L=1,2                                                       KAN   19
      U=0.                                                              KAN   20
      Y=0.                                                              KAN   21
      DO 40 I=2,N                                                       KAN   22
      I1=I-1                                                            KAN   23
      V=0.                                                              KAN   24
      W=0.                                                              KAN   25
      DO 30 J=1,I1                                                      KAN   26
      R=X(J)*(BR(I,L)-BR(J,L))**2                                       KAN   27
      V=V+R                                                             KAN   28
30    W=W+R*Q(L,I,J)                                                    KAN   29
      U=U+X(I)*W                                                        KAN   30
40    Y=Y+X(I)*V                                                        KAN   31
      IF (Y.EQ.0.) GO TO 50                                             KAN   32
      A(L)=U/Y                                                          KAN   33
      GO TO 60                                                          KAN   34
50    A(L)=0.                                                           KAN   35
60    CONTINUE                                                          KAN   36
70    A(1)=.5*A(1)                                                      KAN   37
C     COMPUTE VISCOSITY AND TRANSLATIONAL THERMAL CONDUCTIVITY          KAN   38
      DO 90 L=1,2                                                       KAN   39
      Y=0.                                                              KAN   40
      DO 80 I=1,N                                                       KAN   41
80    Y=Y+X(I)/(A(L)+B(I,L))                                            KAN   42
90    ZK(L)=3.125*Y/(1.-A(L)*Y)                                         KAN   43
      RETURN                                                            KAN   44
      END                                                               KAN   45-
```

```
      SUBROUTINE KINT (ZKINT)                                          KIN    1
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                             KIN    2
      COMMON /TRANS3/ B(20,2),BR(20,2),A(2),X(20),DH(20)               KIN    3
      COMMON /TRANS5/ N,IELEC,ID1,ID2                                  KIN    4
C     COMPUTE SUM OVER J OF X(J)*DELTA(I,J)                            KIN    5
      IF (N.EQ.1) GO TO 20                                             KIN    6
      N1=N+1                                                           KIN    7
      DO 10 I=2,N                                                      KIN    8
      I1=I-1                                                           KIN    9
      NI=N-I1                                                          KIN   10
      DO 10 J=1,I1                                                     KIN   11
      NJ=N1-J                                                          KIN   12
      DO 10 L=1,2                                                      KIN   13
      B(I,L)=B(I,L)+X(J)*Q(L,J,I)                                      KIN   14
10    B(NI,L)=B(NI,L)+X(NJ)*Q(L,NI,NJ)                                 KIN   15
C     COMPUTE THERMAL CONDUCTIVITY COMPONENT DUE TO INTERNAL EXCITATION KIN   16
C     ENERGY                                                           KIN   17
20    ZKINT=0.                                                         KIN   18
      DO 30 I=1,N                                                      KIN   19
30    ZKINT=ZKINT+DH(I)*X(I)/B(I,1)                                    KIN   20
      RETURN                                                           KIN   21
      END                                                              KIN   22-
```

```
      SUBROUTINE LIST                                                   LIS    1
      LOGICAL SUP3D,WEDGEM,AXISYM                                       LIS    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   LIS    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              LIS    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             LIS    5
      COMMON        A, AFNTS,   AFNX, AMACH,     AR,   ARBA,   ARBB, BZERO,  LIS    6
     1      C,   CARB,     CH,    CHA,  CLNT,     CM,    CMA,    CRA,   CRP,  LIS    7
     2   CRRB,    CRS,   CSTA,     CT,  CTAP,    CTB,    CTC, CTMAX, CTMXX,  LIS    8
     3    CTP,   CTPL,    CTT,     CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,   LIS    9
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,    ENT,   FLUX, HDELX,   PCT,   LIS   10
     5PCTEST,   PPES,  PRESA, PRESB,PRESTH,   PRHO,   RHAP,    RHO,  RHOB,  LIS   11
     6ROBAR,   RHOC,   RHOP,   RHPL, RHTH,ROBARA,ROBARP,   SCPG,    SDT,   LIS   12
     7    SEN,   SHPG,     SC,     SL,  SL64,     SM,     SU,    SU2,  SUMG,  LIS   13
     8   TEST,  TESTB,TPRINT, TSTOP,     UP,     ZP,    ZPA                 LIS   14
      COMMON          BE(64),    BET(20),   BLBK(31),    CAI(64),CAPXTH(20),LIS   15
     1   CCPJ(20), CEACT(64),    CGI(20),   CGMU(20),    CHI(64),  CHII(20),LIS   16
     2CLNIMC(64), CLNPI(64),    CMW(20),   ETAI(64),   ETAJ(20),    GJA(20),LIS   17
     3    GJB(20),PERTGJ(20),    PGJ(20),     PI(64), PICHI(64),     QM(20),LIS   18
     4     QQ(64),    SAJ(20), SDCHI(64),   SENT(20),    SHJ(20),   SHJA(20),LIS   19
     5   SKIL(20),     SS(20),     TB(30),    TFA(20),    TFB(20),    TFC(20),LIS   20
     6    TFD(20),    TFE(20),    TFK(20),   THEV(20), XMJAT(20),   XNUI(64) LIS   21
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),              LIS   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       LIS   23
      COMMON       IC,     IM,   INEQ,  INEQV,     IP,   IRUN,    ISC, ISCP1,  LIS   24
     1   ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,   LIS   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,   LIS   26
     3 ISW5B, ISW6A, ISW6B,  IUPD,  IZERO,    JJK,     LC,     M1,   NFIT,  LIS   27
     4     N,    NNN,    NNS,    NOS,    NQT, NTEST                      LIS   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20)   LPIJ(20,10) LIS   29
      COMMON ACOM,ELMENT,HP                                             LIS   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),D  (20),GJ(20),SBLIS   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                LIS   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              LIS   33
     1     (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),GJ(1)),        LIS   34
     2     (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            LIS   35
     3     (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       LIS   36
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,   CATFAC,TMODEL,XMODP1,DXMODP,   LIS   37
     1     TPLATE,KDIM                                                 LIS   38
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         LIS   39
     1     NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                 LIS   40
      COMMON /ROMAIN/ HS,SUPGO,MFITER                                   LIS   41
      COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,MBL                        LIS   42
      COMMON /NEWMP/ FACMP,NMODPT                                       LIS   43
      COMMON /RDLIST/ IGAS,IGASO,NOZZO,ICHANO                           LIS   44
      DIMENSION WORD(3,2)                                               LIS   45
      DATA WORD/6HTWO-DI,6HMENSIO,3HNAL,6HAXIALL,6HY-SYMM,5HETRIC/      LIS   46
      COMMON /LN/ ISATOM,ISMOL,JATOM,JMOL                               LIS   47
      COMMON /MASSFL/ SMASS,CTMXXI,TSTOPI,IS(20)                        LIS   48
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  LIS   49
     1     SW,RO,JDIM,IPCINT                                           LIS   50
      COMMON /RENE/ GAMIN,HTEST,TETEST,QTEST,DCHLL,DCHRAT,CCHI,TTEST,   LIS   51
     1     GTEST                                                       LIS   52
      COMMON /POLYAT/ THEVE(4,20)                                       LIS   53
      COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,     LIS   54
     1     NWX,NANGLE,NRADLE,WEDGEM,AXISYM,ISW9B                       LIS   55
```

```
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),
     1     KTR(25),ICH(20),IPA(25)
      COMMON /RDMOD/ LEWIS,IAMBIP
      DIMENSION COM1(2,2),COM2(2,2)
      DATA COM1 /' ',' ','(AMBIP','OLAR)'/, COM2 /' ',' ','(NOT U',
     1    'SED)'/
      IF (SUPGO) GO TO 190
      WRITE (6,350)
      WRITE (6,360) (ELMENT(I),CAPQ(I),CMW(I),I=1,ISC)
      WRITE (6,370)
      DO 30 I=1,ISR
      IF (QQ(I)) 20,10,20
10    WRITE (6,380) I,CAI(I),ETAI(I),CEACT(I),CCHI
      GO TO 30
20    WRITE (6,390) I,CAI(I),ETAI(I),CEACT(I),CCHI,(KUR(I,J),J=1,ISS)
30    CONTINUE
      WRITE (6,400)
      WRITE (6,410) (ELMENT(I),I=1,ISC)
      DO 40 I=1,ISS
      WRITE (6,420) HP(I),IGU(I),(LPIJ(I,J),J=1,ISC)
40    CONTINUE
      WRITE (6,430)
      WRITE (6,440) (HP(I),I=1,ISS)
      DO 50 I=1,ISR
      WRITE (6,450) I,(XNUIJP(I,J),J=1,ISS)
50    CONTINUE
      WRITE (6,460)
      WRITE (6,440) (HP(I),I=1,ISS)
      DO 60 I=1,ISR
      WRITE (6,450) I,(XNUIJ(I,J),J=1,ISS)
60    CONTINUE
      WRITE (6,470)
      DO 100 J=1,ISS
      IF (ETAJ(J).LE.0.) GO TO 100
      IF (IGJ(J)) 70,80,70
70    IF (CTMXX) 80,100,80
80    WRITE (6,480) HP(J),ETAJ(J),SBJ(J),THEVP(J),SHJAP(J),IGM(J)
      IF (ETAJ(J).LT.3.) GO TO 100
      DO 90 K=1,3
90    WRITE (6,550) THEVE(K+1,J)
100   CONTINUE
      WRITE (6,490)
      DO 130 J=1,ISS
      IF (ETAJ(J).LE.0.) GO TO 130
      IF (IGJ(J).EQ.IZERO) GO TO 110
      IF (CTMXX) 110,130,110
110   K=IGM(J)
      IF (K.GT.6) GO TO 120
      WRITE (6,500) HP(J),(GELJ(I,J),ELJ(I,J),I=1,K)
      GO TO 130
120   WRITE (6,500) HP(J),(GELJ(I,J),ELJ(I,J),I=1,6)
      WRITE (6,500) HP(J),(GELJ(I,J),ELJ(I,J),I=7,K)
130   CONTINUE
      IF (NT.EQ.1) GO TO 150
      WRITE (6,560)
```

```
      DO 140 I=1,ISR                                                          LIS 111
1     WRITE (6,570) I,KTF(I),KTR(I),ITR(I),EPAR(1,I),EPAR(2,I),IPA(I)         LIS 112
150   IF (NFIT) 160,190,160                                                   LIS 113
160   WRITE (6,510)                                                           LIS 114
      DO 180 J=1,ISS                                                          LIS 115
      IF (IGJ(J)) 170,180,170                                                 LIS 116
170   WRITE (6,520) HP(J),TFA(J),TFB(J),TFC(J),TFD(J),TFE(J),TFK(J),SHJALIS 117
     1P(J)                                                                    LIS 118
180   CONTINUE                                                                LIS 119
190   IF (INEQV.NE.0) WRITE (6,580)                                           LIS 120
      WRITE (6,590) HP(JATOM),HP(JMOL),(COM1(I,IAMBIP),I=1,2),(COM2(K,LELIS 121
     1WIS),K=1,2)                                                             LIS 122
      IF (ISW3B.NE.0) GO TO 200                                               LIS 123
      WRITE (6,530)                                                           LIS 124
      GO TO 210                                                               LIS 125
200   WRITE (6,540)                                                           LIS 126
210   IF (XMODP1.GE.9.E19.AND.NTS.EQ.0) GO TO 340                             LIS 127
      WRITE (6,600)                                                           LIS 128
      CXMAXI=CXMAX/2.54                                                       LIS 129
      IF (XMODP1.LT.9.E19) WRITE (6,630) NMODPT,XMODP1,CXMAXI                 LIS 130
      IF (NTS.EQ.0) GO TO 240                                                 LIS 131
      GO TO (220,230), NPRFLS                                                 LIS 132
220   WRITE (6,610) (TSDIAM(I),I=1,NTS)                                       LIS 133
      GO TO 240                                                               LIS 134
230   WRITE (6,62 ) (TSDIAM(I),I=1,NTS)                                       LIS 135
240   IF (   T.AXISYM) GO TO 290                                              LIS 136
      WRITE (6,640) TMODEL,TPLATE                                             LIS 137
      IF (FSTAG) 250,260,270                                                  LIS 138
250   WRITE (6,650)                                                           LIS 139
      GO TO 280                                                               LIS 140
260   WRITE (6,660)                                                           LIS 141
      GO TO 280                                                               LIS 142
270   WRITE (6,670)                                                           LIS 143
280   WRITE (6,680) CATFAC                                                    LIS 144
      KDP1=KDIM+1                                                             LIS 145
      WRITE (6,690) (WORD(I,KDP1),I=1,3)                                      LIS 146
290   IF (.NOT.WEDGEM) GO TO 340                                              LIS 147
      WRITE (6,700) (ANGLE(I),I=1,NANGLE)                                     LIS 148
      WRITE (6,710) (RADLE(I),I=1,NRADLE)                                     LIS 149
      IF (NWX.EQ.0) GO TO 300                                                 LIS 150
      WRITE (6,720) NWX,DWX,WX1                                               LIS 151
300   IF (WXI(1).GT.1.E19) GO TO 330                                          LIS 152
      DO 310 I=1,20                                                           LIS 153
      IF (WXI(I).GT.1.E19) GO TO 320                                          LIS 154
      NWXI=I                                                                  LIS 155
310   CONTINUE                                                                LIS 156
320   WRITE (6,730) (WXI(I),I=1,NWXI)                                         LIS 157
330   WRITE (6,740) TWEDGE,WK                                                 LIS 158
340   RETURN                                                                  LIS 159
C                                                                             LIS 160
C                                                                             LIS 161
C                                                                             LIS 162
350   FORMAT (1H ,99H                                    ELEMENT              LIS 163
     1  ATOM FRACTION          ELEMENT MOLECULAR WEIGHTS/)                   LIS 164
360   FORMAT (1H ,31X,A6,15X,1PE14.6,14X,E14.6)                               LIS 165
```

```
      FORMAT (1H ,//122H REACTION NO.       CONSTANT FACTOR AI      TEMP. LIS 166
     1POWER DEPENDENCE      ACTIVATION ENERGY        CHI TEST     THIRD BODLIS 167
     2Y MATRIX/)                                                          LIS 168
380   FORMAT (1H ,6X,I2,13X,1PE14.6,2(11X,E14.6),7X,0PF10.7)               LIS 169
390   FORMAT (1H ,6X,I2,13X,1PE14.6,2(11X,E14.6),7X,0PF10.7,4X,20I1)       LIS 170
400   FORMAT (1H ,//45H SPECIES  THERMAL FIT INDICATOR  ALPHA MATRIX/)     LIS 171
410   FORMAT (1H ,36X,10A6)                                                LIS 172
420   FORMAT (1H ,1X,A6,12X,I1,12X,10(3X,I3))                              LIS 173
430   FORMAT (1H ,//27H REACT NO.  NU PRIME MATRIX/)                       LIS 174
440   FORMAT (1H ,10X,20A6)                                                LIS 175
450   FORMAT (1H ,4X,I2,4X,20(3X,F3.0))                                    LIS 176
460   FORMAT (1H ,//21H REACT NO.  NU MATRIX/)                             LIS 177
470   FORMAT (1H ,//129H SPECIES   ATOMS PER MOLECULE          CHEMICAL COLIS 178
     1NSTANT     CHAR. VIBRATIONAL TEMP.   ENTHALPY OF FORMATION      ELLIS 179
     2ECTRONIC LEVELS/)                                                   LIS 180
480   FORMAT (1H ,1X,A6,7X,4(1PE14.6,11X),5X,I2)                           LIS 181
490   FORMAT (1H ,//49H SPECIES     (DEGENERACY,ELECTRONIC ENERGY LEVEL)/LIS 182
     1)                                                                   LIS 183
500   FORMAT (1H ,1X,A6,3X,6(0PF6.1,1PE14.6))                              LIS 184
510   FORMAT (1H ,//121H SPECIES                  A                B       LIS 185
     1      C                D                E                K      HEAT OF FOLIS 186
     2RMATION/)                                                           LIS 187
520   FORMAT (1H ,1X,A6,8X,1P7E15.6)                                       LIS 188
530   FORMAT (33H0BOUNDARY LAYER EFFECTS NEGLECTED)                        LIS 189
540   FORMAT (32H0BOUNDARY LAYER EFFECTS INCLUDED)                         LIS 190
550   FORMAT (65X,1PE14.6)                                                 LIS 191
560   FORMAT (//24X,42HELECTRON THERMAL NONEQUILIBRIUM PARAMETERS//8X,2HLIS 192
     1IR,7X,3HKTE,7X,3HKTR,7X,3HITR,10X,10HEPAR(1,IR),10X,10HEPAR(2,IR),LIS 193
     27X,3HIPA)                                                           LIS 194
570   FORMAT (4I10,1P2E20.4,I10)                                           LIS 195
580   FORMAT (59H0VIBRATIONAL EXCITATION FROZEN AT THE RESERVOIR TEMPERALIS 196
     1TURE)                                                               LIS 197
590   FORMAT (69H0LEWIS NUMBER CALCULATIONS BASED ON BINARY DIFFUSION COLIS 198
     1EFFICIENT FOR ,A4,3H - ,A4,5X,2A6,3X,2A6)                           LIS 199
600   FORMAT (///44H INPUT DATA FOR MODEL PARAMETER CALCULATIONS)          LIS 200
610   FORMAT (44H SPECIFIED TEST SECTION DIAMETERS IN INCHES=,10F8.2/44XLIS 201
     1,10F8.2)                                                            LIS 202
620   FORMAT (36H SPECIFIED CHANNEL WIDTHS IN INCHES=,10F8.2/36X,10F8.2)LIS 203
630   FORMAT (1H0,I2,48H MODEL POINTS IN A GEOMETRIC PROGRESSION FROM X=LIS 204
     1,1PE9.2,5H TO X=,E9.2,21H INCHES BEYOND THROAT)                     LIS 205
640   FORMAT (19H MODEL TEMPERATURE=,F6.0,31H DEG K, FLAT PLATE TEMPERATLIS 206
     1URE=,F6.0,6H DEG K)                                                 LIS 207
650   FORMAT (28H FROZEN SHOCK LAYER ON MODEL)                             LIS 208
660   FORMAT (33H EQUILIBRIUM SHOCK LAYER ON MODEL)                        LIS 209
670   FORMAT (61H BOTH EQUILIBRIUM AND FROZEN SHOCK LAYERS ON MODEL CALCLIS 210
     1ULATED)                                                             LIS 211
680   FORMAT (27H SURFACE CATALYTIC FACTOR =,F6.3)                         LIS 212
690   FORMAT (1X,3A6,15H MODEL GEOMETRY)                                   LIS 213
700   FORMAT (40H0WEDGE MODEL ANGLES OF ATTACK IN DEGREES,10F9.1)          LIS 214
710   FORMAT (29H LEADING-EDGE RADII IN INCHES,5F12.4)                     LIS 215
720   FORMAT (I4,57H DISTANCES FROM LEADING EDGE WITH A UNIFORM SEPARATILIS 216
     1ON OF,F5.2,20H INCHES, STARTING AT,F5.2,7H INCHES)                  LIS 217
730   FORMAT (48H SPECIFIED DISTANCES FROM LEADING EDGE IN INCHES,10F8.2LIS 218
     1/48X,10F8.2)                                                        LIS 219
740   FORMAT (27H WEDGE SURFACE TEMPERATURE=,F6.0,35H DEGREES K.  NOSE DLIS 220
```

```
      LIS 221
      LIS 222-
      IRAG COEFFICIENT=,F5.3)
      END
```

```
      SUBROUTINE MATINV (A,MM,NN)                                       MAT    1
C     MATRIX INVERSION                                                  MAT    2
C     INVERSE REPLACES ORIGINAL MATRIX                                  MAT    3
C     CALLING SEQUENCE    A - ORIGINAL MATRIX                           MAT    4
C                        MM - SIZE TO BE INVERTED                       MAT    5
C                        NN - FIRST SUBSCRIPT OF A IN DIMENSION         MAT    6
C                             STATEMENT OF MAIN PROGRAM                 MAT    7
      DOUBLE PRECISION A                                                MAT    8
      DATA RNAME /6HMATINV/                                             MAT    9
      DIMENSION A(NN,10),IV(10)                                         MAT   10
      M=MM                                                              MAT   11
      DO 10 I=1,M                                                       MAT   12
10    IV(I)=I                                                           MAT   13
      JA=1                                                              MAT   14
20    JJ=JA                                                             MAT   15
      K=JJ                                                              MAT   16
      K1=K+1                                                            MAT   17
      BIG=ABS(A(K,K))                                                   MAT   18
      IF (K1-M) 30,30,60                                                MAT   19
30    DO 50 I=K1,M                                                      MAT   20
      IF (BIG-ABS(A(I,JJ))) 40,50,50                                    MAT   21
40    K=I                                                               MAT   22
      BIG=ABS(A(I,JJ))                                                  MAT   23
50    CONTINUE                                                          MAT   24
60    BIG=A(K,JJ)                                                       MAT   25
      IF (BIG) 80,70,80                                                 MAT   26
70    CONTINUE                                                          MAT   27
      WRITE (6,300)                                                     MAT   28
      CALL DUMP (RNAME)                                                 MAT   29
      RETURN                                                            MAT   30
80    IF (K-JJ) 90,110,90                                               MAT   31
90    J=IV(K)                                                           MAT   32
      IV(K)=IV(JJ)                                                      MAT   33
      IV(JJ)=J                                                          MAT   34
      DO 100 J=1,M                                                      MAT   35
      B=A(JJ,J)                                                         MAT   36
      A(JJ,J)=A(K,J)                                                    MAT   37
100   A(K,J)=B                                                          MAT   38
110   A(JJ,JJ)=1.0                                                      MAT   39
      JA=JJ+1                                                           MAT   40
      DO 120 J=1,M                                                      MAT   41
120   A(JJ,J)=A(JJ,J)/BIG                                               MAT   42
      JB=JJ-1                                                           MAT   43
      IF (JB) 170,170,130                                               MAT   44
130   DO 160 I=1,JB                                                     MAT   45
      BB=A(I,JJ)                                                        MAT   46
      A(I,JJ)=0.0                                                       MAT   47
      IF (BB) 140,160,140                                               MAT   48
140   DO 150 J=1,M                                                      MAT   49
150   A(I,J)=A(I,J)-BB*A(JJ,J)                                          MAT   50
160   CONTINUE                                                          MAT   51
170   IF (JA-M) 180,180,220                                             MAT   52
180   DO 210 I=JA,M                                                     MAT   53
      BB=A(I,JJ)                                                        MAT   54
      A(I,JJ)=0.0                                                       MAT   55
```

```
      IF (BB) 190,210,190                                              MAT   56
  190 DO 200 J=1,M                                                     MAT   57
  200 A(I,J)=A(I,J)-BB*A(JJ,J)                                         MAT   58
  210 CONTINUE                                                         MAT   59
      GO TO 20                                                         MAT   60
  220 M1=M-1                                                           MAT   61
      IF (M1) 290,290,230                                              MAT   62
  230 DO 280 J=1,M1                                                    MAT   63
      DO 250 I=J,M                                                     MAT   64
      IF (IV(I)-J) 250,240,250                                         MAT   65
  240 K=I                                                              MAT   66
      GO TO 260                                                        MAT   67
  250 CONTINUE                                                         MAT   68
      WRITE (6,310)                                                    MAT   69
      CALL DUMP (RNAME)                                                MAT   70
      RETURN                                                           MAT   71
  260 IV(K)=IV(J)                                                      MAT   72
      DO 270 L=1,M                                                     MAT   73
      AA=A(L,J)                                                        MAT   74
      A(L,J)=A(L,K)                                                    MAT   75
  270 A(L,K)=AA                                                        MAT   76
  280 CONTINUE                                                         MAT   77
  290 RETURN                                                           MAT   78
C                                                                      MAT   79
C                                                                      MAT   80
  300 FORMAT (1H1,26H  MATINV, MATRIX SINGULAR /)                      MAT   81
  310 FORMAT (1H1,37HINDEXING OR STORAGE FAILURE IN MATINV/)           MAT   82
      END                                                              MAT   83-
```

```
      SUBROUTINE MODEL                                                  MOD    1
C     CALCULATES STAGNATION CONDITIONS BEHIND STRONG NORMAL SHOCK AND   MOD    2
C          CONDITIONS ON MODEL.                                         MOD    3
      LOGICAL LO,HI,DATAPE,WEDGEM,AXISYM,ERR,SUPGO,NOTRAN               MOD    4
      REAL ET,ETO                                                       MOD    5
      REAL ACOM(30),ELMENT(10),HP(20)                                   MOD    6
      DOUBLE PRECISION AA,CAPX,CDIJ,GJ                                  MOD    7
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             MOD    8
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  MOD    9
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  MOD   10
     2  CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  MOD   11
     3   CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  MOD   12
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  MOD   13
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  MOD   14
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  MOD   15
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  MOD   16
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                 MOD   17
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),MOD   18
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),MOD   19
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),MOD   20
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),MOD   21
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),MOD   22
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),MOD   23
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) MOD   24
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          MOD   25
     1 XNUIJ(64,20),XNUIJP(64,20)                                       MOD   26
      COMMON       IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,  MOD   27
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,   MOD   28
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    MOD   29
     3 ISW5B, ISW6A,    SB,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,   MOD   30
     4   NIT,   NNN,   NQS,   NQS,   NQT, NTEST                        MOD   31
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) MOD   32
      COMMON ACOM,ELMENT,HP                                             MOD   33
      DIMENSION GJ(20)                                                  MOD   34
      EQUIVALENCE (GJ(1),CAPX(1))                                       MOD   35
      COMMON /TEMPPY/ SAVEC(20)                                         MOD   36
      COMMON /STAG/ T2E,EPSLON,T2F,EPSF                                 MOD   37
      DIMENSION SHV(2,2)                                                MOD   38
      EQUIVALENCE (SHV(1,1),T2E)                                        MOD   39
      COMMON /BL/ DELPL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  MOD   40
     1      SW,RO,JDIM,IPCINT                                          MOD   41
      COMMON /TRPROP/ VISC,PR,SIGMA,FLEWIS                             MOD   42
      DIMENSION ELN(2),LIMSCO(2),QFRE(2),QFRF(2)                        MOD   43
      DATA ELN /0.52,0.63/, LIMSCO /20,19/                              MOD   44
      COMMON /ERROR/ ERR                                                MOD   45
      COMMON /EQC/ ZCAP(20),ZSEN,ZCH,ZCM,ZRBP,ZRHO                      MOD   46
      DIMENSION VGP(2),SOD(2),SQ2F(2)                                   MOD   47
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,   CATFAC,TMODEL,XMODP1,DXMODP,   MOD   48
     1      TPLATE,KDIM                                                MOD   49
      COMMON /CONVRT/ CF(6)                                             MOD   50
      COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAPE     MOD   51
      COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,     MOD   52
     1      NWX,NANGLE,NRADLE,WEDGEM,AXISYM,ISW9B                      MOD   53
      COMMON /WMOD/ TS                                                  MOD   54
      COMMON /OUTMOD/ TB7                                               MOD   55
```

```
      COMMON /RDMOD/ LEWIS,IAMBIP                                          MOD   56
      COMMON /COLDSP/ CGMW,QPJ(10),IJCS(10),NCS                            MOD   57
      COMMON /RDMAIN/ HS,SUPGC,MFITER,NOTRAN                               MOD   58
      DIMENSION QUANT(20,2)                                                MOD   59
      DATA QUANT /3HT2E,4HRT2E,4HPT2E,5HMUT2E,3HLEE,4HST2E,5HMWT2E,        MOD   60
     1     6HSIGT2E,3HPRE,5HEPSLE,4HQEFE,4HQEFF,5HQESRI,6HDELREF,          MOD   61
     2     6HDELREH,4HQEHE,4HQEHF,5HK2PFE,5HHRATE,3HQFP,3HT2F,4HRT2F,      MOD   62
     3     4HPT2F,5HMUT2F,3HLEF,4HST2F,5HMWT2F,6HSIGT2F,3HPRF,5HEPSLF,     MOD   63
     4     4HQFFE,4HQFFF,5HQFSRI,6HDELRFF,6HDELRFH,4HQFHE,4HQFHF,          MOD   64
     5     5HK2PFF,5HHRATF,1H   /                                          MOD   65
      DIMENSION TITLE2(3,2)                                                MOD   66
      DATA TITLE2 /6HEQUILI,6HBRIUM ,5HSHOCK,6HFROZEN,6H SHOCK,1H /        MOD   67
      DATA EQUALS /2H =/                                                   MOD   68
      DATA CATM /1.01325E6/                                                MOD   69
      DATA SQ2S /0.747,1./                                                 MOD   70
      NAMELIST /MODDMP/T2E,EPSLON,T2F,EPSF,PS,ZCH,ZCM,ZRHO,HS,R1,P1,       MOD   71
     1     ICOUNT,LO,HI,ERZ,ERRLO,ERRHI,TLO,THI,T2,P2,EPS,EPSOLD,H2,       MOD   72
     2     ALPHA,TS,DISC                                                   MOD   73
      DIMENSION SCOUT(20)                                                  MOD   74
      IF (.NOT.(AXISYM.OR.WEDGEM)) GO TO 480                               MOD   75
      IF (AMACH.GE.1.5) GO TO 10                                           MOD   76
      WRITE (6,490)                                                        MOD   77
      GO TO 480                                                            MOD   78
10    CONTINUE                                                             MOD   79
C     NORMAL SHOCK SOLUTION                                                MOD   80
      CALL ELTIME (ET0,0)                                                  MOD   81
      U1=SQRT(8.31436E7*CTAP/CMA)*SU                                       MOD   82
      R1=RHAP*RHO                                                          MOD   83
      HD=0.5*SU**2                                                         MOD   84
      HS=HD+CH                                                             MOD   85
      R1U12=R1*U1**2/CATM                                                  MOD   86
      P1=PRESA*PRES                                                        MOD   87
      PT=P1+R1U12                                                          MOD   88
      PC=4.*R1*R1U12                                                       MOD   89
      T1=CTAP*CT                                                           MOD   90
      CTSAVE=CT                                                            MOD   91
      CM1=82.055*R1*T1/P1                                                  MOD   92
      IF (EPSLON.GT.0.) GO TO 40                                           MOD   93
      EPSLON=0.01                                                          MOD   94
      IF (NOTRAN) GO TO 40                                                 MOD   95
      DO 30 I=1,ISS                                                        MOD   96
      DO 20 K=1,NCS                                                        MOD   97
      IF (IJCS(K).NE.I) GO TO 20                                           MOD   98
      SAVEC(I)=QPJ(K)                                                      MOD   99
      GO TO 30                                                             MOD  100
20    CONTINUE                                                             MOD  101
      SAVEC(I)=0.                                                          MOD  102
30    CONTINUE                                                             MOD  103
      CALL TRANSP (TMODEL,PRESA)                                           MOD  104
      PRW=PR                                                               MOD  105
      WRITE (6,500) PRW                                                    MOD  106
40    IF (EPSF.EQ.0.) EPSF=0.01                                            MOD  107
      DO 440 ISOLN=1,2                                                     MOD  108
      IF (ISOLN.EQ.1.AND..NOT.AXISYM) GO TO 440                            MOD  109
      IF (FSTAG.LT.0..AND.ISOLN.EQ.1) GO TO 440                            MOD  110
```

```
      EPS=SHV(2,ISOLN)                                                  MOD 111
      T2=SHV(1,ISOLN)                                                   MOD 112
      ICOUNT=0                                                          MOD 113
      LO=.FALSE.                                                        MOD 114
      HI=.FALSE.                                                        MOD 115
50    IF (ISOLN.EQ.1) GO TO 70                                          MOD 116
      DO 60 I=1,ISS                                                     MOD 117
60    SAVEC(I)=GJ(I)*CM1                                                MOD 118
      GO TO 110                                                         MOD 119
70    P2=P1    J12*(1.-EPS)                                             MOD 120
80    ICOUNT=ICOUNT+1                                                   MOD 121
      IF (ICOUNT.GT.30) GO TO 450                                       MOD 122
      CALL EQCALC (T2,P2)                                               MOD 123
      IF (ERR) RETURN                                                   MOD 124
      EPSOLD=EPS                                                        MOD 125
      EPS=R1/ZRHO                                                       MOD 126
      IF (ABS(EPS/EPSOLD-1.).LE.0.001) GO TO 130                        MOD 127
      ALPHA=P2/ZRHO                                                     MOD 128
      DISC=PT**2-PC*ALPHA                                               MOD 129
      IF (DISC.GE.0.) GO TO 100                                         MOD 130
      EPS=EPSOLD                                                        MOD 131
      IF (LO) GO TO 90                                                  MOD 132
      T2=0.9*T2                                                         MOD 133
      GO TO 50                                                          MOD 134
90    T2=0.5*(T2+TLO)                                                   MOD 135
      GO TO 50                                                          MOD 136
100   P2=0.5*(PT+SQRT(DISC))                                            MOD 137
      GO TO 80                                                          MOD 138
110   ICOUNT=ICOUNT+1                                                   MOD 139
      CT=T2/CTAP                                                        MOD 140
      IF (ICOUNT.GT.30) GO TO 450                                       MOD 141
      CALL THERM                                                        MOD 142
      ZCM=CM1                                                           MOD 143
      ZCH=0.                                                            MOD 144
      DO 120 I=1,ISS                                                    MOD 145
120   ZCH=ZCH+SAVEC(I)*SHJ(I)                                           MOD 146
      QP=P1*T2/T1                                                       MOD 147
      DISC=PT**2-4.*R1U12*QP                                            MOD 148
      IF (DISC.LT.0.) GO TO 450                                         MOD 149
      P2=0.5*(PT+SQRT(DISC))                                            MOD 150
      R2=R1*P2/QP                                                       MOD 151
      EPS=R1/R2                                                         MOD 152
130   H2=HS-HD*EPS**2                                                   MOD 153
      ZCH=ZCH*CMA/ZCM                                                   MOD 154
      IF (ABS(ZCH/H2-1.).LE.0.001) GO TO 170                            MOD 155
      ERZ=ZCH-H2                                                        MOD 156
      IF (ERZ) 140,140,150                                              MOD 157
140   ERRLO=ERZ                                                         MOD 158
      TLO=T2                                                            MOD 159
      LO=.TRUE.                                                         MOD 160
      IF (HI) GO TO 160                                                 MOD 161
      T2=1.1*T2                                                         MOD 162
      GO TO 50                                                          MOD 163
150   ERRHI=ERZ                                                         MOD 164
      THI=T2                                                            MOD 165
```

```
      HI=.TRUE.                                                          MOD 166
      IF (LO) GO TO 160                                                  MOD 167
      T2=0.9*T2                                                          MOD 168
      GO TO 50                                                           MOD 169
160   T2=T2-ERZ*(THI-TLO)/(ERRHI-ERRLO)                                  MOD 170
      GO TO 50                                                           MOD 171
C     SUBSONIC FLOW FROM THE SHOCK TO THE STAGNATION POINT ON THE MODEL  MOD 172
170   PS=P2+0.5*R1U12*EPS                                                MOD 173
      TS=T2                                                              MOD 174
      HI=.FALSE.                                                         MOD 175
      ICOUNT=0                                                           MOD 176
180   ICOUNT=ICOUNT+1                                                    MOD 177
      IF (ICOUNT.GT.20) GO TO 450                                        MOD 178
      GO TO (190,200), ISOLN                                             MOD 179
190   CALL EQCALC (TS,PS)                                                MOD 180
      IF (ERR) RETURN                                                    MOD 181
      GO TO 220                                                          MOD 182
200   CT=TS/CTAP                                                         MOD 183
      CALL THERM                                                         MOD 184
      ZCH=0.                                                             MOD 185
      DO 210 I=1,ISS                                                     MOD 186
210   ZCH=ZCH+SAVEC(I)*SHJ(I)                                            MOD 187
220   ZCH=ZCH*CMA/ZCM                                                    MOD 188
      IF (ABS(ZCH/HS-1.).LE.0.001) GO TO 260                             MOD 189
      ERZ=ZCH-HS                                                         MOD 190
      IF (ERZ) 230,230,240                                               MOD 191
230   TLO=TS                                                             MOD 192
      ERRLO=ERZ                                                          MOD 193
      IF (HI) GO TO 250                                                  MOD 194
      TS=1.1*TS                                                          MOD 195
      GO TO 180                                                          MOD 196
240   THI=TS                                                             MOD 197
      ERRHI=ERZ                                                          MOD 198
      HI=.TRUE.                                                          MOD 199
250   TS=TS-ERZ*(THI-TLO)/(ERRHI-ERRLO)                                  MOD 200
      GO TO 180                                                          MOD 201
260   IF (ISOLN.EQ.1) GO TO 280                                          MOD 202
      IF (.NOT.AXISYM.OR.FSTAG.EQ.0.) GO TO 300                          MOD 203
      ZSEN=-ALOG(PS)                                                     MOD 204
      DO 270 I=1,ISS                                                     MOD 205
      IF (SAVEC(I).LE.0.) GO TO 270                                      MOD 206
      ZSEN=ZSEN+SAVEC(I)*(SENT(I)-ALOG(SAVEC(I)))                        MOD 207
270   CONTINUE                                                           MOD 208
      ZRHO=R2*PS/P2*T2/TS                                                MOD 209
C     VGP(1) = VELOCITY GRADIENT PARAMETER FOR HEMISPHERE                MOD 210
C     VGP(2) = VELOCITY GRADIENT PARAMETER FOR FLAT-FACED MODEL          MOD 211
C     SOD(1) = STANDOFF DISTANCE FOR HEMISPHERE (CYLINDER IF KDIM=0)     MOD 212
C     SOD(2) = STANDOFF DISTANCE FOR FLAT-FACED MODEL                    MOD 213
280   VGP(1)=SQRT(2.*PS*CATM*(1.-P1/PS)/ZRHO)/U1                         MOD 214
      SQTE=SQRT(EPS)                                                     MOD 215
      IF (KDIM.EQ.0.) GO TO 290                                          MOD 216
      SOD(1)=0.78*EPS                                                    MOD 217
      IF (EPS.GT.0.1901) SOD(1)=SOD(1)*(1.+3.5*(EPS-0.19)**1.6)          MOD 218
      VGP(2)=EPS/(1.54+4.2*(SQTE-0.4)**2)                                MOD 219
      SOD(2)=1.12*EPS**0.39                                              MOD 220
```

```
      GO TO 300                                                          MOD 221
  290 SOD(1)=1./(2./EPS/((1.+0.5*EPS)*ALOG(1.3333333/EPS)-0.5*EPS)-1.)   MOD 222
      SOD(2)=-1.23*SQTE*ALOG(0.79*SQTE)                                  MOD 223
      VGP(2)=0.64*EPS                                                    MOD 224
  300 SHV(1,ISOLN)=T2                                                    MOD 225
      SHV(2,ISOLN)=EPS                                                   MOD 226
      IF (.NOT.AXISYM.OR.(ISOLN.EQ.2.AND.FSTAG.EQ.0.)) GO TO 440         MOD 227
      IF (ISOLN.EQ.2) GO TO 320                                          MOD 228
      DO 310 I=1,ISS                                                     MOD 229
  310 SAVEC(I)=ZCAP(I)                                                   MOD 230
  320 HCF=0.                                                             MOD 231
      IF (NOTRAN) GO TO 360                                              MOD 232
      DO 340 I=1,ISS                                                     MOD 233
      NEL=0                                                              MOD 234
      DO 330 J=1,ISC                                                     MOD 235
      IF (LPIJ(I,J).EQ.0) GO TO 330                                      MOD 236
      IF(LPIJ(I,J).NE.1) GO TO 340                                       MOD 236A
      NEL=NEL+1                                                          MOD 237
      IF (NEL.GT.1) GO TO 340                                            MOD 238
  330 CONTINUE                                                           MOD 239
      HCF=HCF+SAVEC(I)*SHJA(I)                                           MOD 240
  340 CONTINUE                                                           MOD 241
      HCF=HCF*CRP/ZCM                                                    MOD 242
      CALL TRANSP (TS,PS)                                                MOD 243
      IF (ERR) RETURN                                                    MOD 244
      HSTAG=CRP/CMA*ZCH                                                  MOD 245
      HW=CP*ALL*TMODEL                                                   MOD 246
      HRATIO=HCF/(HSTAG-HW)                                              MOD 247
      IF (LEWIS.EQ.2) HRATIO=0.                                          MOD 248
      QSRI=0.0750*SQRT(PS)*(HSTAG-HW)                                    MOD 249
C     0.075=1.8*0.0417. WHERE FACTOR 1.8 CONVERTS ENTHALPIES FROM CAL/GMMOD 250
C          TO BTU/LB                                                     MOD 251
      DO 350 L=1,2                                                       MOD 252
      QS=0.51/(PRW*PR)**0.3*SQ2F(KDIM+1)*SQRT(ZRHO*VISC*U1*VGP(L))*(HSTAMOD 253
     1G-HW)                                                              MOD 254
      QFRE(L)=QS*(1.+(FLEWIS**ELN(1)-1.)*HRATIO)                         MOD 255
  350 QFRF(L)=QS*(1.+(FLEWIS**ELN(2)*CATFAC-1.)*HRATIO)                  MOD 256
      CKS1=(EPS/(1.+EPS))**2*R1*U1/VISC*30.48                            MOD 257
  360 SCOUT(1)=TS                                                        MOD 258
      SCOUT(2)=CF(1)*ZRHO/RHAP                                           MOD 259
      SCOUT(3)=PS                                                        MOD 260
      SCOUT(4)=VISC*CF(3)                                                MOD 261
      SCOUT(5)=FLEWIS                                                    MOD 262
      SCOUT(6)=ZETEN*CRA/ZCM                                             MOD 263
      SCOUT(7)=ZCM                                                       MOD 264
      SCOUT(8)=SIGMA                                                     MOD 265
      SCOUT(9)=PR                                                        MOD 266
      SCOUT(10)=EPS                                                      MOD 267
      IF (NOTRAN) GO TO 370                                              MOD 268
      SCOUT(11)=QFRE(2)                                                  MOD 269
      SCOUT(12)=QFRF(2)                                                  MOD 270
      SCOUT(13)=QSRI                                                     MOD 271
      SCOUT(14)=SOD(2)                                                   MOD 272
      SCOUT(15)=SOD(1)                                                   MOD 273
      SCOUT(16)=QFRE(1)                                                  MOD 274
```

```
      SCOUT(17)=QFRF(1.                                                    MOD 275
      SCOUT(18)=CKS1                                                       MOD 276
      SCOUT(19)=HRATIO                                                     MOD 277
      IF (ISOLN.EQ.2) GO TO 390                                            MOD 278
      HCON=CRF/CMA                                                         MOD 279
      HE=HCON*CH                                                           MOD 280
      HR=HE+SQRT(PR)*HCON*(CHA-CH)                                         MOD 281
      HSTAR=0.5*(HE+HW)+0.22*(HR-HE)                                       MOD 282
      TSTAR=CTAP*HSTAR/HS/HCON                                             MOD 283
      RHOSTR=RHO*RHAP/TSTAR*CTAP*CT                                        MOD 284
      CALL TRANSP (TSTAR,PRES*PRESA)                                       MOD 285
      IF (ERR) RETURN                                                      MOD 286
      SCOUT(20)=0.222/PR**0.667*SQRT(RHOSTR*VISC*U1)*(HR-HW)               MOD 287
      GO TO 390                                                            MOD 288
  370 IF (.NOT.DATAPE) GO TO 390                                           MOD 289
      DO 380 N=11,20                                                       MOD 290
  380 SCOUT(N)=0.                                                          MOD 291
  390 WRITE (6,510) (TITLE2(I,ISOLN),I=1,3)                                MOD 292
      LMSC=LIMSCO(ISOLN)                                                   MOD 293
      IF (NOTRAN) LMSC=10                                                  MOD 294
      WRITE (6,520) (QUANT(I,ISOLN),EQUALS,SCOUT(I),I=1,LMSC)              MOD 295
      IF (ISOLN.EQ.2.OR.ISW6B.EQ.0) GO TO 420                              MOD 296
      WRITE (6,530) (TITLE2(I,ISOLN),I=1,3)                                MOD 297
      IF (IC.EQ.0) GO TO 400                                               MOD 298
      EMF=SAVEC(1)                                                         MOD 299
      SAVEC(1)=EMF*ZRHO/RHAP*TB7/ZCM                                       MOD 300
  400 ISS5P1=(ISS-1)/5+1                                                   MOD 301
      DO 410 K=1,ISS5P1                                                    MOD 302
      LIM1=5*(K-1)+1                                                       MOD 303
      LIM2=MIN0(5*K,ISS)                                                   MOD 304
  410 WRITE (6,540) (HP(I),SAVEC(I),I=LIM1,LIM2)                           MOD 305
      IF (IC.EQ.0) GO TO 420                                               MOD 306
      SAVEC(1)=EMF                                                         MOD 307
  420 IF (.NOT.DATAPE.OR.ISOLN.NE.1) GO TO 430                             MOD 308
      NRCOUT=NRCOUT+1                                                      MOD 309
      IMP=IMP+1                                                            MOD 310
      ITYPER=3                                                             MOD 311
      WRITE (ITPOUT) ITYPER,IMP,SCOUT(3),SCOUT(1),SCOUT(11),SCOUT(16),SCMOD 312
     1OUT(20),SCOUT(19),SCOUT(5),SCOUT(9),SCOUT(4),SCOUT(2)                MOD 313
  430 IF (FSTAG.EQ.C.AND..NOT.WEDGEM) GO TO 460                            MOD 314
  440 CONTINUE                                                             MOD 315
      GO TO 460                                                            MOD 316
  450 WRITE (6,550)                                                        MOD 317
      WRITE (6,MODDMP)                                                     MOD 318
  460 CT=CTSAVE                                                            MOD 319
      IF (NOTRAN) GO TO 470                                                MOD 320
      IF (WEDGEM) CALL WEDGE                                               MOD 321
  470 CALL ELTIME (ET,0)                                                   MOD 322
      DET=ET-ET0                                                           MOD 323
      WRITE (6,560) DET                                                    MOD 324
  480 RETURN                                                               MOD 325
C                                                                          MOD 326
CCC                                                                        MOD 327
                                                                           MOD 328
  490 FORMAT (64H0MODEL PARAMETER ROUTINE CALLED FOR A MACH NUMBER LESS    MOD 329
```

```
     1THAN 1.5.)                                                                    MOD  330
 500  FORMAT (80H0----------FROZEN PRANDTL NUMBER AT STAGNATION-POINT WAMOD  331
     1LL TEMPERATURE ON MODEL =,F7.4)                                               MOD  332
 510  FORMAT (46X,19HMODEL CONDITIONS - ,3A6)                                        MOD  333
 520  FORMAT (1X,A6,A2,F8.0,1CX,A6,A2,4X,1PE10.3,4X,A6,A2,4X,E10.3,4X,A6MOD  334
     1,A2,4X,E10.3,4X,A6,A2,CPF11.3/1X,A6,A2,0PF10.2,8X,A6,A2,F10.2,8X,AMOD  335
     26,A2,4X,1PE10.3,4X,A6,A2,0PF11.3,7X,A6,A2,F11.3/3(1X,A6,A2,F10.2,7MOD  336
     3X),2(1X,A6,A2,F11.3,6X)/3(1X,A6,A2,F10.2,7X),1X,A6,A2,F11.3,7X,A6,MOD  337
     4A2,F10.2)                                                                     MOD  338
 530  FORMAT (46X,17HMOLE FRACTIONS -- ,3A6)                                         MOD  339
 540  FORMAT (5(1X,A4,3X,1H=,4X,1PE10.3,3X))                                         MOD  340
 550  FORMAT (57H0ITERATION TO FIND STAGNATION CONDITIONS DID NOT CONVERMOD  341
     1GE)                                                                           MOD  342
 560  FORMAT (46H0ELAPSED TIME IN MODEL AND WEDGE CALCULATIONS=,F4.0,8H MOD  343
     1SECONDS)                                                                      MOD  344
      END                                                                           MOD  345-
```

```
      SUBROUTINE NEWRAP                                                 NEW    1
      LOGICAL ERR,NODEPS                                                NEW    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   NEW    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              NEW    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             NEW    5
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO  NEW    6
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  NEW    7
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  NEW    8
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DDTEST, DELT1,  NEW    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  NEW   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  NEW   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  NEW   12
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  NEW   13
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                 NEW   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),NEW   15
     1   CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),NEW   16
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),NEW   17
     3    GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),NEW   18
     4     QQ(64),    SAJ(20), SDCHI(64),   SENT(20),   SHJ(20),  SHJA(20),NEW   19
     5   SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),NEW    0
     6    TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) NEW    1
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          NEW   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       NEW   23
      COMMON      IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  NEW   24
     1   ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  NEW   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,  NEW   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,  NEW   27
     4    NIT,   NNN,   NNS,   NOS,   NOT, NTEST                       NEW   28
      COMMON    IGJ(20),    IGM(20),    ITB(5),    KUR(64,20),  LPIJ(20,10) NEW   29
      COMMON ACOM,ELMENT,HP                                             NEW   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBNEW   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                NEW   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              NEW   33
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       NEW   34
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            NEW   35
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       NEW   36
      COMMON /ERROR/ ERR                                                NEW   37
      DATA RNAME /6HNEWRAP/                                             NEW   38
      CALL THERM                                                        NEW   39
      NODEPS=ISMCNR.EQ.0                                                NEW   40
      IF (NODEPS) GO TO 30                                              NEW   41
      DO 20 I=1,ISMCNR                                                  NEW   42
      IL=I+ISC                                                          NEW   43
      CHII(I)=-XMJAT(IL)                                                NEW   44
      DO 10 J=IM,ISC                                                    NEW   45
      CHII(I)=CHII(I)+CDIJ(I,J)*XMJAT(J)                                NEW   46
   10 CONTINUE                                                          NEW   47
   20 CONTINUE                                                          NEW   48
   30 DO 40 I=IM,ISSNR                                                  NEW   49
      CAPX(I)=GJ(I)*CM                                                  NEW   50
   40 PGJ(I)=SENT(I)-CRRB*CGI(I)                                        NEW   51
      IF (NODEPS) GO TO 60                                              NEW   52
      DO 50 J=1,ISMCNR                                                  NEW   53
      I=J+ISC                                                           NEW   54
   50 SKIL(J)=CAPX(I)                                                   NEW   55
```

C. 6

ORIGINAL PAGE IS
OF POOR QUALITY

```
60      NIT=0                                                                   NEW   56
        M2=ISC+2                                                                NEW   57
        ZP=ALOG(PRES)                                                           NEW   58
70      DO 90 J=IM,ISC                                                          NEW   59
        AA(J,M2)=QM(J)-CAPX(J)                                                  NEW   60
        IF (NODEPS) GO TO 90                                                    NEW   61
        DO 80 I=1,ISMCNR                                                        NEW   62
80      AA(J,M2)=AA(J,M2)-(CDIJ(I,J)-QM(J)*BET(I))*SKIL(I)                      NEW   63
90      CONTINUE                                                                NEW   64
        AA(M1,M2)=-ZP-ZPA                                                       NEW   65
        DO 100 L=IM,ISSNR                                                       NEW   66
100     AA(M1,M2)=AA(M1,M2)+CAPX(L)*(PGJ(L)-CGMU(L))                            NEW   67
        DO 150 J=IM,ISC                                                         NEW   68
        DO 150 K=IM,ISC                                                         NEW   69
        IF (J-K) 120,110,120                                                    NEW   70
110     AA(J,K)=CAPX(J)                                                         NEW   71
        GO TO 130                                                               NEW   72
120     AA(J,K)=0.0                                                             NEW   73
130     IF (NODEPS) GO TO 150                                                   NEW   74
        DO 140 I=1,ISMCNR                                                       NEW   75
140     AA(J,K)=AA(J,K)+CDIJ(I,K)*SKIL(I)*(CDIJ(I,J)-QM(J)*BET(I))              NEW   76
150     CONTINUE                                                                NEW   77
        DO 170 K=IM,ISC                                                         NEW   78
        AA(M1,K)=(1.-PGJ(K)+CGMU(K))*CAPX(K)                                    NEW   79
        IF (NODEPS) GO TO 170                                                   NEW   80
        DO 160 I=1,ISMCNR                                                       NEW   81
        L=I+ISC                                                                 NEW   82
160     AA(M1,K)=AA(M1,K)+CDIJ(I,K)*(1.-PGJ(L)+CGMU(L))*CAPX(L)                 NEW   83
170     CONTINUE                                                                NEW   84
        AA(M1,M1)=1.0                                                           NEW   85
        IF (NODEPS) GO TO 190                                                   NEW   86
        DO 180 I=1,ISMCNR                                                       NEW   87
        L=I+ISC                                                                 NEW   88
180     AA(M1,M1)=AA(M1,M1)+BET(I)*(1.-PGJ(L)+CGMU(L))*CAPX(L)                  NEW   89
190     DO 210 J=IM,ISC                                                         NEW   90
        AA(J,M1)=0.0                                                            NEW   91
        IF (NODEPS) GO TO 210                                                   NEW   92
        DO 200 I=1,ISMCNR                                                       NEW   93
        L=I+ISC                                                                 NEW   94
200     AA(J,M1)=AA(J,M1)+BET(I)*CAPX(L)*(CDIJ(I,J)-QM(J)*BET(I))               NEW   95
210     CONTINUE                                                                NEW   96
        IF (JJK) 240,220,240                                                    NEW   97
220     IF (CAPX(1)-1.0D-20) 230,230,270                                        NEW   98
230     JJK=1                                                                   NEW   99
        ISMCNR=ISMCNR-IC                                                        NEW  100
        ISSNR=ISSNR-IC                                                          NEW  101
        NODEPS=ISMCNR.EQ.0                                                      NEW  102
        IM=IM+1                                                                 NEW  103
240     AA(1,1)=1.0                                                             NEW  104
        DO 250 J=2,M2                                                           NEW  105
250     AA(1,J)=0.0                                                             NEW  106
        CAPX(1)=0.0                                                             NEW  107
        LL=ISSNR+1                                                              NEW  108
        DO 260 I=LL,ISS                                                         NEW  109
260     CAPX(I)=0.0                                                             NEW  110
```

```
2.      CALL DSMSOL (AA,M1,22)                                          NEW 111
        IF (ERR) RETURN                                                 NEW 112
        NIT=NIT+1                                                       NEW 113
        DO 340 K=IM,ISC                                                 NEW 114
        ZB=1.+AA(K,M2)                                                  NEW 115
        IF (ZB) 280,280,310                                             NEW 116
280     CAPX(K)=CAPX(K)/2.                                              NEW 117
        IF (CAPX(K)) 290,300,290                                        NEW 118
290     CGMU(K)=ALOG(CAPX(K))                                           NEW 119
        GO TO 340                                                       NEW 120
300     WRITE (6,480) K                                                 NEW 121
        CGMU(K)=CGMU(K)-0.69314718                                      NEW 122
        GO TO 340                                                       NEW 123
310     CAPX(K)=CAPX(K)*ZB                                              NEW 124
        IF (CAPX(K)) 320,330,320                                        NEW 125
320     CGMU(K)=ALOG(CAPX(K))                                           NEW 126
        GO TO 340                                                       NEW 127
330     WRITE (6,480) K                                                 NEW 128
        CGMU(K)=CGMU(K)+ALOG(ZB)                                        NEW 129
340     CONTINUE                                                        NEW 130
        ZC=1.+AA(M1,M2)                                                 NEW 131
        IF (ZC) 350,350,360                                             NEW 132
350     PRES=PRES/2.                                                    NEW 133
        IF (PRES) 370,380,370                                           NEW 134
360     PRES=PRES*ZC                                                    NEW 135
        IF (PRES) 370,390,370                                           NEW 136
370     ZP=ALOG(PRES)                                                   NEW 137
        GO TO 400                                                       NEW 138
380     WRITE (6,490)                                                   NEW 139
        ZP=ZP-0.69314718                                                NEW 140
        GO TO 400                                                       NEW 141
390     WRITE (6,490)                                                   NEW 142
        ZP=ZP+ALOG(ZC)                                                  NEW 143
400     IF (NODEPS) GO TO 430                                           NEW 144
        DO 420 I=1,ISMCNR                                               NEW 145
        SKIL(I)=CHII(I)+BET(I)*(ZP+ZPA)                                 NEW 146
        DO 410 L=IM,ISC                                                 NEW 147
410     SKIL(I)=SKIL(I)+CDIJ(I,L)*CGMU(L)                               NEW 148
        J=I+ISC                                                         NEW 149
        CGMU(J)=SKIL(I)                                                 NEW 150
        SKIL(I)=EXP(SKIL(I))                                            NEW 151
        CAPX(J)=SKIL(I)                                                 NEW 152
420     CONTINUE                                                        NEW 153
430     DO 450 K=IM,M1                                                  NEW 154
        IF (ABS(AA(K,M2)).LE.TEST) GO TO 450                            NEW 155
        IF (NIT-NTEST) 70,70,440                                        NEW 156
440     WRITE (6,500)                                                   NEW 157
        CALL DUMP (RNAME)                                               NEW 158
        RETURN                                                          NEW 159
450     CONTINUE                                                        NEW 160
        CH=0.0                                                          NEW 161
        CM=0.0                                                          NEW 162
        DO 460 I=IM,ISSNR                                               NEW 163
        CM=CM+CAPX(I)*CGI(I)                                            NEW 164
460     CH=CH+CAPX(I)*SHJ(I)                                            NEW 165
```

```
      RHOBAR=PRES*CM/(CMA*CT)                                          NEW 166
      RHO=(RHOBAR*RUBARA)/(1.+RHOBAR*ROBARP*BZERO)                     NEW 167
      CH=CH*(CMA/CM)                                                   NEW 168
      CH=CH+(PRES/RHOBAR)*(RHOBAR*(ROBARA/RHO)-1.)                     NEW 169
      DO 470 I=IM,ISSNR                                                NEW 170
470   GJ(I)=CAPX(I)/CM                                                 NEW 171
      SU2=2.*(CHA-CH)                                                  NEW 172
      SU=SQRT(SU2)                                                     NEW 173
      FLUX=RHO*SU                                                      NEW 174
      RETURN                                                           NEW 175
C                                                                      NEW 176
C                                                                      NEW 177
C                                                                      NEW 178
480   FORMAT (16H IN NEWRAP,CAPX(,I2,3H)=0)                            NEW 179
490   FORMAT (14H IN NEWRAP,P=0)                                       NEW 180
500   FORMAT (35H1TOO MANY NEWTON-RAPHSON ITERATIONS)                  NEW 181
      END                                                              NEW 182-
```

```
      SUBROUTINE NEXTMP (ITS,XMOD1,XMODEL)                                  NEX    1
      LOGICAL ERR                                                           NEX    2
      COMMON /ERROR/ ERR                                                    NEX    3
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),             NEX    4
     1     NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                      NEX    5
      COMMON /NEWMP/ FACMP,NMODPT                                           NEX    6
      COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,MBL                            NEX    7
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,    CATFAC,TMODEL,XMODP1,DXMODP,      NEX    8
     1     TPLATE,KDIM                                                      NEX    9
      DATA ONE /1./                                                         NEX   10
      IF (NPRFLS.EQ.1) GO TO 20                                             NEX   11
      IF (ITS.LE.NTS) GO TO 10                                              NEX   12
      XMODEL=1.E30                                                          NEX   13
      GO TO 50                                                              NEX   14
10    CALL FINDXC (TSAR(ITS),MBL,XMOD2)                                     NEX   15
      IF (ERR) RETURN                                                       NEX   16
      GO TO 40                                                              NEX   17
20    IF (ITS.GT.NTS) GO TO 30                                              NEX   18
      CALL FINDX (TSAR(ITS),ONE,XMOD2)                                      NEX   19
      IF (ERR) RETURN                                                       NEX   20
      IF (XMOD1-XMOD2) 30,30,40                                             NEX   21
30    XMODEL=XMOD1                                                          NEX   22
      XMOD1=XMOD1*FACMP                                                     NEX   23
      GO TO 50                                                              NEX   24
40    XMODEL=XMOD2                                                          NEX   25
      ITS=ITS+1                                                             NEX   26
50    RETURN                                                                NEX   27
      END                                                                   NEX   28
```

```
      SUBROUTINE NONEQ                                                  NON    1
      REAL ACOM(30),ELMENT(10),HP(20)                                   NON    2
      LOGICAL ERR,MODLPT,XMSET,FAILED,COUPLD,AXISYM,WEDGEM,NOREAC,DATAPENON    3
      DOUBLE PRECISION AA,AAA,CJ,GJ,CAPX,DCI,CI,GSQ,DCA,AIN,CDIJ,ONE    NON    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             NON    5
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,  NON    6
     1      C,  CARB,    CH,   CHA,  CLNT,     CM,   CMA,   CRA,   CRP,  NON    7
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,    CTB,   CTC, CTMAX, CTMXX,  NON    8
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,   NON    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,   NON   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,   NON   11
     6RHOBAR,  RHOC,  RHOP,  RHPL, RHTH,ROBARA,ROBARP,  SCPG,   SDT,    NON   12
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,   NON   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  NON   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),NON   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),NON   16
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),NON   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),   QM(20),NON   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),NON   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),NON   20
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),  XNUI(64)  NON   21
      COMMON            BETA(64,20),   ELJ(10,20),  GELJ(10,20),        NON   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       NON   23
      COMMON      IC,    IM,   INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,  NON   24
     1   ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  NON   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    NON   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,   NON   27
     4    NIT,   NNN,   NNS,   NQS,   NQT, NTEST                       NON   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) NON   29
      COMMON ACOM,ELMENT,HP                                             NON   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBNON   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                NON   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              NON   33
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       NON   34
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            NON   35
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       NON   36
      DIMENSION CI(10),DCI(10),DCA(10),GSQ(20)                          NON   37
      EQUIVALENCE (GSQ(1),AA(1,24))                                     NON   38
      EQUIVALENCE (CCI(1),AA(1,23)), (DCA(1),AA(11,23))                 NON   39
      COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAPE     NON   40
      COMMON /INGNE/ AIN(10,10)                                         NON   41
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  NON   42
     1      SW,R0,JDIM,IPCINT                                           NON   43
      COMMON /NEQ/ OMDST(2),DDELBL(2)                                   NON   44
      COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,MBL                        NON   45
      COMMON /ERROR/ ERR                                                NON   46
      COMMON /CERROR/ FAILED                                            NON   47
      COMMON /BLNE/ XI(2),ORDINP(2),DELBLP(2),DDBLP(2),AMPP,AMP,        NON   48
     1      COUPLD,ISMD                                                 NON   49
      COMMON /BLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOT(2),      NON   50
     1      TAUW(2)                                                     NON   51
      DIMENSION BLNEAR(10), ISUPR(64),CAISV(64)                         NON   52
      EQUIVALENCE (BLNEAR(1),XI(1)), (ISUPR(1),AA(1,1))                 NON   53
      COMMON /RENE/ GAMIN,HTEST,TETEST,QTEST,DCHLL,DCHRAT,CCHI,TTEST,   NON   54
     1      GTEST                                                       NON   55
```

```
      COMMON /GLIM2/ NOREAC(64)                                          NON   56
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),          NON   57
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                 NON   58
      DATA RNAME /5HNONEQ/                                               NON   59
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,    CATFAC,TMODEL,XMODP1,DXMODP,   NON   60
     1      TPLATE,KDIM                                                 NON   61
      COMMON /STAG/ T2E,EPSLON,T2F,EPSF                                  NON   62
      COMMON /MODP2/ MODLPT                                              NON   63
      COMMON /THRT/ RSA                                                  NON   64
      COMMON /SWITCH/ XPB,DXPB,W                                         NON   65
      COMMON /TEMPRY/ SAVEC(20)                                          NON   66
      COMMON /AVG/ WSAVE                                                 NON   67
      COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,      NON   68
     1      NWX,NANGLE,NRADLE,WEDGEM,AXISYM,ISW9B                       NON   69
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),        NON   70
     1      KTR(25),ICH(20),IPA(25)                                     NON   71
      COMMON /TNERK/ SDTE,CTEB,DCHA,CHB,SDCHA,DQMAX,IFAIL                NON   72
      COMMON /STEPS/ NSTEPS                                              NON   73
      COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE                                NON   74
      COMMON /AEGEOM/ SQRTA,S1,S2                                        NON   75
      COMMON /SS/ CAS,US                                                 NON   76
      DIMENSION DELBLB(2),IF(30)                                         NON   77
      DATA TENTH /10000./, ONE /1.D0/                                    NON   78
      WRITE (6,1080)                                                     NON   79
C     INITIALIZATIONS                                                    NON   80
      W=WSAVE                                                            NON   81
      DQMAX=0.                                                           NON   82
      QEOLD=0.                                                           NON   83
      QDPE=0.                                                            NON   84
      NSTEPS=0                                                           NON   85
      ISTEPS=0                                                           NON   86
      ISUBD=0                                                            NON   87
      T2E=CTAP                                                           NON   88
      EPSLON=0.                                                          NON   89
      T2F=CTAP                                                           NON   90
      EPSF=0.                                                            NON   91
      MODLPT=.FALSE.                                                     NON   92
      XMSET=.FALSE.                                                      NON   93
      XMODEL=1.E30                                                       NON   94
      IRA=0                                                              NON   95
      ALOGT=ALOG(TENTH)                                                  NON   96
      ICOUNT=0                                                           NON   97
      DO 10 I=1,ISR                                                      NON   98
10    NOREAC(I)=.FALSE.                                                  NON   99
      IPOINT=1                                                           NON  100
      IS5=ISW5B                                                          NON  101
      DELTSV=DELT1                                                       NON  102
      IF (ISW3B.NE.0) DELT1=AMIN1(0.049*(1.-CTMAX),DELT1)                NON  103
      HDELX=0.5*DELTAX                                                   NON  104
      DXOLD=DELTAX                                                       NON  105
      DO 20 I=1,NPRFLS                                                   NON  106
20    DDELBL(I)=0.                                                       NON  107
      CTT=1.0                                                            NON  108
      CRS=SEN                                                            NON  109
      SC=1.1                                                             NON  110
```

```
      SCD=2.                                                          NON 111
      ITB(1)=0                                                        NON 112
      NNS=0                                                           NON 113
      NNN=0                                                           NON 114
      ISCP1=ISC+1                                                     NON 115
      ISSP1=ISS+1                                                     NON 116
      ISSP2=ISS+2                                                     NON 117
      ISSP3=ISS+3                                                     NON 118
      ISSP4=ISS+4                                                     NON 119
      INEQ=0                                                          NON 120
      DO 30 I=1,ISR                                                   NON 121
      PI(I)=1.0                                                       NON 122
      ZZZ=CAI(I)                                                      NON 123
      CAI(I)=ALOG(ZZZ)-ETAI(I)*ALOGT                                  NON 124
      CAISV(I)=CAI(I)                                                 NON 125
      BE(I)=0.0                                                       NON 126
      XNUI(I)=0.0                                                     NON 127
      DO 30 J=1,ISS                                                   NON 128
      BETA(I,J)=XNUIJP(I,J)-XNUIJ(I,J)                                NON 129
      BE(I)=BE(I)+BETA(I,J)                                           NON 130
30    XNUI(I)=XNUI(I)+XNUIJ(I,J)                                      NON 131
      ZZZ=CMA/CRP                                                     NON 132
      CSTA=.5*ALOG(ZZZ)                                               NON 133
      SL64=SL/6468.                                                   NON 134
C     COMPUTE AND TEST RANK OF BETA MATRIX                            NON 135
      IX=0                                                            NON 136
      I=0                                                             NON 137
      I2=1                                                            NON 138
40    I=I+1                                                           NON 139
      IF (I-ISS) 50,50,150                                            NON 140
50    DO 70 J=I2,ISR                                                  NON 141
      IF (BETA(J,I)) 60,70,60                                         NON 142
60    KA=J                                                            NON 143
      IX=IX+1                                                         NON 144
      GO TO 80                                                        NON 145
70    CONTINUE                                                        NON 146
      GO TO 40                                                        NON 147
80    K=KA                                                            NON 148
      DO 90 M=1,ISS                                                   NON 149
      AB=BETA(K,M)                                                    NON 150
      BETA(K,M)=BETA(IX,M)                                            NON 151
90    BETA(I,M)=AB                                                    NON 152
      I2=IX+1                                                         NON 153
      IF (I-ISS) 100,100,150                                          NON 154
100   DO 130 J=I2,ISR                                                NON 155
      CA=BETA(J,I)/BETA(IX,I)                                         NON 156
      IF (CA) 110,140,110                                             NON 157
110   DO 130 N=I,ISS                                                  NON 158
      BETA(M,N)=BETA(M,N)-CA*BETA(IX,N)                               NON 159
      IF (ABS(BETA(M,N))-0.01) 120,120,130                            NON 160
120   BETA(M,N)=0.0                                                   NON 161
130   CONTINUE                                                        NON 162
140   CONTINUE                                                        NON 163
      GO TO 40                                                        NON 164
150   IF (IX-ISMC) 160,170,160                                        NON 165
```

```
160   WRITE (6,1090)                                                    NON 166
C     ERROR EXIT NO. 1 ***************************************************NON 167
C     BETA MATRIX OF INCORRECT RANK                                     NON 168
      IERR=1                                                            NON 169
      GO TO 1000                                                        NON 170
C     RESET BETA MATRIX (DESTROYED DURING RANK CALCULATION)             NON 171
170   DO 180 I=1,ISR                                                   NON 172
      DO 180 J=1,ISS                                                    NON 173
180   BETA(I,J)=XNUIJP(I,J)-XNUIJ(I,J)                                  NON 174
      AR=0.0                                                            NON 175
      LC=0                                                              NON 176
      DELT=DELT1                                                        NON 177
      DO 200 I=1,ISC                                                    NON 178
      CI(I)=0.                                                          NON 179
      DO 190 J=1,ISS                                                    NON 180
190   CI(I)=CI(I)+GJA(J)*LPIJ(J,I)                                      NON 181
200   CONTINUE                                                          NON 182
      GO TO 220                                                         NON 183
C     END OF INITIALIZATIONS                                            NON 184
210   IF (MODLPT) CALL NEXTMP (ITS,XMOD1,XMODEL)                        NON 185
      GO TO 230                                                         NON 186
220   CALL DERIVS                                                       NON 187
      IF (ERR) RETURN                                                   NON 188
      IF (.NOT.FAILED) GO TO 230                                        NON 189
C     ERROR EXIT NO. 2 ***************************************************NON 190
      IERR=2                                                            NON 191
      GO TO 1000                                                        NON 192
230   IF (XMSET.OR.AMACH.LT.1.5) GO TO 240                              NON 193
      XMSET=.TRUE.                                                      NON 194
      XMOD1=XMP1                                                        NON 195
      ITS=1                                                             NON 196
      CALL NEXTMP (ITS,XMOD1,XMODEL)                                    NON 197
240   IF (IUPD) 250,350,250                                             NON 198
C     TESTS FOR SWITCH TO DOWNSTREAM REGION                             NON 199
250   IF (DLOGA) 450,260,260                                            NON 200
260   IF (CX) 450,450,270                                               NON 201
270   IF (AR) 300,280,300                                               NON 202
280   AR=1.0                                                            NON 203
      DO 290 I=1,NPRFLS                                                 NON 204
290   OMDST(I)=1.-DELBL(I)                                              NON 205
300   IF (AFNX.LT.DATEST) GO TO 450                                     NON 206
C     SAVE DATA AT SWITCH POINT FOR POSSIBLE RESTART                    NON 207
      TB(1)=CX                                                          NON 208
      TB(2)=CT                                                          NON 209
      TB(3)=SC                                                          NON 210
      TB(4)=CTT                                                         NON 211
      TB(5)=DELTAX                                                      NON 212
      TB(6)=HDELX                                                       NON 213
      TB(7)=CTE                                                         NON 214
      TB(8)=SCD                                                         NON 215
      TB(9)=CHA                                                         NON 216
      ITB(1)=INEQ                                                       NON 217
      ITB(2)=NNS                                                        NON 218
      ITB(3)=NNN                                                        NON 219
      ITB(4)=NRCOUT                                                     NON 220
```

```
      DO 310 I=1,ISS                                                    NON 221
310   TB(I+10)=GJ(I)                                                    NON 222
      BLBK(1)=RSA                                                       NON 223
      IF (ISW38.EQ.0) GO TO 340                                         NON 224
      DO 320 I=1,10                                                     NON 225
320   BLBK(I+1)=BLNEAR(I)                                               NON 226
      BLBK(12)=XPB                                                      NON 227
      BLBK(13)=DXPB                                                     NON 228
      DO 330 I=1,NPRFLS                                                 NON 229
      BLBK(I+13)=BLINT(I)                                               NON 230
      BLBK(I+15)=XSN(I)                                                 NON 231
      BLBK(I+17)=DELBL(I)                                               NON 232
330   BLBK(I+19)=DDELBL(I)                                              NON 233
C     SWITCH TO DOWNSTREAM REGION                                       NON 234
340   CALL THROAT                                                       NON 235
      IF (ERR) RETURN                                                   NON 236
      GO TO 450                                                         NON 237
350   IF (DLOGR) 450,450,360                                            NON 238
360   WRITE (6,1100)                                                    NON 239
370   IF ((AFNX-DATEST)/DATEST.LT.0.05) GO TO 380                       NON 240
C     ERROR EXIT NO. 3 ***************************************************NON 241
C     POSITIVE DLOGR FAR BEYOND SWITCH POINT                            NON 242
      IERR=3                                                            NON 243
      GO TO 1000                                                        NON 244
C     SWITCH TO DOWNSTREAM REGION WAS PREMATURE.                        NON 245
C     INCREASE DATEST AND RESTART UPSTREAM SOLUTION AT SWITCH POINT.    NON 246
380   DATEST=2.*(DATEST-1.)+1.                                          NON 247
      IUPD=1                                                            NON 248
      ARBB=ARBB+1.0                                                     NON 249
      IF (ARBB.LT.ARBA) GO TO 390                                       NON 250
C     ERROR EXIT NO. 4 ***************************************************NON 251
      IERR=4                                                            NON 252
      GO TO 1000                                                        NON 253
390   TSZ=TPRINT                                                        NON 254
      TPRINT=0.0                                                        NON 255
      CALL PRTA                                                         NON 256
      IF (ERR) RETURN                                                   NON 257
      ITAPE=NRCOUT-ITB(4)                                               NON 258
      IF (ITAPE.EQ.0) GO TO 410                                         NON 259
C     BACKSPACE BINARY OUTPUT TAPE TO ELIMINATE RECORDS OF INVALID      NON 260
C          STEPS BEYOND SWITCH POINT                                    NON 261
      DO 400 I=1,ITAPE                                                  NON 262
400   BACKSPACE ITPOUT                                                  NON 263
410   TPRINT=TSZ                                                        NON 264
C     RESET DATA FOR RESTART AT SWITCH POINT                            NON 265
      CX=TB(1)                                                          NON 266
      CT=TB(2)                                                          NON 267
      SC=TB(3)                                                          NON 268
      CTT=TB(4)                                                         NON 269
      DELTAX=TB(5)                                                      NON 270
      HDELX=TB(6)                                                       NON 271
      CTE=TB(7)                                                         NON 272
      SCD=TB(8)                                                         NON 273
      CHA=TB(9)                                                         NON 274
      INEQ=ITB(1)                                                       NON 275
```

```
      NNS=ITB(2)                                                        NON 276
      NNN=ITB(3)                                                        NON 277
      NRCOUT=ITB(4)                                                     NON 278
      DO 420 I=1,ISS                                                    NON 279
  420 GJ(I)=TB(I+10)                                                    NON 280
      RSA=BLBK(1)                                                       NON 281
      IF (ISW38.EQ.0) GO TO 220                                         NON 282
      DO 430 I=1,10                                                     NON 283
  430 BLNEAR(I)=BLBK(I+1)                                               NON 284
      XPB=BLBK(12)                                                      NON 285
      DXPB=BLBK(13)                                                     NON 286
      DO 440 I=1,NPRFLS                                                 NON 287
      BLINT(I)=BLBK(I+13)                                               NON 288
      XSN(I)=BLBK(I+15)                                                 NON 289
      DELBL(I)=BLBK(I+17)                                               NON 290
  440 DDELBL(I)=BLBK(I+19)                                              NON 291
      ISMD=30                                                           NON 292
      COUPLD=.FALSE.                                                    NON 293
      GO TO 220                                                         NON 294
  450 CONTINUE                                                          NON 295
      IF (INEQ) 610,460,610                                             NON 296
C     PERTURBATION SOLUTION                                             NON 297
  460 CALL PERT                                                         NON 298
      IF (ERR) RETURN                                                   NON 299
      IF (IUPD.EQ.0.AND.DELT.EQ.DELT1) DELT=DELTSV                      NON 300
C     DETERMINE MAXIMUM AND MINIMUM PERTURBATIONS IN CHI(I)             NON 301
      DCHMAX=0.                                                         NON 302
      DCHMIN=1.E30                                                      NON 303
      DO 480 I=1,ISR                                                    NON 304
      ADCH=ABS(SDCHI(I))                                                NON 305
      IF (ADCH-DCHMAX) 480,480,470                                      NON 306
  470 DCHMAX=ADCH                                                       NON 307
      IMAX=I                                                            NON 308
  480 DCHMIN=AMIN1(DCHMIN,ADCH)                                         NON 309
      WRITE (6,1110) DCHMIN,DCHMAX,IMAX                                 NON 310
      IF (DCHMAX.EQ.0.) GO TO 580                                       NON 311
      RATIO=DCHMIN/DCHMAX                                               NON 312
      IF (RATIO.GE.DCHRAT) GO TO 510                                    NON 313
C     EXCESSIVELY SMALL RATIO OF MINIMUM AND MAXIMUM PERTURBATIONS      NON 314
C          IN CHI(I)                                                    NON 315
      DO 500 J=1,ISS                                                    NON 316
      IF (BETA(IMAX,J)*PICHI(IMAX).GE.0.) GO TO 500                     NON 317
      IF (GJ(J).GT.GAMIN) GO TO 500                                     NON 318
      IF (IRA.LE.50) GO TO 490                                          NON 319
      WRITE (6,1010)                                                    NON 320
      GO TO 530                                                         NON 321
C     ARTIFICIAL INCREASE OF RATE CONSTANT TO PREVENT PREMATURE START   NON 322
C          OF INTEGRATION                                               NON 323
  490 RATIO2=1.1*DCHRAT/RATIO                                           NON 324
      CAI(IMAX)=CAI(IMAX)+ALOG(RATIO2)                                  NON 325
      WRITE (6,1020) RATIO,HP(J),IMAX,GJ(J),IMAX,RATIO2                 NON 326
      IRA=IRA+1                                                         NON 327
      GO TO 220                                                         NON 328
  500 CONTINUE                                                          NON 329
C     TESTS FOR SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION        NON 330
```

```
510   IF (DCHMAX.LT.CCHI) GO TO 580                                          NON 331
      IF (DCHMAX.LE.PCTEST*CCHI) GO TO 540                                   NON 332
C     MAXIMUM SDCHI(I) IS TOO LARGE -- BACKSTEP THE PERTURBATION             NON 333
C          SOLUTION TO FIND THE CORRECT SWITCHING POINT                      NON 334
      DELT=DELT/2.0                                                          NON 335
      ISUBD=ISUBD+1                                                          NON 336
      IF (ISUBD.GT.23) GO TO 520                                             NON 337
      CT=CT+DELT                                                             NON 338
      CTE=CT                                                                 NON 339
      NAMELIST /NEQDMP/    INEQ,IUPD,DLOGA,AR,CX,CT,DELT,AFNTS,              NON 340
     1     UP,SDCHI,DDELBL,PCT,PRHO,GJ,PERTGJ                               NON 341
      IF (ISWSB.NE.0) WRITE (6,NEQDMP)                                       NON 342
      GO TO 600                                                              NON 343
520   WRITE (6,1120) ISUBD                                                   NON 344
530   WRITE (6,NEQDMP)                                                       NON 345
C     ERROR EXIT NO. 5 ***************************************************NON 346
      IERR=5                                                                 NON 347
      GO TO 1000                                                             NON 348
C     SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION                       NON 349
540   INEQ=1                                                                 NON 350
      DXM=0.01*DCHMIN/DCHLL                                                  NON 351
      IF (DELTAX.LE.DXM) GO TO 550                                           NON 352
      DELTAX=DXM                                                             NON 353
      HDELX=0.5*DELTAX                                                       NON 354
      TB(5)=DELTAX                                                           NON 355
      TB(6)=HDELX                                                            NON 356
      DXOLD=DELTAX                                                           NON 357
550   CT=CT+PCT                                                              NON 358
      CTE=CT                                                                 NON 359
      RHO=RHO+PRHO                                                           NON 360
      DO 560 I=1,ISR                                                         NON 361
      IF (CAI(I).EQ.CAISV(I)) GO TO 560                                      NON 362
      CAI(I)=CAISV(I)                                                        NON 363
      WRITE (6,1030) I                                                       NON 364
560   CONTINUE                                                               NON 365
      DO 570 J=1,ISS                                                         NON 366
570   GJ(J)=GJ(J)+PERTGJ(J)                                                  NON 367
      CALL DERIVS                                                            NON 368
      IF (ERR) RETURN                                                        NON 369
      IF (.NOT.FAILED) GO TO 610                                             NON 370
C     ERROR EXIT NO. 6 ***************************************************NON 371
      IERR=6                                                                 NON 372
      GO TO 1000                                                             NON 373
C     CONTINUE PERTURBATION SOLUTION                                         NON 374
580   IF (CX.LT.XMODEL) GO TO 590                                            NON 375
      MODLPT=.TRUE.                                                          NON 376
      CALL NEXTMP (ITS,XMOD1,XMODEL)                                         NON 377
C     PRINT POINT IN PERTURBATION SOLUTION                                   NON 378
590   CALL PRTA                                                              NON 379
      IF (ERR) RETURN                                                        NON 380
      IF (CX.GE.CXMAX) GO TO 660                                             NON 381
      IF (CT.LE.TSTOP) GO TO 660                                             NON 382
      ISUBD=0                                                                NON 383
      CT=CT-DELT                                                             NON 384
      CTE=CT                                                                 NON 385
```

```
60    CALL NEWRAP                                                        NON 386
      IF (ERR) RETURN                                                    NON 387
      AFNTS=SM/FLUX                                                      NON 388
      CALL AXFIT                                                         NON 389
      IF (ERR) RETURN                                                    NON 390
      GO TO 220                                                          NON 391
C     NUMERICAL INTEGRATION                                              NON 392
510   IF (CT-1.0) 630,630,620                                            NON 393
520   WRITE (6,1130)                                                     NON 394
C     ERROR EXIT NO. 7 ***************************************************NON 395
      IERR=7                                                             NON 396
      GO TO 1000                                                         NON 397
630   IF (CX.GE.CXMAX) GO TO 660                                         NON 398
      IF (CT.LE.TSTOP) GO TO 660                                         NON 399
      NNN=NNN+1                                                          NON 400
C     PRINT POINT IN SOLUTION BY INTEGRATION                             NON 401
      CALL PRTA                                                          NON 402
      IF (ERR) RETURN                                                    NON 403
      IF (NNN-NQS) 670,640,640                                           NON 404
640   DELTAX=SC*DELTAX                                                   NON 405
      HDELX=DELTAX/2.0                                                   NON 406
      NNS=NNS+1                                                          NON 407
      NNN=0                                                              NON 408
      IF (NNS-NQS) 670,650,650                                           NON 409
650   SC=SC+0.1                                                          NON 410
      SCO=SC                                                             NON 411
      NNS=0                                                              NON 412
      GO TO 670                                                          NON 413
C     FINAL POINT                                                        NON 414
660   TPRINT=0.0                                                         NON 415
      CALL PRTA                                                          NON 416
      AXISYM=.TRUE.                                                      NON 417
      CALL MODEL                                                         NON 418
      WRITE (6,1160) ISTEPS                                              NON 419
C                                                                        NON 420
C     MAIN RETURN STATEMENT.....................................         NON 421
C                                                                        NON 422
      RETURN                                                             NON 423
C     SAVE DATA AT BEGINNING OF INTEGRATION STEP FOR POSSIBLE RESTART    NON 424
670   DO 680 I=1,ISS                                                     NON 425
      GJB(I)=GJ(I)                                                      NON 426
680   CONTINUE                                                           NON 427
      CXB=CX                                                             NON 428
      CTB=CT                                                             NON 429
      CTSB=CTS                                                           NON 430
      CARB=ARNX                                                          NON 431
      RHOB=RHO                                                           NON 432
      CRS=SEN                                                            NON 433
      CHR=CHA                                                            NON 434
      DELBLB(1)=DELBL(1)                                                 NON 435
      IF (NPR.LS.EQ.2) DELBLB(2)=DELBL(2)                                NON 436
690   IF (CX+DELTAX.LT.XMODEL) GO TO 700                                 NON 437
      MODLPT=.TRUE.                                                      NON 438
      DXSAVE=DELTAX                                                      NON 439
      HDXSAV=HDELX                                                       NON 440
```

```
      DELTAX=XMODEL-CX                                                  NON 441
      HDELX=0.5*DELTAX                                                  NON 442
700   IF (CX+DELTAX.LT.CXMAX) GO TO 710                                 NON 443
      DELTAX=CXMAX-CX                                                   NON 444
      HDELX=0.5*DELTAX                                                  NON 445
C     INTEGRATION STEP                                                  NON 446
710   CALL RNKT                                                         NON 447
      IF (ERR) RETURN                                                   NON 448
      IF (FAILED) GO TO 740                                             NON 449
      IF (.NOT.MODLPT) GO TO 720                                        NON 450
      DELTAX=DXSAVE                                                     NON 451
      HDELX=HDXSAV                                                      NON 452
720   CONTINUE                                                          NON 453
C     TESTS FOR VALID INTEGRATION STEP                                  NON 454
      IFAIL=1                                                           NON 455
      IF (ABS(SDT/CT).GT.TTEST) GO TO 740                               NON 456
      IF (NT.EQ.1) GO TO 730                                            NON 457
      IFAIL=2                                                           NON 458
      IF (ABS(SDTE/CTE).GT.TETEST) GO TO 740                            NON 459
      IFAIL=3                                                           NON 460
      IF (ABS(SDCHA/CHA).GT.HTEST) GO TO 740                            NON 461
730   IFAIL=4                                                           NON 462
      IF (ALOGA*CX.GE.-0.01) GO TO 760                                  NON 463
C     INTEGRATION STEP FAILED. REDUCE STEP SIZE AND RESTART AT          NON 464
C          BEGINNING OF STEP                                            NON 465
740   DELTAX=DELTAX/SCD                                                 NON 466
      HDELX=DELTAX/2.0                                                  NON 467
      MODLPT=.FALSE.                                                    NON 468
      ICOUNT=ICOUNT+1                                                   NON 469
      IF (ICOUNT.GT.30.OR.DELTAX.LE.1.E-10) GO TO 990                   NON 470
      IF (ICOUNT) =IFAIL                                                NON 471
      DO 750 I=1,ISS                                                    NON 472
750   GJ(I)=GJB(I)                                                      NON 473
      CT=CTB                                                            NON 474
      CTE=CTEB                                                          NON 475
      CX=CXB                                                            NON 476
      AFNX=CARB                                                         NON 477
      RHO=RHOB                                                          NON 478
      SEN=CRS                                                           NON 479
      CHA=CHB                                                           NON 480
      DELBL(1)=DELBLB(1)                                                NON 481
      IF (NFRFLS.EQ.2) DELBL(2)=DELBLB(2)                               NON 482
      NNN=0                                                             NON 483
      NNS=0                                                             NON 484
      SC=SC-0.1                                                         NON 485
      SC=AMAX1(SC,1.1)                                                  NON 486
      SCD=1.1*SCD                                                       NON 487
      CALL DERIVS                                                       NON 488
      IF (ERR) RETURN                                                   NON 489
      IF (.NOT.FAILED) GO TO 690                                        NON 490
C     ERROR EXIT NO. 8 ***************************************************NON 491
      IERR=8                                                            NON 492
      GO TO 1000                                                        NON 493
C     FURTHER VALIDITY CHECKS ON STEP                                   NON 494
760   DO 790 I=1,ISS                                                    NON 495
```

```
      IFAIL=I+10                                                        NON 496
      GJ(I)=GJB(I)+SDGJ(I)                                              NON 497
      IF (GJ(I).LT.0.) GO TO 770                                        NON 498
      IF (ABS(SDGJ(I)/GJB(I)).LE.GTEST) GO TO 790                       NON 499
  770 IF (GJB(I).GE.GAMIN) GO TO 740                                    NON 500
      IF (SDGJ(I).GE.0.) GO TO 740                                      NON 501
C     SUPPRESS REACTIONS INVOLVING STEP-SIZE CONTROLLING SPECIES        NON 502
C           WITH CONCENTRATION BELOW GAMIN AND FALLING                  NON 503
      NSUPR=0                                                           NON 504
      DO 780 K=1,ISR                                                    NON 505
      IF (BETA(K,I).EQ.0.) GO TO 780                                    NON 506
      NOREAC(K)=.TRUE.                                                  NON 507
      NSUPR=NSUPR+1                                                     NON 508
      ISUPR(NSUPR)=K                                                    NON 509
  780 CONTINUE                                                          NON 510
      WRITE (6,1050) HP(I),GJB(I),(ISUPR(K),K=1,NSUPR)                  NON 511
      GO TO 740                                                         NON 512
C     VALID STEP. APPLY INCREMENTS TO DEPENDENT VARIABLES.              NON 513
  790 CONTINUE                                                          NON 514
      CT=CTB+SDT                                                        NON 515
      IF (NT.EQ.1) GO TO 800                                            NON 516
      CTE=CTEB+SDTE                                                     NON 517
      CHA=CHB+SDCHA                                                     NON 518
C     RESTORE CONSERVATION OF CHEMICAL ELEMENTS                         NON 519
  800 DO 820 J=1,ISC                                                    NON 520
      CJ=0.                                                             NON 521
      DO 810 I=1,ISS                                                    NON 522
  810 CJ=CJ+GJ(I)*LPIJ(I,J)                                             NON 523
  820 DCI(J)=CI(J)-CJ                                                   NON 524
      DO 830 I=1,ISS                                                    NON 525
      GSQ(I)=GJ(I)**2                                                   NON 526
      IF (GSQ(I).GT.0..OR.I.GT.ISC) GO TO 830                           NON 527
      WRITE (6,1060) I                                                  NON 528
C     ERROR EXIT NO. 9 ***************************************************NON 529
      IERR=9                                                            NON 530
      GO TO 1000                                                        NON 531
  830 CONTINUE                                                          NON 532
      ISMCP1=ISMC+1                                                     NON 533
      DO 850 K=1,ISC                                                    NON 534
      DCA(K)=0.                                                         NON 535
      DO 840 J=1,ISC                                                    NON 536
  840 DCA(K)=DCA(K)+DCI(J)*AIN(J,K)                                     NON 537
      DCA(K)=DCA(K)/GSQ(K)                                              NON 538
  850 CONTINUE                                                          NON 539
      DO 890 I=1,ISMC                                                   NON 540
      II=I+ISC                                                          NON 541
      AA(I,ISMCP1)=0.                                                   NON 542
      DO 860 K=1,ISC                                                    NON 543
  860 AA(I,ISMCP1)=AA(I,ISMCP1)+CDIJ(I,K)*DCA(K)                        NON 544
      AA(I,ISMCP1)=AA(I,ISMCP1)*GSQ(II)                                 NON 545
      DO 880 L=1,ISMC                                                   NON 546
      AA(I,L)=0.                                                        NON 547
      DO 870 K=1,ISC                                                    NON 548
  870 AA(I,L)=AA(I,L)+CDIJ(I,K)*CDIJ(L,K)/GSQ(K)                        NON 549
      AA(I,L)=AA(I,L)*GSQ(II)                                           NON 550
```

```
      IF (I.EQ.L) AA(I,L)=AA(I,L)+ONE                                    NON 551
880   CONTINUE                                                           NON 552
890   CONTINUE                                                           NON 553
      CALL DSMSOL (AA,ISMC,22)                                           NON 554
      IF (ERR) RETURN                                                    NON 555
      DO 900 I=1,ISMC                                                    NON 556
      II=I+ISC                                                           NON 557
900   GJ(II)=GJ(II)+AA(I,ISMCP1)                                         NON 558
      DO 930 K=1,ISC                                                     NON 559
      GJ(K)=0.                                                           NON 560
      DO 910 J=1,ISC                                                     NON 561
910   GJ(K)=GJ(K)+CI(J)*AIN(J,K)                                         NON 562
      DO 920 L=1,ISMC                                                    NON 563
      I=L+ISC                                                            NON 564
920   GJ(K)=GJ(K)-GJ(I)*CDIJ(L,K)                                        NON 565
930   CONTINUE                                                           NON 566
      DO 940 I=1,ISS                                                     NON 567
      IFAIL=I+30                                                         NON 568
      IF (GJ(I)) 740,940,940                                             NON 569
940   CONTINUE                                                           NON 570
C     COMPUTE SUPPLEMENTARY FLOW VARIABLES AND DERIVATIVES AT END OF     NON 571
C           STEP (AND BEGINNING OF NEXT STEP)                            NON 572
      CALL DERIVS                                                        NON 573
      IF (ERR) RETURN                                                    NON 574
      IFAIL=60                                                           NON 575
      IF (FAILED) GO TO 740                                              NON 576
      IF (NT.EQ.1) GO TO 960                                             NON 577
C     FINAL VALIDITY CHECK IN ELECTRONIC NONEQUILIBRIUM MODEL --         NON 578
C           TEST ON CHANGE IN ENERGY TRANSFER TO ELECTRONS               NON 579
      IF (QEOLD.EQ.0.) GO TO 950                                         NON 580
      IFAIL=5                                                            NON 581
      DQEM=AMAX1(QTEST*ABS(QEOLD),DQMAX)                                 NON 582
      IF (ABS(QDPE-QEOLD).GT.DQEM) GO TO 740                             NON 583
      DQMAX=AMAX1(DQMAX,DQEM)                                            NON 584
950   QEOLD=QDPE                                                         NON 585
C     FULLY SUCCESSFUL STEP. SET UP FOR NEXT STEP.                       NON 586
960   SCD=SC                                                             NON 587
      TP=CT*CTAP                                                         NON 588
      TEP=CTE*CTAP                                                       NON 589
      IF (ISS.GE.0) GO TO 970                                            NON 590
      WRITE (6,1140) CX,DELTAX,TP,TEP,CHA,QDPE,ICOUNT                    NON 591
      IF (ICOUNT.EQ.0) GO TO 980                                         NON 592
      WRITE (6,1040) (IF(K),K=1,ICOUNT)                                  NON 593
970   ICOUNT=0                                                           NON 594
980   ISTEPS=ISTEPS+1                                                    NON 595
      DELTAX=AMAX1(DELTAX,0.7*DXOLD)                                     NON 596
      DXOLD=DELTAX                                                       NON 597
      GO TO 210                                                          NON 598
990   WRITE (6,1150)                                                     NON 599
C     ERROR EXIT NO. 10**************************************************NON 600
      IERR=10                                                            NON 601
      IF (IUPD.EQ.0) GO TO 370                                           NON 602
1000  WRITE (6,1070) IERR                                                NON 603
      CALL DUMP (RNAME)                                                  NON 604
      RETURN                                                             NON 605
```

```
C                                                                       NON 606
C                                                                       NON 607
C                                                                       NON 608
1010  FORMAT (71H0MORE THAN 50 RATE-CONSTANT INCREASES REQUIRED IN PERTUNON 609
     1RBATION SOLUTION)                                                 NON 610
1020  FORMAT (20H0=====)CHMIN/DCHMAX=,1PE10.3,41H IS TOO SMALL. CONCENTRNON 611
     1ATION OF REACTANT ,A4,16H IN REACTION NO.,I3,3H IS,E10.3,8HMOLE/GMNON 612
     2./36H =====RATE CONSTANT FOR REACTION NO.,I3,25H INCREASED BY A FANON 613
     3CTOR OF,E10.3,27H FOR PERTURBATION SOLUTION./)                    NON 614
1030  FORMAT (36H =====RATE CONSTANT FOR REACTION NO.,I3,27H RESTORED TONON 615
     1 CORRECT VALUE,)                                                  NON 616
1040  FORMAT (24H CAUSES OF STEP FAILURES,20I4)                         NON 617
1050  FORMAT (28H0..........CONCENTRATION OF ,A4,10H FROZEN AT,1PE10.3,9NON 618
     1H MOLE/GM,/11X,20HSUPPRESSED REACTIONS,3413/30X,30I3)             NON 619
1060  FORMAT (4H0GJ(,I2,16H)**2 UNDERFLOWED)                            NON 620
1070  FORMAT (15H0ERROR EXIT NO.,I3,11H FROM NONEQ)                     NON 621
1080  FORMAT (24H1NONEQUILIBRIUM SOLUTION//)                            NON 622
1090  FORMAT (33H1BETA MATRIX OF INSUFFICIENT RANK)                     NON 623
1100  FORMAT (18H DLOGR IS POSITIVE)                                    NON 624
1110  FORMAT (8H DCHMIN=,1PE10.3,10X,7HDCHMAX=,E10.3,10X,5HIMAX=,I2)    NON 625
1120  FORMAT (55H0BACKSTEPPING OF PERTURBATION SOLUTION TERMINATED AFTERNON 626
     1,I3,31H STEPS. DIAGNOSTIC DATA FOLLOW.)                           NON 627
1130  FORMAT (41H1TEMPERATURE GREATER THAN RESERVOIR VALUE)             NON 628
1140  FORMAT (3H0X=,1PE12.5,5X,7HDELTAX=,E10.3,5X,2HT=,0PF8.2,5X,4HTEP=,NON 629
     1F8.2,5X,4HCHA=,1PE15.7,5X,5HQDPE=,E10.3,5X,7HICOUNT=,I2)          NON 630
1150  FORMAT (60H0NONEQUILIBRIUM INTEGRATION UNABLE TO TAKE A CONVERGENTNON 631
     1 STEP)                                                            NON 632
1160  FORMAT (33H0NONEQUILIBRIUM SOLUTION REQUIRED,I5,18H INTEGRATION STNON 633
     1EPS)                                                              NON 634
      END                                                               NON 635
```

```
      SUBROUTINE NRMAX                                                  NRM    1
      LOGICAL ERR                                                       NRM    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   NRM    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              NRM    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             NRM    5
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  NRM    6
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CPA,   CRP,  NRM    7
     2  CRRB,    RS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,   NRM    8
     3   CTP,  C  L,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELTI,   NRM    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT   NRM   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,  RHO,  RHOB,   NRM   11
     6RHOBAR,  RHOC,  RHOP,  RHPL, RHTH,ROBARA,ROBARP,  SCPG,   SDT,   NRM   12
     7   SEN,  SHPG,    SC,    SL, SL64,    SM,   SU,   SU2,  SUMG,   NRM   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,  ZPA                  NRM   14
      COMMON          BE(  ),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),NRM   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),NRM   16
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),NRM   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICH1(64),    QM(20),NRM   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),NRM   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),NRM   20
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),   XNUI(64) NRM   21
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          NRM   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       NRM   23
      COMMON      IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,  NRM   24
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,   NRM   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    NRM   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,   NRM   27
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        NRM   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) NRM   29
      COMMON ACOM,ELMENT,HP                                             NRM   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBNRM   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                NRM   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              NRM   33
     1     (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),        NRM   34
     2     (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),             NRM   35
     3     (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                        NRM   36
      COMMON /ERROR/ ERR                                                NRM   37
      M1=ISC+1                                                          NRM   38
      DELT=DELT1                                                        NRM   39
      PRES=1.0                                                          NRM   40
      T1=1.0                                                            NRM   41
      CT=T1                                                             NRM   42
      DO 10 J=1,ISS                                                     NRM   43
   10 GJ(J)=GJA(J)                                                      NRM   44
      F1=0.                                                             NRM   45
      CT=CT-DELT                                                        NRM   46
   20 T2=CT                                                             NRM   47
      CALL NEWRAP                                                       NRM   48
      IF (ERR) RETURN                                                   NRM   49
      F2=FLUX                                                           NRM   50
   30 CT=CT-DELT                                                        NRM   51
      T3=CT                                                             NRM   52
      CALL NEWRAP                                                       NRM   53
      IF (ERR) RETURN                                                   NRM   54
      F3=FLUX                                                           NRM   55
```

```
      IF (F3-F2) 40,60,80                                          NRM   56
40    IF (DELT-TESTB) 90,50,50                                     NRM   57
50    DELT=DELT/2.                                                 NRM   58
      CT=T1-DELT                                                   NRM   59
      GO TO 20                                                     NRM   60
60    IF (DELT-TESTB) 90,70,70                                     NRM   61
70    T1=T2                                                        NRM   62
      F1=F2                                                        NRM   63
      DELT=DELT/2.                                                 NRM   64
      CT=T1-DELT                                                   NRM   65
      GO TO 20                                                     NRM   66
80    T1=T2                                                        NRM   67
      T2=T3                                                        NRM   68
      F1=F2                                                        NRM   69
      F2=F3                                                        NRM   70
      GO TO 30                                                     NRM   71
90    Z=F1-F2+F3-F2                                                NRM   72
      IF (ABS(Z/F2).GT..00001) GO TO 100                           NRM   73
      CT=T2                                                        NRM   74
      GO TO 110                                                    NRM   75
100   CT=T2+DELT*(F3-F1)/(2.*Z)                                    NRM   76
110   CONTINUE                                                     NRM   77
      CALL NEWRAP                                                  NRM   78
      IF (ERR) RETURN                                              NRM   79
      SM=FLUX                                                      NRM   80
      RHTH=RHO                                                     NRM   81
      CTMAX=CT                                                     NRM   82
      PRESTH=PRES                                                  NRM   83
      DO 120 I=1,ISS                                               NRM   84
120   CAPXTH(I)=CAPX(I)                                            NRM   85
      RETURN                                                       NRM   86
      END                                                          NRM   87-
```

```
      SUBROUTINE OUT                                                    OUT    1
      WRITE (6,10)                                                      OUT    2
      WRITE (6,20)                                                      OUT    3
      WRITE (6,30)                                                      OUT    4
      WRITE (6,40)                                                      OUT    5
      WRITE (6,50)                                                      OUT    6
      WRITE (6,60)                                                      OUT    7
      WRITE (6,70)                                                      OUT    8
      WRITE (6,80)                                                      OUT    9
      WRITE (6,90)                                                      OUT   10
      WRITE (6,100)                                                     OUT   11
      WRITE (6,110)                                                     OUT   12
      RETURN                                                            OUT   13
C                                                                       OUT   14
CCC                                                                     OUT   15
C                                                                       OUT   16
10    FORMAT (23H1DEFINITIONS OF SYMBOLS//)                             OUT   17
20    FORMAT (31H ARAT     = GEOMETRIC AREA RATIO/31H ARATEF  = EFFECTIVEOUT   18
     1 AREA RATIO/96H DELREF  = EQUILIBRIUM SHOCK STANDOFF DISTANCE ON FOUT   19
     2LAT-FACED MODEL, DIVIDED BY MODEL NOSE RADIUS/99H DELREH  = EQUILIOUT   20
     3BRIUM SHOCK STANDOFF DISTANCE ON HEMISPHERICAL MODEL, DIVIDED BY MOUT   21
     4ODEL NOSE RADIUS/91H DELRFF  = FROZEN SHOCK STANDOFF DISTANCE ON FOUT   22
     5LAT-FACED MODEL, DIVIDED BY MODEL NOSE RADIUS/94H DELRFH  = FROZENOUT   23
     6 SHOCK STANDOFF DISTANCE ON HEMISPHERICAL MODEL, DIVIDED BY MODEL OUT   24
     7NOSE RADIUS/73H DELSTR  = DISPLACEMENT THICKNESS OF BOUNDARY LAYEROUT   25
     8 ON NOZZLE WALL (INCH))                                           OUT   26
30    FORMAT (64H DELSTW  = BOUNDARY LAYER DISPLACEMENT THICKNESS ON WEDOUT   27
     1GE (INCH)/35H DIAM    = NOZZLE DIAMETER (INCHES)/56H EPSLE   = DENOUT   28
     2SITY RATIO ACROSS EQUILIBRIUM NORMAL SHOCK/51H EPSLF   = DENSITY ROUT   29
     3ATIO ACROSS FROZEN NORMAL SHOCK/52H GAMA   = FROZEN SPECIFIC-HEATOUT   30
     4 RATIO IN FREE STREAM/43H H        = ENTHALPY IN FREE STREAM (BTU/LOUT   31
     59)/58H HEIGHT  = SECOND TRANSVERSE DIMENSION OF CHANNEL (INCHES)/3OUT   32
     67H HR       = RECOVERY ENTHALPY (BTU/LB)/74H HRATE   = RATIO OF DISOUT   33
     7SOCIATION ENTHALPY TO (HS-HW) FOR EQUILIBRIUM SHOCK)              OUT   34
40    FORMAT (69H HRATF   = RATIO OF DISSOCIATION ENTHALPY TO (HS-HW) FOOUT   35
     1R FROZEN SHOCK/39H HS       = STAGNATION ENTHALPY (BTU/LB)/84H K2PFOUT   36
     2E   = SHOCK-BOUNDARY LAYER INTERACTION PARAMETER PER FOOT FOR EQUIOUT   37
     3LIBRIUM SHOCK/79H K2PFF   = SHOCK-BOUNDARY LAYER INTERACTION PARAMOUT   38
     4ETER PER FOOT FOR FROZEN SHOCK/83H LEE      = ATOM-MOLECULE LEWIS NOUT   39
     5UMBER AT STAGNATION CONDITION FOR EQUILIBRIUM SHOCK/78H LEF      = OUT   40
     6ATOM-MOLECULE LEWIS NUMBER AT STAGNATION CONDITION FOR FROZEN SHOCOUT   41
     7K/37H M        = MACH NUMBER IN FREE STREAM/47H MU       = VISCOSITYOUT   42
     8 IN FREE STREAM (LB/FT-SEC)/78H MUT2E   = VISCOSITY AT STAGNATION OUT   43
     9CONDITION FOR EQUILIBRIUM SHOCK (LB/FT-SEC)/73H MUT2F   = VISCOSITOUT   44
     $Y AT STAGNATION CONDITION FOR FROZEN SHOCK (LB/FT-SEC)/52H MW      OUT   45
     $ = MOLECULAR WEIGHT IN FREE STREAM (GM/MOLE))                     OUT   46
50    FORMAT (83H MWT2E   = MOLECULAR WEIGHT AT STAGNATION CONDITION FOROUT   47
     1 EQUILIBRIUM SHOCK (GM/MOLE)/78H MWT2F   = MOLECULAR WEIGHT AT STAOUT   48
     2GNATION CONDITION FOR FROZEN SHOCK (GM/MOLE)/53H N        = BOUNDAROUT   49
     3Y-LAYER PRESSURE-GRADIENT PARAMETER/40H P        = PRESSURE IN FREEOUT   50
     4 STREAM (ATM)/71H PRE      = PRANDTL NUMBER AT STAGNATION CONDITIONOUT   51
     5 FOR EQUILIBRIUM SHOCK/66H PRF      = PRANDTL NUMBER AT STAGNATION OUT   52
     6CONDITION FOR FROZEN SHOCK/68H PRREF   = PRANDTL NUMBER AT REFERENOUT   53
     7CE TEMPERATURE IN BOUNDARY LAYER/58H PT2E    = STAGNATION PRESSUREOUT   54
     8 FOR EQUILIBRIUM SHOCK (ATM)/53H PT2F    = STAGNATION PRESSURE FOROUT   55
```

```
     9 FROZEN SHOCK (ATM)/50H PWW      = SURFACE PRESSURE ON A WEDGE MODEOUT  56
     $L (ATM))                                                          OUT  57
60    FORMAT (130H QEFE     = STAGNATION-POINT HEAT FLUX TIMES SQRT(R) FOOUT  58
     1R EQUILIBRIUM SHOCK ON A FLAT-FACED MODEL WITH AN EQUILIBRIUM BOUNDOUT  59
     2DARY LAYER/20X,40H(BTU/SQ FT-SEC)FT**0.5         FAY-RIDDELL/124H QEOUT  60
     3FF     = STAGNATION-POINT HEAT FLUX TIMES SQRT(R) FOR EQUILIBRIUM SOUT  61
     4HOCK ON A FLAT-FACED MODEL WITH A FROZEN BOUNDARY LAYER/20X,40H(BTOUT  62
     5U/SQ FT-SEC)FT**0.5         FAY-RIDDELL/127H QEHE     = STAGNATION-POOUT  63
     6INT HEAT FLUX TIMES SQRT(R) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICOUT  64
     7AL MODEL WITH AN EQUILIBRIUM BOUNDARY/20X,41HLAYER (BTU/SQ FT-SEC)OUT  65
     8FT**0.5  FAY-RIDDELL/127H QEHF     = STAGNATION-POINT HEAT FLUX TIMOUT  66
     9ES SQRT(R) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICAL MODEL WITH A FOUT  67
     $ROZEN BOUNDARY LAYER/20X,41H(BTU/SQ FT-SEC)FT**0.5          FAY-RIDDOUT  68
     $ELL)                                                              OUT  69
70    FORMAT (62H QELET     = NET POWER ADDED TO THE ELECTRON GAS (CAL/CU CUT  70
     1CM-SEC)/120H QESRI    = STAGNATION-POINT HEAT FLUX TIMES SQRT(R) FOOUT  71
     2R EQUILIBRIUM SHOCK ON A MODEL (BTU/SQ FT-SEC)FT**0.5  SRI FORMULAOUT  72
     3/125H QFFE     = STAGNATION-POINT HEAT FLUX TIMES SQRT(R) FOR FROZEOUT  73
     4N SHOCK ON A FLAT-FACED MODEL WITH AN EQUILIBRIUM BOUNDARY LAYER/20OUT  74
     5X,41H(BTU/SQ FT-SEC)FT**0.5          FAY-RIDDELL/119H QFFF     = STAOUT  75
     6GNATION-POINT HEAT FLUX TIMES SQRT(R) FOR FROZEN SHOCK ON A FLAT-FOUT  76
     7ACED MODEL WITH A FROZEN BOUNDARY LAYER/20X,41H(BTU/SQ FT-SEC)FT**OUT  77
     80.5          FAY-RIDDELL)                                         OUT  78
80    FORMAT (128H QFHE     = STAGNATION-POINT HEAT FLUX TIMES SQRT(R) FOOUT  79
     1R FROZEN SHOCK ON A HEMISPHERICAL MODEL WITH AN EQUILIBRIUM BOUNDAOUT  80
     2RY LAYER/20X,40H(BTU/SQ FT-SEC)FT**0.5         FAY-RIDDELL/122H QFHFOUT  81
     3     = STAGNATION-POINT HEAT FLUX TIMES SQRT(R) FOR FROZEN SHOCK ONOUT  82
     4 A HEMISPHERICAL MODEL WITH A FROZEN BOUNDARY LAYER/20X,40H(BTU/SQOUT  83
     5 FT-SEC)FT**0.5         FAY-RIDDELL/79H QFP      = HEAT FLUX TO A FLAOUT  84
     6T PLATE 1 FT FROM THE LEADING EDGE (BTU/SQ FT-SEC))               OUT  85
90    FORMAT (117H QFSRI    = STAGNATION-POINT HEAT FLUX TIMES SQRT(R) FOOUT  86
     1R FROZEN SHOCK ON MODEL (BTU/SQ FT-SEC)FT**0.5        SRI FORMULA/41OUT  87
     2H QRAD     = RADIATED POWER (CAL/CU CM-SEC)/51H QW       = HEAT FLUXOUT  88
     3 TO NOZZLE WALL (BTU/SQ FT-SEC)/64H QWW      = HEAT FLUX TO SURFACEOUT  89
     4 OF A WEDGE MODEL (BTU/SQ FT-SEC)/44H R        = DENSITY IN FREE STOUT  90
     5REAM (LB/CU FT)/50H REPF     = REYNOLDS NUMBER PER FOOT IN FREE STROUT  91
     6EAM/54H RETH     = REYNOLDS NUMBER BASED ON MOMENTUM THICKNESS/95H OUT  92
     7RETHTR   = CRITICAL REYNOLDS NUMBER (BASED ON MOMENTUM THICKNESS) FOUT  93
     8OR BOUNDARY LAYER TRANSITION/75H RT2E     = DENSITY AT STAGNATION COUT  94
     9ONDITION FOR EQUILIBRIUM SHOCK (LB/CU FT)/48H S        = ENTROPY INOUT  95
     $ FREE STREAM (BTU/LB-DEG R)/58H SIGMA    = ELECTRICAL CONDUCTIVITY OUT  96
     $IN FREE STREAM (MHO/CM))                                          OUT  97
100   FORMAT (89H SIGT2E   = ELECTRICAL CONDUCTIVITY AT STAGNATION CONDITOUT  98
     1ION FOR EQUILIBRIUM SHOCK (MHO/CM)/84H SIGT2F   = ELECTRICAL CONDUCOUT  99
     2TIVITY AT STAGNATION CONDITION FOR FROZEN SHOCK (MHO/CM)/49H STANTOUT 100
     3N   = STANTON NUMBER BASED ON TOTAL ENTHALPY/79H ST2E     = ENTROPY OUT 101
     4AT STAGNATION CONDITION FOR EQUILIBRIUM SHOCK (BTU/LB-DEG R)/74H SOUT 102
     5T2F     = ENTROPY AT STAGNATION CONDITION FOR FROZEN SHOCK (BTU/LB-OUT 103
     6DEG R)/45H T          = TEMPERATURE IN FREE STREAM (DEG K))       OUT 104
110   FORMAT (76H T2E       = TEMPERATURE AT STAGNATION CONDITION FOR EQUIOUT 105
     1LIBRIUM SHOCK (DEG K)/71H T2F       = TEMPERATURE AT STAGNATION CONDOUT 106
     2ITION FOR FROZEN SHOCK (DEG K)/50H TAUW     = SHEAR STRESS AT NOZZLOUT 107
     3E WALL (LBF/SQ FT)/39H TELEC    = ELECTRON TEMPERATURE (DEG K)/69H OUT 108
     4THETA    = MOMENTUM THICKNESS OF BOUNDARY LAYER ON NOZZLE WALL (INCOUT 109
     5H)/43H V         = VELOCITY OF FREE STREAM (FT/SEC)/57H WIDTH    = FIOUT 110
```

```
6RST TRANSVERSE DIMENSION OF CHANNEL (INCHES)/93H X          = AXIAL DOUT 111
7ISTANCE ALONG NOZZLE, MEASURED FROM THROAT AND POSITIVE DOWNSTREAMOUT 112
8 (INCHES)/67H XSN      = N AVERAGED OVER SEVERAL PRECEDING POINTS OOUT 113
9F THE SOLUTION/85H XW          = DISTANCE FROM LEADING EDGE OF WEDGE, OUT 114
$MEASURED ALONG WEDGE SURFACE (INCHES)/115H YS          = ORDINATE OF SOUT 115
$HOCK FROM LINE PARALLEL TO FREE STREAM FLOW, PASSING THROUGH LEADIOUT 116
$NG EDGE OF WEDGE (INCHES)/71H ZETA     = NONDIMENSIONAL STREAMWISE OUT 117
$COORDINATE IN BLUNT WEDGE ANALYSIS)                                OUT 118
 END                                                                OUT 119-
```

```
      SUBROUTINE OUT1                                                   OU1    1
C     OUTPUT ROUTINE                                                    OU1    2
      LOGICAL ERR,DATAPE,SUPGO,NOTRAN                                   OU1    3
      REAL ACOM(30),ELMENT(10),HP(20)                                   OU1    4
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              OU1    5
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             OU1    6
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO, OU1    7
     1       C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP, OU1    8
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX, OU1    9
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1, OU1   10
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT, OU1   11
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB, OU1   12
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT, OU1   13
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG, OU1   14
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                OU1   15
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),OU1   16
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),OU1   17
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),OU1   18
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),OU1   19
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),OU1   20
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),OU1   21
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),  XNUI(64) OU1   22
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          OU1   23
     1 XNUIJ(64,20),XNUIJP(64,20)                                       OU1   24
      COMMON      IC,     IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1, OU1   25
     1  ISMC,ISMCNR,    ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  OU1   26
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    OU1   27
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,  OU1   28
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        OU1   29
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) OU1   30
      COMMON ACOM,ELMENT,HP                                             OU1   31
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBOU1   32
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                OU1   33
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              OU1   34
     1     (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       OU1   35
     2     (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            OU1   36
     3     (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                       OU1   37
      COMMON /RDOUT/ FLOW,FACNAM,CHANAM,LIMOUT                          OU1   38
      COMMON /TEMPRY/ SAVEC(20)                                         OU1   39
      COMMON /BLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOT(2),      OU1   40
     1     TAUW(2)                                                      OU1   41
      COMMON /CONVRT/ CF(6)                                             OU1   42
      DIMENSION VARNAM (35),DLGDX(20)                                   OU1   43
      EQUIVALENCE (DLGDX(1),AA(1,1))                                    OU1   44
      DATA VARNAM /'X','T','H','P','R',                                 OU1   45
     1     'DIAM','V','M','S','GAMMA',                                  OU1   46
     2     'ARATEF','REPF','MW','MU','SIGMA',                           OU1   47
     3     'ARAT','DELSTR','THETA','QW','TAUW',                         OU1   48
     4     'HR','PRREF','STANTN','RETH','RETHTR',                       OU1   49
     5     'HEIGHT','DELSTR','THETA','QW','TAUW',                       OU1   50
     6     'HR','PRREF','STANTN','RETH','RETHTR'/                       OU1   51
      DIMENSION VARN2(2),FVO2(2,2)                                      OU1   52
      DATA VARN2 /'N','XSN'/                                            OU1   53
      DATA DNAM /'DIAM'/, WNAM /'WIDTH'/, ANAM /'ARAT'/                 OU1   54
      COMMON /OUTPUT/ FVOUT(35),GJMF(20)                                OU1   55
```

```
50      FAREA=2.*RTHCM(1)                                                       OU1  111
        GO TO 100                                                               OU1  112
60      IF (NPRFLS.EQ.1) GO TO 80                                               OU1  113
        DO 70 I=1,2                                                             OU1  114
70      DELSTP(I)=(1.-OMDST(I))*R0                                              OU1  115
        FAREA=4.*(RTHCM(1)-DELSTP(1))*(RTHCM(2)-DELSTP(2))                      OU1  116
        GO TO 100                                                               OU1  117
80      IF (JDIM.EQ.0) GO TO 90                                                 OU1  118
        FAREA=PZ*(OMDST(1)*RTHCM(1))**2                                         OU1  119
        GO TO 100                                                               OU1  120
90      FAREA=2.*OMDST(1)*RTHCM(1)                                              OU1  121
100     FAREA=FAREA/929.03                                                      OU1  122
        WRITE (6,570)                                                           OU1  123
        TFLOW=FAREA*CF(1)*CF(2)*SM                                              OU1  124
        IF (ISW2B.GT.0) GO TO 110                                               OU1  125
        WRITE (6,580) FLOW                                                      OU1  126
110     IF (IC.EQ.0) GO TO 120                                                  OU1  127
        DEL=SAVEC(1)*RHO*TB7/CM                                                 OU1  128
        GO TO 130                                                               OU1  129
120     DEL=0.                                                                  OU1  130
130     IF (NPRFLS.EQ.2) GO TO 140                                              OU1  131
        WRITE (6,590) FACNAM,DIAM(1)                                            OU1  132
        GO TO 150                                                               OU1  133
140     WRITE (6,600) CHANAM,DIAM(1),DIAM(2),FACNAM                             OU1  134
150     WRITE (6,610) POUT,TOUT,HOUT,SOUT,ROUT,UOUT,FLX,TFLOW,GAMMA,CM,DELOU1  135
        WRITE (6,620)                                                           OU1  136
        WRITE (6,630) (HP(I),SAVEC(I),I=1,ISS)                                  OU1  137
        RETURN                                                                  OU1  138
160     FVOUT(1)=CF(4)*CX                                                       OU1  139
        FVOUT(2)=CT*CTAP                                                        OU1  140
        FVOUT(3)=CF(5)*CH                                                       OU1  141
        FVOUT(4)=PRES*PRESA                                                     OU1  142
        FVOUT(5)=CF(1)*RHO                                                      OU1  143
        FVOUT(7)=CF(2)*SU                                                       OU1  144
        FVOUT(8)=AMACH                                                          OU1  145
        FVOUT(9)=SEN                                                            OU1  146
        FVOUT(11)=AFNX                                                          OU1  147
        I6=I6+1                                                                 OU1  148
        IF (ISW3B.EQ.0) GO TO 180                                               OU1  149
        DO 170 I=1,NPRFLS                                                       OU1  150
        II=7+10*I                                                               OU1  151
        FVOUT(II)=CF(6)*DELBL(I)                                                OU1  152
        FVOUT(II+1)=CF(6)*THETA(I)                                              OU1  153
        FVOUT(II+2)=QWDOT(I)                                                    OU1  154
        FVOUT(II+3)=TAUW(I)                                                     OU1  155
        FVOUT(II+4)=1.8*HR                                                      OU1  156
170     FVOUT(II+5)=PRREF                                                       OU1  157
180     CONTINUE                                                                OU1  158
        VARNAM(6)=DNAM                                                          OU1  159
        VARNAM(16)=ANAM                                                         OU1  160
        IF (NPRFLS.EQ.1) GO TO 200                                              OU1  161
        CALL GVAR2 (CX,Y,Z)                                                     OU1  162
        IF (ERR) RETURN                                                         OU1  163
        IF (ISW3B.NE.0) GO TO 190                                               OU1  164
        CHDIMS(2)=2.*Y*CF(4)                                                    OU1  165
```

PRECEDING PAGE BLANK NOT FILMED

```
      CHDIMS(1)=2.*Z*CF(4)                                              OU1  166
      FVOUT(6)=1.E20                                                    OU1  167
      GO TO 200                                                         OU1  168
  190 FVOUT(16)=2.*Z*CF(4)                                              OU1  169
      FVOUT(26)=2.*Y*CF(4)                                              OU1  170
      VARNAM(16)=WNAM                                                   OU1  171
  200 CONTINUE                                                          OU1  172
      FVOUT(10)=GAMMA                                                   OU1  173
      IF (ISW38.NE.0) GO TO 250                                         OU1  174
      IF (NPRFLS.EQ.2) GO TO 210                                        OU1  175
      CALL GMAR (CX,Y)                                                  OU1  176
      IF (ERR) RETURN                                                   OU1  177
      FVOUT(6)=2.*Y*CF(4)                                               OU1  178
  210 IF (NOTRAN) GO TO 220                                             OU1  179
      CALL TRANSP (FVOUT(2),FVOUT(4))                                   OU1  180
      IF (ERR) RETURN                                                   OU1  181
  220 FVOUT(14)=VISC*CF(3)                                              OU1  182
      IF (VISC.EQ.0.) GO TO 230                                         OU1  183
      FVOUT(12)=FVOUT(5)*FVOUT(7)/FVOUT(14)                             OU1  184
      GO TO 240                                                         OU1  185
  230 FVOUT(12)=0.                                                      OU1  186
  240 FVOUT(15)=SIGMA                                                   OU1  187
      GO TO 260                                                         OU1  188
  250 FVOUT(12)=REPF                                                    OU1  189
      FVOUT(14)=CF(3)*VISC                                              OU1  190
      FVOUT(15)=SIGMA                                                   OU1  191
  260 FVOUT(13)=CM                                                      OU1  192
      HO=1.8*CHA*CRP/CMA                                                OU1  193
      IF (ISW38.EQ.0) GO TO 340                                         OU1  194
      IF (FVOUT(7).EQ.0.) GO TO 270                                     OU1  195
      STFAC=1./(FVOUT(5)*FVOUT(7)*(HO-HW))                              OU1  196
  270 FVOUT(23)=STFAC*FVOUT(19)                                         OU1  197
      FVOUT(24)=FVOUT(12)*FVOUT(18)/12.                                 OU1  198
      FVOUT(25)=200.*EXP(0.224*FVOUT(8))                                OU1  199
      IF (NPRFLS.EQ.1) GO TO 280                                        OU1  200
      FVOUT(33)=STFAC*FVOUT(29)                                         OU1  201
      FVOUT(34)=FVOUT(12)*FVOUT(28)/12.                                 OU1  202
      FVOUT(35)=FVOUT(25)                                               OU1  203
  280 IF (LIMOUT.EQ.0) GO TO 300                                        OU1  204
      DO 290 I=1,NPRFLS                                                 OU1  205
      FVO2(1,I)=SN(I)                                                   OU1  206
  290 FVO2(2,I)=XSN(I)                                                  OU1  207
  300 CALL GEOMAR (CX,ARAT,DARAT)                                       OU1  208
      IF (ERR) RETURN                                                   OU1  209
      IF (NPRFLS.EQ.2) GO TO 310                                        OU1  210
      FVOUT(16)=ARAT                                                    OU1  211
      GO TO 320                                                         OU1  212
  310 FVOUT(6)=ARAT                                                     OU1  213
      VARNAM(6)=ANAM                                                    OU1  214
      GO TO 340                                                         OU1  215
  320 IF (JDIM.EQ.0) GO TO 330                                          OU1  216
      FVOUT(6)=RTHCM(1)*SQRT(ARAT)*CF(4)*2.                             OU1  217
      GO TO 340                                                         OU1  218
  330 FVOUT(6)=RTHCM(1)*ARAT*CF(4)*2.                                   OU1  219
  340 INEQP1=INEQ+1                                                     OU1  220
```

```
      WRITE (6,640) (ASTRSK,I=1,8),(ASOLN(J,ISOLN),J=1,3),(ASTRSK,K=1,3)OU1  221
     1,NSTEPS,STEP(ISOLN),(ASTRSK,L=1,4),TYPSLN(INEQP1,ISOLN)             OU1  222
      WRITE (6,650) (VARNAM(I),FVOUT(I),I=1,15)                           OU1  223
      IF (ISW3B.EQ.0) GO TO 360                                           OU1  224
      DO 350 L=1,NPRFLS                                                   OU1  225
      I1=6+10*L                                                           OU1  226
      I2=I1+9                                                             OU1  227
      WRITE (6,660) (VARNAM(I),FVOUT(I),I=I1,I2)                          OU1  228
  350 CONTINUE                                                            OU1  229
  360 IF (NPRFLS.EQ.1.OR.ISW3B.NE.0) GO TO 370                            OU1  230
      WRITE (6,670) CHDIMS(1),CHDIMS(2)                                   OU1  231
  370 IF (LIMOUT.EQ.0) GO TO 380                                          OU1  232
      WRITE (6,560) ((VARN2(I),J,FVO2(I,J),I=1,2),J=1,NPRFLS)             OU1  233
  380 IF (NT.EQ.1.OR.ISOLN.NE.3) GO TO 390                                OU1  234
      TNOUT(1)=CTAP*CTE                                                   OU1  235
      TNOUT(2)=QDPR                                                       OU1  236
      TNOUT(3)=QDPE                                                       OU1  237
      TNOUT(4)=H0                                                         OU1  238
      WRITE (6,680) (TNNAM(I),TNOUT(I),I=1,4)                             OU1  239
  390 IF (ISW6B.EQ.0.OR.ISOLN.EQ.1) GO TO 440                             OU1  240
      IF (I6.LT.IS6) GO TO 440                                            OU1  241
      I6=0                                                                OU1  242
      WRITE (6,690)                                                       OU1  243
      IF (IC.EQ.0) GO TO 400                                              OU1  244
      EMF=SAVEC(1)                                                        OU1  245
      SAVEC(1)=EMF*RHO*TB7/CM                                             OU1  246
  400 ISS5P1=(ISS-1)/5+1                                                  OU1  247
      DO 410 K=1,ISS5P1                                                   OU1  248
      LIM1=5*(K-1)+1                                                      OU1  249
      LIM2=MIN0(5*K,ISS)                                                  OU1  250
  410 WRITE (6,700) (HP(I),SAVEC(I),I=LIM1,LIM2)                          OU1  251
      IF (IC.EQ.0) GO TO 420                                              OU1  252
      SAVEC(1)=EMF                                                        OU1  253
  420 IF (ISW6B.GT.0.OR.ISOLN.NE.3) GO TO 440                             OU1  254
      WRITE (6,510)                                                       OU1  255
      WRITE (6,520) (I,PI(I),I=1,ISR)                                     OU1  256
      WRITE (6,530) (I,CHI(I),I=1,ISR)                                    OU1  257
      WRITE (6,540) (I,PICHI(I),I=1,ISR)                                  OU1  258
      DO 430 I=1,ISS                                                      OU1  259
  430 DLGDX(I)=DGJ(I)/GJ(I)                                               OU1  260
      WRITE (6,550) (HP(I),DLGDX(I),I=1,ISS)                              OU1  261
  440 IF (.NOT.DATAPE) GO TO 500                                          OU1  262
      IF (ITYPER.NE.1) GO TO 450                                          OU1  263
      IMP=0                                                               OU1  264
      IX=0                                                                OU1  265
  450 IX=IX+1                                                             OU1  266
      NRCOUT=NRCOUT+1                                                     OU1  267
      ITYPER=2                                                            OU1  268
      XXX(1)=FVOUT(1)                                                     OU1  269
      XXX(2)=FVOUT(2)                                                     OU1  270
      XXX(3)=FVOUT(4)                                                     OU1  271
      XXX(4)=FVOUT(5)                                                     OU1  272
      XXX(5)=FVOUT(7)                                                     OU1  273
      XXX(6)=FVOUT(3)                                                     OU1  274
      XXX(7)=FVOUT(8)                                                     OU1  275
```

```
      IF (ISW3D) 460,480,460                                                    OU1  276
460   IBL=17+10*(NBL-1)                                                         OU1  277
      XXX(9)=FVOUT(IBL)                                                         OU1  278
      IF (NPRFLS.EQ.2) GO TO 470                                                OU1  279
      XXX(8)=FVOUT(16)                                                          OU1  280
      GO TO 490                                                                 OU1  281
470   XXX(8)=FVOUT(6)                                                           OU1  282
      GO TO 490                                                                 OU1  283
480   XXX(8)=FVOUT(11)                                                          OU1  284
      XXX(9)=0.                                                                 OU1  285
490   XXX(10)=FVOUT(12)                                                         OU1  286
      WRITE (ITPOUT) ITYPER,IX,(XXX(I),I=1,10)                                  OU1  287
500   RETURN                                                                    OU1  288
C                                                                               OU1  289
C                                                                               OU1  290
C                                                                               OU1  291
510   FORMAT (57X,18HREACTION RATE DATA)                                        OU1  292
520   FORMAT (3H PI,11X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)                         OU1  293
530   FORMAT (4H CHI,10X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)                        OU1  294
540   FORMAT (6H PICHI,6X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)                       OU1  295
550   FORMAT (4H DLG,12X,7(2X,A4,1PE10.1)/8(2X,A4,1PE10.1),4X)                  OU1  296
560   FORMAT (1X,A3,1H(,I1,3H) =,4X,1PE10.3,4X,3(A3,1H(,I1,3H) =,4X,E10.0OU1  297
     13,4X))                                                                    OU1  298
570   FORMAT (1H0)                                                              OU1  299
580   FORMAT (19H GAS FLOW RATE     =,F10.3,2X,7H LB/SEC)                       OU1  300
590   FORMAT (10H NOZZLE - ,A4,5X,F10.3,2X,21H INCH THROAT DIAMETER)            OU1  301
600   FORMAT (9H CHANNEL ,A4,6X,F10.3,3H BY,F7.3,17H INCH THROAT FOR ,A4OU1  302
     1)                                                                         OU1  303
610   FORMAT (19H PRESSURE          =,F10.3,2X,4H ATM/19H TEMPERATURE     OU1  304
     1   =,F7.0,5X,6H DEG K/19H ENTHALPY          =,F7.0,5X,7H BTU/LB/,9H OU1  305
     2ENTROPY          =,F9.2,3X,13H BTU/LB-DEG R/19H DENSITY          =OU1  306
     3,F12.5,9H LB/CU FT/19H VELOCITY          =,F7.0,5X,7H FT/SEC/19H MAOU1  307
     4SS FLUX        =,F10.3,2X,13H LB/SQ FT-SEC/19H COMPUTED FLOW     =,OU1  308
     5F10.3,2X,7H LB/SEC/19H GAMMA             =,F10.3/19H MOLECULAR WEIGOU1  309
     6HT =,F9.2,3X,8H GM/MOLE/19H ELECTRON DENSITY =,3X,1PE9.2,13H ELECTOU1  310
     7RONS/CC)                                                                  OU1  311
620   FORMAT (//10X,22HSPECIES MOLE FRACTIONS/)                                 OU1  312
630   FORMAT (13X,A4,4X,1PE10.3)                                                OU1  313
640   FORMAT (//6H *****,14A6,I6,6A6//)                                         OU1  314
650   FORMAT (1X,A6,2H =,F11.3,7X,A6,2H =,F8.0,10X,A6,2H =,F8.0,10X,A6,2OU1  315
     1H =,4X,1PE10.3,4X,A6,2H =,4X,E10.3/1X,A6,2H =,0PF11.3,7X,A6,2H =,FOU1  316
     28.0,10X,A6,2H =,F11.3,7X,A6,2H =,F10.2,8X,A6,2H =,F11.3/1X,A6,2H =OU1  317
     3,F11.3,7X,A6,2H =,4X,1PE10.3,4X,A6,2H =,0PF10.2,8X,A6,2H =,4X,1PE1OU1  318
     40.3,4X,A6,2H =,4X,F10.3)                                                  OU1  319
660   FORMAT (1X,A6,2H =,F11.3,7X,A6,2H =,F11.3,7X,A6,2H =,F11.3,7X,A6,2OU1  320
     1H =,4X,1PE10.3,4X,A6,2H =,4X,E10.3/1X,A6,2H =,0PF8.0,10X,A6,2H =,FOU1  321
     212.4,6X,A6,2H =,4X,1PE10.3,4X,A6,2H =,0PF8.0,10X,A6,2H =,F8.0)           OU1  322
670   FORMAT (9H WIDTH  =,F11.3,6X,9H HEIGHT =,F11.3)                           OU1  323
680   FORMAT (1X,A6,2H =,F8.0,10X,A6,2H =,4X,1PE10.3,4X,A6,2H =,4X,E10.3OU1  324
     1,4X,A6,2H =,4X,F10.3)                                                     OU1  325
690   FORMAT (45X,41HSPECIES MOLE FRACTIONS IN THE FREE STREAM)                  OU1  326
700   FORMAT (5(1X,A4,3X,1H=,4X,1PE10.3,3X))                                    OU1  327
      END                                                                       OU1  328-
```

```
      SUBROUTINE PERT                                                   PER    1
      LOGICAL ERR                                                       PER    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   PER    3
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              PER    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             PER    5
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  PER    6
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  PER    7
     2  CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,    PER    8
     3   CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,    PER    9
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,   PER   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,   PER   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,   PER   12
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,   PER   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  PER   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),PER   15
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),PER   16
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),PER   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),   QM(20),PER   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),PER   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),PER   20
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64), PER   21
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),         PER   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       PER   23
      COMMON       IC,     IM,   INFO,  INEQV,    IP,   IRUN,    ISC, ISCP1,  PER   24
     1  ISMC,ISMCNR,    ISR,    ISS,  ISSNR,  ISSP1,  ISSP2,  ISSP3, ISSP4,  PER   25
     2 ISW1A, ISW1B,  ISW2A,  ISW2B,  ISW3A,  ISW3B,  ISW4A,  ISW4B, ISW5A,  PER   26
     3 ISW5B, ISW6A,  ISW6B,   IUPD,  IZERO,    JJK,     LC,    M1,   NFIT,  PER   27
     4   NIT,   NNN,   NNS,    NQS,    NQT, NTEST                       PER   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) PER   29
      COMMON ACOM,ELMENT,HP                                             PER   30
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBPER   31
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                PER   32
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              PER   33
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),       PER   34
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),            PER   35
     3      (  JA(1),SHJAP(1)),(THEV(1),THEVP(1))                       PER   36
      COMMON /ERROR/ ERR                                                PER   37
      DO 10 I=1,ISSP2                                                   PER   38
      DO 10 J=1,ISSP3                                                   PER   39
10    AA(I,J)=0.                                                        PER   40
      DO 20 I=1,ISC                                                     PER   41
      DO 20 J=1,ISS                                                     PER   42
20    AA(I,J)=LPIJ(J,I)                                                 PER   43
      DO 40 I=ISCP1,ISS                                                 PER   44
      DO 40 J=1,ISS                                                     PER   45
      Z=0.                                                              PER   46
      DO 30 K=1,ISR                                                     PER   47
30    Z=Z+(BETA(K,I)*PI(K)*BETA(K,J))                                   PER   48
      Z=Z/GJ(J)                                                         PER   49
40    AA(I,J)=-Z                                                        PER   50
      DO 50 J=1,ISS                                                     PER   51
50    AA(ISSP1,J)=(CMA*SHJ(J)/(SU2))                                    PER   52
      DO 80 I=ISCP1,ISS                                                 PER   53
      Z1=0.                                                             PER   54
      DO 70 K=1,ISR                                                     PER   55
```

```
      Z=0.                                                                  PER   56
      DO 60 J=1,ISS                                                         PER   57
60    Z=Z+(BETA(K,J))*(SHJ(J)/CT-1.)                                        PER   58
70    Z1=Z1+Z*(BETA(K,I)*PI(K))                                             PER   59
80    AA(I,ISSP1)=Z1/CT                                                     PER   60
      Z=0.                                                                  PER   61
      DO 90 J=1,ISS                                                         PER   62
90    Z=Z+GJ(J)*CCPJ(J)                                                     PER   63
      AA(ISSP1,ISSP1)=(CMA*Z)/(SU2)                                         PER   64
      DO 110 I=ISCP1,ISS                                                    PER   65
      Z1=0.                                                                 PER   66
      DO 100 K=1,ISR                                                        PER   67
100   Z1=Z1+(BETA(K,I)*PI(K)*BE(K))                                         PER   68
110   AA(I,ISSP2)=-Z1/RHO                                                   PER   69
      AA(ISSP1,ISSP2)=-1./RHO                                               PER   70
      DO 120 I=1,ISS                                                        PER   71
120   AA(ISSP2,I)=SENT(I)-1.0-ALOG(GJ(I)*RHO*CT*82.055*RHAP*CTAP)           PER   72
      Z=0.                                                                  PER   73
      DO 130 J=1,ISS                                                        PER   74
130   Z=Z+GJ(J)*(CCPJ(J)-1.)                                                PER   75
      AA(ISSP2,ISSP1)=Z/CT                                                  PER   76
      AA(ISSP2,ISSP2)=-1./(CM*RHO)                                          PER   77
      DO 140 I=ISCP1,ISS                                                    PER   78
140   AA(I,ISSP3)=DGJ(I)                                                    PER   79
      CALL DSMSOL (AA,ISSP2,22)                                             PER   80
      IF (ERR) RETURN                                                       PER   81
      DO 150 I=1,ISS                                                        PER   82
150   PERTGJ(I)=AA(I,ISSP3)                                                 PER   83
      PCT=AA(ISSP1,ISSP3)                                                   PER   84
      PRHO=AA(ISSP2,ISSP3)                                                  PER   85
      DO 170 I=1,ISR                                                        PER   86
      Z=0.                                                                  PER   87
      Z1=0.                                                                 PER   88
      DO 160 J=1,ISS                                                        PER   89
      Z=Z+BETA(I,J)*((SHJ(J)/CT)-1.)                                        PER   90
160   Z1=Z1+(BETA(I,J)*PERTGJ(J))/GJ(J)                                     PER   91
170   SDCHI(I)=-BE(I)*PRHO/RHO+Z*PCT/CT-Z1                                  PER   92
      RETURN                                                                PER   93
      END                                                                   PER   94-
```

```
      FUNCTION PIOMEG(TE)                                                PIO    1
      COMMON /TNEQ/ TLIST(30),POM(30)                                    PIO    2
      DO 10 I=2,30                                                       PIO    3
      IF (TE.LT.TLIST(I)) GO TO 20                                       PIO    4
10    CONTINUE                                                           PIO    5
      PIOMEG=POM(30)                                                     PIO    6
      GO TO 30                                                           PIO    7
20    PIOMEG=POM(I-1)+(POM(I)-POM(I-1))*(TE-TLIST(I-1))/(TLIST(I)-TLIST(PIO    8
     1I-1))                                                              PIO    9
30    RETURN                                                             PIO   10
      END                                                                PIO   11-
```

```
      SUBROUTINE PROP                                                    PRO    1
      REAL ACOM(30),ELMENT(10),HP(20)                                    PRO    2
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                               PRO    3
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                              PRO    4
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,  PRO    5
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  PRO    6
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  PRO    7
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  PRO    8
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  PRO    9
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  PRO   10
     6RHOBAR,  RHOC,  RHOP,  RHPL, RHTH,ROBARA,ROBARP,  SCPG,   SDT,  PRO   11
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  PRO   12
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                 PRO   13
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),PRO   14
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),PRO   15
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),PRO   16
     3    GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),PRO   17
     4     QQ(64),    SAJ(20), SDCHI(64),   SENT(20),   SHJ(20),  SHJA(20),PRO   18
     5   SKIL(20),     SS(20),     TB(30),   TFA(20),   TFB(20),   TFC(20),PRO   19
     6    TFD(20),    TFE(20),    TFK(20), THEV(20), XMJAT(20),  XNUI(64)  PRO   20
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),           PRO   21
     1 XNUIJ(64,20),XNUIJP(64,20)                                        PRO   22
      COMMON       IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  PRO   23
     1   ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  PRO   24
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,  PRO   25
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,     M1,   NFIT,  PRO   26
     4    NIT,    NNN,    NNS,    NQS,    NQ1, NTEST                     PRO   27
      COMMON    IGJ(20),    IGM(20),    ITB(5),    KUR(64,20),  LPIJ(20,10) PRO   28
      COMMON ACOM,ELMENT,HP                                              PRO   29
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBPRO   30
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                 PRO   31
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),               PRO   32
     1     (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),        PRO   33
     2     (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),             PRO   34
     3     (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                        PRO   35
      DATA RNAME /4HPROP/                                                PRO   36
      S4=0.                                                              PRO   37
      SHPG=0.                                                            PRO   38
      DO 10 J=1,ISS                                                      PRO   39
      S4=S4+GJA(J)*SENT(J)                                               PRO   40
   10 SHPG=SHPG+GJA(J)*SHJ(J)                                            PRO   41
      ZP=CMA*(S4+ENT)                                                    PRO   42
      IF (ZP.LT.80.) GO TO 20                                            PRO   43
      CALL DUMP (RNAME)                                                  PRO   44
      RETURN                                                             PRO   45
   20 PRES=EXP(ZP)                                                       PRO   46
      CH=CMA*SHPG                                                        PRO   47
      RHOBAR=PRES/CT                                                     PRO   48
      RHO=(RHOBAR*ROBARA)/(1.+RHOBAR*ROBARP*BZERO)                       PRO   49
      CH=CH+(PRES/RHOBAR)*(RHOBAR*ROBARA/RHO-1.0)                        PRO   50
      SU2=2.*(CHA-CH)                                                    PRO   51
      SU=SQRT(SU2)                                                       PRO   52
      FLUX=RHO*SU                                                        PRO   53
      RETURN                                                             PRO   54
      END                                                                PRO   55-
```

```
      SUBROUTINE PRTA                                                   PRT    1
      LOGICAL ERR,MODLPT                                                PRT    2
      REAL ACOM(30),ELMENT(10),HP(20)                                   PRT    3
      DOUBLE PRECISION AA,CAPX,CDIJ                                     PRT    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             PRT    5
      COMMON       A0, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,  PRT    6
     1      C,  CARL,     CH,    CHA,  CLNT,     CM,    CMA,    CRA,   CRP,  PRT    7
     2   CRRB,   CRS,   CSTA,     CT,  CTAP,    CTB,    CTC, CTMAX, CTMXX,  PRT    8
     3    CTP,  CTPL,    CTT,     CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  PRT    9
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,   ENT,   FLUX, HDELX,   PCT,  PRT   10
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,    RHO,  RHOB,  PRT   11
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  PRT   12
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  PRT   13
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  PRT   14
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),PRT   15
     1  CCPJ(20), CFACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),PRT   16
     2CLN1MC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),PRT   17
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),PRT   18
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),PRT   19
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),PRT   20
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) PRT   21
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),          PRT   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       PRT   23
      COMMON      IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  PRT   24
     1  ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  PRT   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,  PRT   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,    JJK,    LC,    M1,  NFIT,  PRT   27
     4   NIT,    NNN,    NNS,    NQS,    NQT, NTEST                     PRT   28
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) PRT   29
      COMMON ACOM,ELMENT,HP                                             PRT   30
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  PRT   31
     1      SW,R0,JDIM,IPCINT                                           PRT   32
      COMMON /NEQ/ OMDST(2),DDELBL(2)                                   PRT   33
      COMMON /SS/ CAS,US                                                PRT   34
      COMMON /ERROR/ ERR                                                PRT   35
      COMMON /MODP2/ MODLPT                                             PRT   36
      COMMON /STEPS/ NSTEPS                                             PRT   37
      COMMON /TNONEQ/ CTF,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),       PRT   38
     1      KTR(25),ICH(20),IPA(25)                                     PRT   39
      NAMELIST /PRTDMP/ AFNTS,      CAPX,CHI,DLOGA,DLOGR,DT,IUPD,       PRT   40
     1      PCT,PERTGJ,PI,PRHO,SDCHI,CAS ,DELBL,OMDST,DDELBL            PRT   41
      NSTEPS=NSTEPS+1                                                   PRT   42
      IF (ISW3B.EQ.0) GO TO 10                                          PRT   43
      AS=SU*US/AMACH                                                    PRT   44
      CAS=AS*AS/CT                                                      PRT   45
   10 CALL BLCALL (.TRUE.)                                              PRT   46
      IF (ERR) RETURN                                                   PRT   47
      IF (ISW5B.NE.0) WRITE (6,PRTDMP)                                  PRT   48
      IF (INEQ.NE.0) GO TO 20                                           PRT   49
      CT=CT+PCT                                                         PRT   50
      CTE=CT                                                            PRT   51
      RHO=RHO+PRHO                                                      PRT   52
   20 IF (ABS(CT-CTT).LT.TPRINT.AND..NOT.MODLPT) GO TO 30               PRT   53
      CTT=CT                                                            PRT   54
      CALL OUT2 (3)                                                     PRT   55
```

```
      IF (ERR) RETURN                                                   PRT   56
      NSTEPS=0                                                          PRT   57
      IF (.NOT.MODLPT) GO TO 30                                         PRT   58
      CALL MODEL                                                        PRT   59
      MODLPT=.FALSE.                                                    PRT   60
30    IF (INEQ.NE.0) GO TO 40                                           PRT   61
      CT=CT-PCT                                                         PRT   62
      CTE=CT                                                            PRT   63
      RHO=RHO-PRHO                                                      PRT   64
40    RETURN                                                            PRT   65
C                                                                       PRT   66
      END                                                               PRT   67-
```

```
      SUBROUTINE PUTQIN (X)                                              PUT    1
C     SUBROUTINE PUTQIN SUPERVISES THE COMPUTATION OF COLLISION CROSS    PUT    2
C     SECTIONS FOR THE TRANSPORT CALCULATIONS.                           PUT    3
      LOGICAL WROTE                                                      PUT    4
      DIMENSION OM(3)                                                    PUT    5
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                               PUT    6
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ         PUT    7
      COMMON /TRANS8/ NV(15),N                                           PUT    8
      COMMON /RDTR/ ISW88                                                PUT    9
      DATA IDBG /-1/                                                     PUT   10
      IDBG=IDBG+1                                                        PUT   11
      WROTE=.FALSE.                                                      PUT   12
C     INITIALIZE Q ARRAY TO ZERO.                                        PUT   13
      DO 10 I=1,N                                                        PUT   14
      DO 10 J=1,N                                                        PUT   15
      DO 10 K=1,3                                                        PUT   16
10    Q(K,I,J)=0.                                                        PUT   17
      CALL QCOUL (OM,X,V)                                                PUT   18
      Q(1,2,1)=OM(1)                                                     PUT   19
C     COMPUTE CROSS SECTION VALUES AND ACCUMULATE IN Q ARRAY.            PUT   20
      M=2                                                                PUT   21
      LQ1=1                                                              PUT   22
      DO 240 L=1,NKQ                                                     PUT   23
      K=KQ(L)                                                            PUT   24
      LQ2=NQ(L)                                                          PUT   25
      GO TO (20,30,40,50,60,70,80,90,100,110,120,130,140,150,160), K     PUT   26
20    CONTINUE                                                           PUT   27
      GO TO 190                                                          PUT   28
30    OM(1)=V(M)*Q(1,2,1)                                                PUT   29
      OM(3)=OM(1)*1.5625                                                 PUT   30
      OM(2)=V(M+1)*OM(1)                                                 PUT   31
      GO TO 190                                                          PUT   32
40    CALL QEXP (OM,V(M))                                                PUT   33
      GO TO 190                                                          PUT   34
50    CALL QEX (OM,V(M))                                                 PUT   35
      IF (V(M+3)) 190,190,170                                            PUT   36
60    CALL QTAB (OM,V(M))                                                PUT   37
      GO TO 190                                                          PUT   38
70    CALL QREPP (OM,V(M))                                               PUT   39
      GO TO 190                                                          PUT   40
80    CONTINUE                                                           PUT   41
C     17   CALL QATTP (OM,V(M))                                         PUT   42
      GO TO 190                                                          PUT   43
90    CALL QLJ (OM,V(M))                                                 PUT   44
      GO TO 190                                                          PUT   45
100   CALL QSAME (OM,V(M))                                               PUT   46
      GO TO 190                                                          PUT   47
110   CALL QMIX (LQ1,LQ2)                                                PUT   48
      GO TO 210                                                          PUT   49
120   CALL Q11 (LQ1,LQ2,V(M))                                            PUT   50
      GO TO 210                                                          PUT   51
130   CALL Q12 (OM,V(M))                                                 PUT   52
      GO TO 190                                                          PUT   53
140   CALL Q13 (LQ1,LQ2,V(M))                                            PUT   54
      GO TO 210                                                          PUT   55
```

```
   15   CALL Q14 (LQ1,LQ2,V(M))                                          PUT   56
        GO TO 210                                                        PUT   57
  160   CONTINUE                                                         PUT   58
        GO TO 210                                                        PUT   59
  170   DO 180 LQ=LQ1,LQ2                                                PUT   60
        I=IQ(LQ)                                                         PUT   61
        J=JQ(LQ)                                                         PUT   62
        Q(1,I,J)=QM(1)                                                   PUT   63
  180   Q(3,I,J)=QM(3)                                                   PUT   64
        GO TO 210                                                        PUT   65
  190   DO 200 LQ=LQ1,LQ2                                                PUT   66
        I=IQ(LQ)                                                         PUT   67
        J=JQ(LQ)                                                         PUT   68
        DO 200 KK=1,3                                                    PUT   69
  200   Q(KK,I,J)=Q(KK,I,J)+QM(KK)                                       PUT   70
  210   M=M+NV(K)                                                        PUT   71
  C     DEBUG OUTPUT                                                     PUT   72
        IF (ISW8B.EQ.0) GO TO 240                                        PUT   73
        IF (IDBG.EQ.0) GO TO 220                                         PUT   74
        IF (IDBG.NE.ISW8B) GO TO 240                                     PUT   75
  220   WROTE=.TRUE.                                                     PUT   76
        WRITE (6,250) L,K,NQ(L),M,LQ1                                    PUT   77
        DO 230 I=1,N                                                     PUT   78
  230   WRITE (6,260) (Q(1,J,I),J=1,I)                                   PUT   79
  240   LQ1=LQ2+1                                                        PUT   80
        IF (WROTE) IDBG=0                                                PUT   81
        RETURN                                                           PUT   82
  C                                                                      PUT   83
  250   FORMAT (1HC/10I5/)                                               PUT   84
  260   FORMAT (1P10E12.4)                                               PUT   85
        END                                                              PUT   86
```

```
      SUBROUTINE QCOUL (OM1,X,C)                                        QCO    1
C     SUBROUTINE QCOUL COMPUTES THE FACTOR 0.8*QC IN THE EFFECTIVE      QCO    2
C     COULOMB CROSS SECTION.                                            QCO    3
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              QCO    4
      OM1=2.2346E10/T**2                                                QCO    5
      IF (C) 20,20,10                                                   QCO    6
10    Y=(T**4)/(1.E15*X+1.E-35*(T**4))                                  QCO    7
      OM1=.5*OM1*ALOG(2.091E1*Y+1.517E2*Y**.666667)                     QCO    8
20    RETURN                                                            QCO    9
      END                                                               QCO   10-
```

```
      SUBROUTINE QEX (OM,V)                                              QEX    1
C     SUBROUTINE QEX COMPUTES CROSS SECTIONS FOR A RESONANT EXCHANGE     QEX    2
C     PROCESS.                                                           QEX    3
      DIMENSION OM(3),V(3)                                               QEX    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                               QEX    5
      A=V(1)                                                             QEX    6
      B=V(2)                                                             QEX    7
      ZM=V(3)                                                            QEX    8
      Y=A-.217147*B*ALOG(T/ZM)                                           QEX    9
      OM(1)=2.*(Y-4.461*B)**2+.03724*B**2                                QEX   10
      OM(3)=2.*(Y-4.317*B)**2-.06754*B**2                                QEX   11
      OM(2)=0.                                                           QEX   12
      RETURN                                                             QEX   13
      END                                                                QEX   14-
```

```
      SUBROUTINE QEXP (OM,V)                                          QXP    1
C     SUBROUTINE QEXP COMPUTES CROSS SECTIONS FOR AN EXPONENTIAL      QXP    2
C     INTERACTION POTENTIAL.                                          QXP    3
      DIMENSION OM(3),V(3)                                            QXP    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                            QXP    5
      A=V(1)                                                          QXP    6
      RHO=V(2)                                                        QXP    7
      N1=V(3)                                                         QXP    8
      ALPHA=ALOG(A/T)                                                 QXP    9
      CALL QINTRP (ALPHA,OM,N1,50)                                    QXP   10
      OM(1)=12.5664*(ALPHA*RHO)**2*OM(1)                              QXP   11
      OM(2)=OM(2)*OM(1)                                               QXP   12
      OM(3)=OM(3)*OM(1)                                               QXP   13
      RETURN                                                          QXP   14
      END                                                             QXP   15-
```

```
      SUBROUTINE QINTRP (A,B,N1,N)                                      QIN    1
C     SUBROUTINE QINTRP INTERPOLATES TABULAR DATA FOR THE CROSS SECTION  QIN    2
C     COMPUTATIONS                                                      QIN    3
      COMMON /RDTR/ ISW8B                                               QIN    4
      DIMENSION B(3)                                                    QIN    5
      COMMON /TRANS4/ TL(1000),OM(1000,3)                               QIN    6
      I=N1                                                              QIN    7
      IF (TL(I)-A) 10,40,30                                             QIN    8
10    N2=N1+N-2                                                         QIN    9
      DO 20 I=N1,N2                                                     QIN   10
      IF (TL(I+1)-A) 20,40,40                                           QIN   11
20    CONTINUE                                                          QIN   12
      I=N2                                                              QIN   13
30    IF (ISW8B.EQ.0) GO TO 40                                          QIN   14
      WRITE (6,60) I,A,TL(I),TL(I+1)                                    QIN   15
40    DO 50 J=1,3                                                       QIN   16
50    B(J)=OM(I+1,J)+(A-TL(I+1))*(OM(I,J)-OM(I+1,J))/(TL(I)-TL(I+1))    QIN   17
      RETURN                                                            QIN   18
C                                                                       QIN   19
60    FORMAT (25H0      EXTRAPOLATED VALUESI6,1P4E14.4/1H )             QIN   20
      END                                                               QIN   21-
```

```
      SUBROUTINE QLJ (OM,V)                                              QLJ    1
C     SUBROUTINE QLJ COMPUTES CROSS SECTIONS FOR THE LENNARD-JONES 6-12  QLJ    2
C     POTENTIAL.                                                         QLJ    3
      DIMENSION OM(3), V(3)                                              QLJ    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                               QLJ    5
      EPSLN=V(1)                                                         QLJ    6
      SIGMA=V(2)                                                         QLJ    7
      N1=V(3)                                                            QLJ    8
      TSTAR=T/EPSLN                                                      QLJ    9
      CALL QINTRP (TSTAR,OM,N1,37)                                       QLJ   10
      OM(1)=3.1416*SIGMA**2*OM(1)                                        QLJ   11
      OM(2)=OM(2)*OM(1)                                                  QLJ   12
      OM(3)=OM(3)*OM(1)                                                  QLJ   13
      RETURN                                                             QLJ   14
      END                                                                QLJ   15-
```

```
      SUBROUTINE QMIX (LQ1,LQ2)                                          QMX    1
C     SUBROUTINE QMIX COMPUTES CROSS SECTIONS FROM THE EMPIRICAL MIXING  QMX    2
C     RULE                                                               QMX    3
      DIMENSION SQT(3,20)                                                QMX    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                               QMX    5
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ         QMX    6
      IF (LQ2.LT.LQ1) GO TO 70                                           QMX    7
      DO 10 I=1,20                                                       QMX    8
10    SQT(1,I)=0.                                                        QMX    9
      DO 60 LQ=LQ1,LQ2                                                   QMX   10
      I=IQ(LQ)                                                           QMX   11
      J=JQ(LQ)                                                           QMX   12
      IF (SQT(1,I).GT.0.) GO TO 30                                       QMX   13
      DO 20 K=1,3                                                        QMX   14
20    SQT(K,I)=.5*SQRT(Q(K,I,I))                                         QMX   15
30    IF (SQT(1,J).GT.0.) GO TO 50                                       QMX   16
      DO 40 K=1,3                                                        QMX   17
40    SQT(K,J)=.5*SQRT(Q(K,J,J))                                         QMX   18
50    DO 60 K=1,3                                                        QMX   19
60    Q(K,I,J)=(SQT(K,I)+SQT(K,J))**2+Q(K,I,J)                           QMX   20
70    RETURN                                                             QMX   21
      END                                                                QMX   22
```

```
      SUBROUTINE OREPP (OM,VV)                                          ORE    1
C     SUBROUTINE OREPP COMPUTES CROSS SECTIONS FOR AN INVERSE POWER LAW  ORE    2
C     INTERACTION POTENTIAL.                                            ORE    3
      DIMENSION OM(3),VV(3)                                             ORE    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              ORE    5
      COMMON /TRANS4/ TL(1000),OMEGA1(1000),ASTAR(1000),BSTAR(1000)     ORE    6
      ITL=VV(1)                                                         ORE    7
      OM(1)=OMEGA1(ITL)*T**(-2./VV(2))                                  ORE    8
      OM(2)=ASTAR(ITL)*OM(1)                                            ORE    9
      OM(3)=BSTAR(ITL)*OM(1)                                            ORE   10
      RETURN                                                            ORE   11
      END                                                               ORE   12-
```

```
      SUBROUTINE QSAME (QM,VV)                                          QSA    1
C     SUBROUTINE QSAME SETS THE CROSS SECTION FOR A GIVEN SPECIES PAIR  QSA    2
C     EQUAL TO THOSE FOR ANOTHER PAIR.                                  QSA    3
      DIMENSION QM(3),VV(5)                                             QSA    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              QSA    5
      I=VV(1)                                                           QSA    6
      J=VV(2)                                                           QSA    7
      C=VV(3)                                                           QSA    8
      QM(1)=C*Q(1,I,J)                                                  QSA    9
      QM(2)=C*Q(2,I,J)*VV(4)                                            QSA   10
      QM(3)=C*Q(3,I,J)*VV(5)                                            QSA   11
      RETURN                                                            QSA   12
      END                                                               QSA   13-
```

```
      SUBROUTINE QTAB (OM,V)                                          QTA    1
C     USE TABULAR DATA FOR CROSS SECTIONS.                            QTA    2
      DIMENSION CM(3),V(3)                                            QTA    3
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                            QTA    4
      A=V(1)                                                          QTA    5
      N1=V(2)                                                         QTA    6
      NL=V(3)                                                         QTA    7
      CALL QINTRP (T,CM,N1,NL)                                        QTA    8
      OM(1)=A*CM(1)                                                   QTA    9
      OM(2)=A*CM(2)                                                   QTA   10
      OM(3)=OM(3)*OM(1)                                               QTA   11
      RETURN                                                          QTA   12
      END                                                             QTA   13-
```

```
      SUBROUTINE Q11 (LQ1,LQ2,VI)                                        Q11    1
C     SUBROUTINE Q11 MULTIPLIES CROSS SECTIONS BY A RAMP FUNCTION OF     Q11    2
C     TEMPERATURE.                                                       Q11    3
      DIMENSION VV(3),VI(2)                                              Q11    4
      COMMON /TRANS1/ T.0(3,20,20),ZM2(20)                               Q11    5
      DATA VV/3*1./                                                      Q11    6
      T0=VI(1)                                                           Q11    7
      T1=VI(2)                                                           Q11    8
      VV(1)=AMAX1(0.,(T-T0)/(T1-T0))                                     Q11    9
      IF (VV(1).LT.1.) CALL Q14 (LQ1,LQ2,VV)                             Q11   10
10    RETURN                                                             Q11   11
      END                                                                Q11   12-
```

```
      SUBROUTINE Q12 (OM,VV)                                              Q12    1
C     SUBROUTINE Q12 COMPUTES CROSS SECTIONS FROM THE GENERALIZED MIXINGQ12    2
C     RULE (EQ. 27).                                                      Q12    3
      DIMENSION OM(3),VV(4)                                               Q12    4
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                                Q12    5
      I=VV(1)                                                             Q12    6
      J=VV(2)                                                             Q12    7
      K=VV(3)                                                             Q12    8
      L=VV(4)                                                             Q12    9
      DO 10 M=1,3                                                         Q12   10
10    OM(M)=.25*(SQRT(Q(M,I,J))+SQRT(Q(M,K,L)))**2                        Q12   11
      RETURN                                                              Q12   12
      END                                                                 Q12   13-
```

```
      SUBROUTINE Q13 (LQ1,LQ2,VV)                                       Q13    1
C     SUBROUTINE Q13 SETS TWO CROSS SECTIONS EQUAL FOR A SPECIES PAIR.  Q13    2
      DIMENSION VV(3)                                                   Q13    3
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              Q13    4
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ        Q13    5
      K1=VV(1)                                                          Q13    6
      K2=VV(2)                                                          Q13    7
      C=VV(3)                                                           Q13    8
      DO 10 LQ=LQ1,LQ2                                                  Q13    9
      I=IQ(LQ)                                                          Q13   10
      J=JQ(LQ)                                                          Q13   11
10    Q(K1,I,J)=C*Q(K2,I,J)                                             Q13   12
      RETURN                                                            Q13   13
      END                                                               Q13   14-
```

```
      SUBROUTINE Q14 (LQ1,LQ2,VV)                                       Q14    1
C     SUBROUTINE Q14 MULTIPLIES CROSS SECTIONS BY A CONSTANT.           Q14    2
      DIMENSION VV(3)                                                   Q14    3
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              Q14    4
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ        Q14    5
      C2=VV(1)*VV(2)                                                    Q14    6
      C3=VV(1)*VV(3)                                                    Q14    7
      DO 10 LQ=LQ1,LQ2                                                  Q14    8
      I=IQ(LQ)                                                          Q14    9
      J=JQ(LQ)                                                          Q14   10
      Q(1,I,J)=VV(1)*Q(1,I,J)                                           Q14   11
      Q(2,I,J)=C2*Q(2,I,J)                                              Q14   12
10    Q(3,I,J)=C3*Q(3,I,J)                                              Q14   13
      RETURN                                                            Q14   14
      END                                                               Q14   15-
```

```
      SUBROUTINE RADIUS (ITYPE,X,R,AG,AGJ,L)                            RAD    1
      DIMENSION YZ(2),NPR(2)                                            RAD    2
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  RAD    3
     1      SW,RO,JDIM,IPOINT                                           RAD    4
      COMMON /BLRAD/ YOZO                                               RAD    5
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         RAD    6
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                 RAD    7
      DATA NPR /2,1/                                                    RAD    8
      GO TO (10,10,40), ITYPE                                           RAD    9
10    CALL GMAR (X,R)                                                   RAD   10
      RATIO=R/RO                                                        RAD   11
      GO TO (20,30,30), ITYPE                                           RAD   12
20    AG=RATIO                                                          RAD   13
      AGJ=1.                                                            RAD   14
      GO TO 50                                                          RAD   15
30    AG=RATIO**2                                                       RAD   16
      AGJ=AG                                                            RAD   17
      GO TO 50                                                          RAD   18
40    CALL GMAR2 (X,YZ(1),YZ(2))                                        RAD   19
      AG=YZ(1)*YZ(2)/YOZO                                               RAD   20
      R=YZ(L)                                                           RAD   21
      M=NPR(L)                                                          RAD   22
      AGJ=YZ(M)**2                                                      RAD   23
50    RETURN                                                            RAD   24
      END                                                               RAD   25-
```

```
      SUBROUTINE READ                                                   REA    1
      LOGICAL SUPGO,READXS,DATAPE,READG,AAMS,SETIGS                     REA    2
      LOGICAL WEDGEM,AXISYM,AXIMOD,NOTRAN                               REA    3
      REAL SHAPD(4,3),SPNAME(30)                                        REA    4
      REAL ACOM(30),ELMENT(10),HP(20)                                   REA    5
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              REA    6
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             REA    7
      COMMON        A, AFNTS,   AFNX, AMACH,     AR,   ARBA,   ARBB, BZERD,   REA    8
     1      C,   CARB,     CH,    CHA,   CLNT,     CM,    CMA,    CRA,   CRP,   REA    9
     2   CRRB,    CRS,   CSTA,     CT,   CTAP,    CTB,    CTC, CTMAX, CTMXX,   REA   10
     3    CTP,   CTPL,    CTT,     CX,    CXB, CXMAX,DATEST,DBTEST, DELT1,   REA   11
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,    ENT,   FLUX, HDELX,    PCT,   REA   12
     5PCTEST,   PRES, PRESA, PRESB,PRESTH,   PRHC,   RHAP,    RHO,   RHOB,   REA   13
     6RHOBAR,   RHOC,   RHOP,   RHPL,   RHTH,ROBARA,ROBARP,   SCPG,    SDT,   REA   14
     7    SEN,   SHPG,     SC,     SL,   SL64,     SM,     SU,    SU2,   SUMG,   REA   15
     8   TEST, TESTB,TPRINT, TSTOP,     UP,     ZP,    ZPA                    REA   16
      COMMON          BE(64),    BET(20),   BLBK(31),    CAI(64),CAPXTH(20),REA   17
     1   CCPJ(20), CEACT(64),    CGI(20),   CGMU(20),    CHI(64),   CHII(20),REA   18
     2CLNIMC(64), CLNPI(64),    CMW(20),   ETAI(64),   ETAJ(20),    GJA(20),REA   19
     3    GJB(20),PERTGJ(20),    PGJ(20),     PI(64), PICHI(64),     QM(20),REA   20
     4     QQ(64),    SAJ(20), SDCHI(64),   SENT(20),    SHJ(20),   SHJA(20),REA   21
     5   SKIL(20),     SS(20),     TB(30),    TFA(20),    TFB(20),    TFC(20),REA   22
     6    TFD(20),    TFE(20),    TFK(20),  THEV(20), XMJAT(20),    XNUI(64) REA   23
      COMMON            BETA(64,20),    ELJ(10,20),   GELJ(10,20),          REA   24
     1 XNUIJ(64,20),XNUIJP(64,20)                                       REA   25
      COMMON      IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC,  ISCP1,   REA   26
     1   ISMC,ISMCNR,    ISR,    ISS, ISSNR,  ISSP1,  ISSP2,  ISSP3,  ISSP4,   REA   27
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,   REA   28
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,    JJK,     LC,     M1,   NFIT,   REA   29
     4    NIT,    NNN,    NNS,    NQS,    NQT, NTEST                     REA   30
      COMMON    IGJ(20),    IGM(20),    ITB(5),    KUR(64,20),   LPIJ(20,10) REA   31
      COMMON ACOM,ELMENT,HP                                             REA   32
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         REA   33
     1      NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                 REA   34
      COMMON /AVG/ WSAVE                                                REA   35
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  REA   36
     1      SN,RO,JDIM,IPCINT                                           REA   37
      COMMON /CHAN/ CP1(5),CP2(5),CP3(5),CP4(5),CP5(5)                  REA   38
      COMMON /COLDSP/ CGMW,QPJ(10),IJCS(10),NCS                         REA   39
      COMMON /EELEM/ EEP1(2),EEP2(2),EEP3(2),EEP4(2),EEP5(2),EEP6(2),   REA   40
     1      EEP7(2),EEP8(2),EEP9(2),EEP10(2)                            REA   41
      COMMON /ELEM/ EP1(2),EP2(2),EP3(2),EP4(2),EP5(2),EP6(2),EP7(2),   REA   42
     1      EP8(2),EP9(2),EP10(2)                                       REA   43
      COMMON /LN/ ISATOM,ISMOL,JATOM,JMOL                               REA   44
      COMMON /MASSFL/ SMASS,CTMXXI,TSTOPI,IS(20)                        REA   45
      COMMON /MIXT/ GP1(124),GP2(124),GP3(124),GP4(124),GP5(124),       REA   46
     1      GP6(124)                                                    REA   47
      COMMON /MODPAR/ XMP1,DXMP,FSTAG,    CATFAC,TMODEL,XMODP1,DXMODP,  REA   48
     1      TPLATE,KDIM                                                 REA   49
      COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,NBL                        REA   50
      COMMON /NEWMP/ FACMP,NMODPT                                       REA   51
      COMMON /NOZZ/ ZP1(64),ZP2(64),ZP3(64),ZP4(64),ZP5(64),ZP6(64),    REA   52
     1      ZP7(64),ZP8(64),ZP9(64),ZP10(64),ZP11(64),ZP12(64),ZP13(64), REA   53
     2      ZP14(64),ZP15(64),ZP16(64),ZP17(64),ZP18(64),ZP19(64),      REA   54
     3      ZP20(64)                                                    REA   55
```

```
      COMMON /POLYAT/ THEVE(4,20)                                          REA   56
      COMMON /RDLIST/ IGAS,IGASO,NOZZO,ICHANO                              REA   57
      COMMON /RDMAIN/ HS,SUPGO,MFITER,NOTRAN                               REA   58
      COMMON /RDMOD/ LEWIS,IAMBIP                                          REA   59
      COMMON /RDOUT/ FLOW,FACNAM,CHANAM,LIMOUT                             REA   60
      COMMON /RDTE/ ISW8B                                                  REA   61
      COMMON /ROWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,        REA   62
     1     NWX,NANGLE,NRADLE,WEDGEM,AXISYM,ISW9B                           REA   63
      COMMON /REAC/ RP1(29),RP2(29),RP3(29),RP4(29),RP5(29),RP6(29),       REA   64
     1     RP7(29),RP8(29),RP9(29),RP10(29),RP11(29),RP12(29),RP13(29),    REA   65
     2     RP14(29),RP15(29),RP16(29),RP17(29),RP18(29),RP19(29),          REA   66
     3     RP20(29),RP21(29),RP22(29),RP23(29),RP24(29),RP25(29),          REA   67
     4     RP26(29),RP27(29),RP28(29),RP29(29),RP30(29),RP31(29),          REA   68
     5     RP32(29),RP33(29),RP34(29),RP35(29),RP36(29),RP37(29),          REA   69
     6     RP38(29),RP39(29),RP40(29),RP41(29),RP42(29),RP43(29),          REA   70
     7     RP44(29),RP45(29),RP46(29),RP47(29),RP48(29),RP49(29),          REA   71
     8     RP50(29),RP51(29),RP52(29),RP53(29),RP54(29),RP55(29),          REA   72
     9     RP56(29),RP57(29),RP58(29),RP59(29),RP60(29),RP61(29),          REA   73
     $     RP62(29),RP63(29),RP64(29),RP65(29),RP66(29),RP67(29),          REA   74
     $     RP68(29),RP69(29),RP70(29),RP71(29),RP72(29),RP73(29),          REA   75
     $     RP74(29),RP75(29),RP76(29),RP77(29),RP78(29),RP79(29),          REA   76
     $     RP80(29),RP81(29),RP82(29),RP83(29),RP84(29),RP85(29),          REA   77
     $     RP86(29),RP87(29),RP88(29),RP89(29),RP90(29),RP91(29),          REA   78
     $     RP92(29)                                                        REA   79
      COMMON /READAT/ PRESAI,DELTII,TPRNTI,DELTXI,CXMAXI,                  REA   80
     1     CTAPI,XZEROI,PARAMI(3,12,2),BZEROI,                             REA   81
     2     HSTAG,READXS,READS,AAMS,AXIMOD,                                 REA   82
     3     ICASE,NEELS,ISW7B,INT,ICHAN,NQSI,INEQVI,NRECO,JCS(10),          REA   83
     4     ISCI,ISSI,ISRI,ICI,IE(10),IR(64),NSECTS(2,2),IEEP(10)           REA   84
      COMMON /RENE/ GAMIN,HTEST,TETEST,QTEST,DCHLL,DCHRAT,CCHI,TTEST,      REA   85
     1     GTEST                                                           REA   86
      COMMON /SPEC/ SP1(43),SP2(43),SP3(43),SP4(43),SP5(43),SP6(43),       REA   87
     1     SP7(43),SP8(43),SP9(43),SP10(43),SP11(43),SP12(43),SP13(43),    REA   88
     2     SP14(43),SP15(43),SP16(43),SP17(43),SP18(43),SP19(43),          REA   89
     3     SP20(43),SP21(43),SP22(43),SP23(43),SP24(43),SP25(43),          REA   90
     4     SP26(43),SP27(43),SP28(43),SP29(43),SP30(43)                    REA   91
      COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAPE        REA   92
      COMMON /THRT/ RSA                                                    REA   93
      COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE                                  REA   94
      COMMON /TNE/ TN1(186),TN2(186)                                       REA   95
      COMMON /TNEQ/ TLIST(30),PCM(30)                                      REA   96
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),          REA   97
     1     KTR(25),ICH(20),IPA(25)                                         REA   98
      COMMON /TRANI/  I1 (5), I2 (5), I3 (5), I4 (5), I5 (5),              REA   99
     1                I6 (5), I7 (5), I8 (5), I9 (5), I10(5),              REA  100
     2                I11(5), I12(5), I13(5), I14(5), I15(5),              REA  101
     3                I16(5), I17(5), I18(5), I19(5), I20(5),              REA  102
     4                I21(5), I22(5), I23(5), I24(5), I25(5),              REA  103
     5                I26(5), I27(5), I28(5), I29(5), I30(5),              REA  104
     6                I31(5), I32(5), I33(5), I34(5), I35(5),              REA  105
     7                I36(5), I37(5), I38(5), I39(5), I40(5),              REA  106
     8                I41(5), I42(5), I43(5), I44(5), I45(5),              REA  107
     9                I46(5), I47(5), I48(5), I49(5), I50(5),              REA  108
     $                I51(5), I52(5), I53(5), I54(5), I55(5),              REA  109
     $                I56(5), I57(5), I58(5), I59(5), I60(5),              REA  110
```

```
$                I61(5), I62(5), I63(5), I64(5), I65(5),                REA 111
$                I66(5), I67(5), I68(5), I69(5), I70(5),                REA 112
$                I71(5), I72(5), I73(5), I74(5), I75(5),                REA 113
$                I76(5), I77(5), I78(5), I79(5), I80(5),                REA 114
$                I81(5), I82(5), I83(5), I84(5), I85(5),                REA 115
$                I86(5), I87(5), I88(5), I89(5), I90(5),                REA 116
$                I91(5), I92(5), I93(5), I94(5), I95(5),                REA 117
$                I96(5), I97(5), I98(5), I99(5), I100(5)                REA 118
 COMMON /TRANJ/  J1 (5), J2 (5), J3 (5), J4 (5), J5 (5),                REA 119
1                J6 (5), J7 (5), J8 (5), J9 (5), J10(5),                REA 120
2                J11(5), J12(5), J13(5), J14(5), J15(5),                REA 121
3                J16(5), J17(5), J18(5), J19(5), J20(5),                REA 122
4                J21(5), J22(5), J23(5), J24(5), J25(5),                REA 123
5                J26(5), J27(5), J28(5), J29(5), J30(5),                REA 124
6                J31(5), J32(5),  33(5), J34(5), J35(5),                REA 125
7                J36(5), J37(5), J38(5), J39(5), J40(5),                REA 126
8                J41(5), J42(5), J43(5), J44(5), J45(5),                REA 127
9                J46(5), J47(5), J48(5), J49(5), J50(5),                REA 128
$                J51(5), J52(5), J53(5), J54(5), J55(5),                REA 129
$                J56(5), J57(5), J58(5), J59(5), J60(5),                REA 130
$                J61(5), J62(5), J63(5), J64(5), J65(5),                REA 131
$                J66(5), J67(5), J68(5), J69(5), J70(5),                REA 132
$                J71(5), J72(5), J73(5), J74(5), J75(5),                REA 133
$                J76(5), J77(5), J78(5), J79(5), J80(5),                REA 134
$                J81(5), J82(5), J83(5), J84(5), J85(5),                REA 135
$                J86(5), J87(5), J88(5), J89(5), J90(5),                REA 136
$                J91(5), J92(5), J93(5), J94(5), J95(5),                REA 137
$                J96(5), J97(5), J98(5), J99(5), J100(5)                REA 138
 COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,IIIII(50)             REA 139
 COMMON /TRANS4/ TL(1000),OMEGA1(1000),ASTAR(1000),BSTAR(1000)          REA 140
 COMMON /TRANV/  V1 (5), V2 (5), V3 (5), V4 (5), V5 (5),                REA 141
1                V6 (5), V7 (5), V8 (5), V9 (5), V10(5),                REA 142
2                V11(5), V12(5), V13(5), V14(5), V15(5),                REA 143
3                V16(5), V17(5), V18(5), V19(5), V20(5),                REA 144
4                V21(5), V22(5), V23(5), V24(5), V25(5),                REA 145
5                V26(5), V27(5), V28(5), V29(5), V30(5),                REA 146
6                V31(5), V32(5), V33(5), V34(5), V35(5),                REA 147
7                V36(5), V37(5), V38(5), V39(5), V40(5),                REA 148
8                V41(5), V42(5), V43(5), V44(5), V45(5),                REA 149
9                V46(5), V47(5), V48(5), V49(5), V50(5),                REA 150
$                V51(5), V52(5), V53(5), V54(5), V55(5),                REA 151
$                V56(5), V57(5), V 8(5), V59(5), V60(5),                REA 152
$                V61(5), V62(5), V63(5), V64(5), V65(5),                REA 153
$                V66(5), V67(5), V68(5), V69(5), V70(5),                REA 154
$                V71(5), V72(5), V73(5), V74(5), V75(5),                REA 155
$                V76(5), V77(5), V78(5), V79(5), V80(5),                REA 156
$                V81(5), V82(5), V83(5), V84(5), V85(5),                REA 157
$                V86(5), V87(5), V88(5), V89(5), V90(5),                REA 158
$                V91(5), V92(5), V93(5), V94(5), V95(5),                REA 159
$                V96(5), V97(5), V98(5), V99(5), V100(5)                REA 160
 DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBREA 161
1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                    REA 162
 DIMENSION ASYM(80)                                                    REA 163
 DIMENSION CP(5,5)                                                     REA 164
 DIMENSION DUM(4)                                                      REA 165
```

```
      DIMENSION EEPRP(2,10)                                                 REA 166
      DIMENSION ELNAME(3),NATOM(3)                                          REA 167
      DIMENSION EPRP(2,10),SPRP(43,30),RPRP(29,92),GPRP(124,6),             REA 168
     1     ZPRP(64,10)                                                      REA 169
      DIMENSION NSECTD(2)                                                   REA 170
      DIMENSION SY1(3),SY2(3)                                               REA 171
      DIMENSION TNEP(186,2)                                                 REA 172
      DIMENSION WORD(3,2)                                                   REA 173
      DIMENSION WORD2(2)                                                    REA 174
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),                  REA 175
     1     (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),            REA 176
     2     (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),                 REA 177
     3     (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                            REA 178
      EQUIVALENCE (CP1(1),CP(1,1))                                          REA 179
      EQUIVALENCE (DUM(1),SUMGH)                                            REA 180
      EQUIVALENCE (EEP1(1),EEPRP(1,1))                                      REA 181
      EQUIVALENCE (EP1(1),EPRP(1,1))                                        REA 182
      EQUIVALENCE (GP1(1),GPRP(1,1))                                        REA 183
      EQUIVALENCE (NOZZLE,NPROFL(1))                                        REA 184
      EQUIVALENCE (RP1(1),RPRP(1,1))                                        REA 185
      EQUIVALENCE (SP1(1),SPRP(1,1))                                        REA 186
      EQUIVALENCE(TNEP(1,1),TN1(1))                                         REA 187
      EQUIVALENCE (ZP1(1),ZPRP(1,1))                                        REA 188
      DATA ASYM /1HH,2HHE,2HLI,2HBE,1HB,1HC,1HN,1HO,1HF,2HNE,2HNA,2HMG,     REA 189
     1     2HAL,2HSI,1HP,1HS,2HCL,2HAR,1HK,2HCA,2HSC,2HTI,1HV,2HCR,2HMN,    REA 190
     2     2HFE,2HCO,2HNI,2HCU,2HZN,2HGA,2HGE,2HAS,2HSE,2HBR,2HKR,2HRB,     REA 191
     3     2HSR,1HY,2HZR,2HNB,2HMO,2HTC,2HRU,2HRH,2HPD,2HAG,2HCD,2HIN,      REA 192
     4     2HSN,2HSB,2HTE,1HI ,2HXE,2HCS,2HBA,2HLA,2HCE,2HPR,2HND,2HPM,     REA 193
     5     2HSM,2HEU,2HGD,2HTB,2HDY,2HHO,2HER,2HTM,2HYB,2HLU,2HHF,2HTA,     REA 194
     6     1HW,2HRE,2HOS,2HIR,2HPT,2HAU,2HHG/                               REA 195
      DATA IUPDI /1/                                                        REA 196
      DATA PLUS /1H+/, ONE /1./                                             REA 197
      DATA SHAPD /'STRA','IGHT',' LIN','E','CIRC','LE B','OTTO','M',       REA 198
     1     'CIRC','LE T','OP',' '/                                          REA 199
      DATA SPNAME /'SP1','SP2','SP3','SP4','SP5','SP6','SP7','SP8',        REA 200
     1     'SP9','SP10','SP11','SP12','SP13','SP14','SP15','SP16',          REA 201
     2     'SP17','SP18','SP19','SP20','SP21','SP22','SP23','SP24',         REA 202
     3     'SP25','SP26','SP27','SP28','SP29','SP30'/                       REA 203
      DATA SY1,SY2 /3*1H(,3*1H)/                                            REA 204
      DATA WORD /6HTWO-DI,6HMENSIO,3HNAL,6H  AXIS,6HYMMETR,2HIC/            REA 205
      DATA WORD2 /3HGAP,4HDIAM/                                             REA 206
      NAMELIST /INPUT/                                                      REA 207
C     ******GROUP 1 - GENERAL CONTROL VARIABLES                             REA 208
     1     ISW1A,ISW2A,ISW3A,ISW4A,ISW6A,ISW1B,ISW3B,TWALL,NOTRAN,           REA 209
     1     TSTOPI,CXMAXI,READG,READXS,                                      REA 210
C     ******GROUP 2 - OUTPUT CONTROLS                                       REA 211
     2     ISW6B,ISW7B,TPRNTI,DATAPE,NRECO,IRUN,                            REA 212
C     ******GROUP 3 - RESERVOIR CONDITIONS                                  REA 213
     3     ISW2B,PRESAI,FLOW,CTAPI,HSTAG,MFITER,                            REA 214
C     ******GROUP 4 - GEOMETRY                                              REA 215
     4     NOZZLE,ICHAN,JDIM,NPRFLS,NPROFL,NBL,DIAM,NSECTS,ISHAPE,          REA 216
     4     PARAMI,ATPI,XZEROI,                                              REA 217
C     ******GROUP 5 - GAS MODEL                                             REA 218
     5     IGAS,AAMS,NCS,JCS,QPJ,ISCI,ISSI,ISRI,ICI,IE,IS,IR,ISATOM,        REA 219
     5     ISMOL,CTMXXI,BZEROI,INEQVI,                                      REA 220
```

```
C      ******GROUP 6 - TEST MODEL                                                REA 221
     6      XMODP1,NMODPT,TSDIAM,AXIMOD,KDIM,  STAG,CATFAC,LEWIS,TMODEL,        REA 222
     6      TPLATE,                                                             REA 223
C      ******GROUP 7 - WEDGE MODELS                                              REA 224
     7      NANGLE,ANGLE,NRADLE,RADLE,WX1,DWX,NWX,WXI,TWEDGE,WK,ISW9B,          REA 225
C      ******GROUP 8 - CONTROLS FOR THE FLOW SOLUTIONS                           REA 226
     8      WSAVE,DELT1I,DELTXI,CCHI,NQSI,TTEST,GTEST,HTEST,TETEST,             REA 227
     8      QTEST,GAMIN,DCHLL,DCHRAT,                                           REA 228
C      ******GROUP 9 - ELECTRONIC NONEQUILIBRIUM                                 REA 229
     9      INT,KTF,KTR,ITR,EPAR,BPAR,TLIST,POM,                                REA 230
C      ******GROUP 10 - CONTROLS FOR DIAGNOSTIC DUMPS                            REA 231
     $      ISW5A,ISW4B,ISW5B,ISW8B                                             REA 232
      NAMELIST /EINPUT/      EEP1,EEP2,EEP3,EEP4,EEP5,EEP6,EEP7,EEP8,          REA 233
     1      EEP9,EEP10,NEELS,IEFP,SP1,SP2,SP3,SP4,SP5,SP6,SP7,SP8,SP9,          REA 234
     2      SP10,SP11,SP12,SP13,SP14,SP15,SP16,SP17,SP18,SP19,SP20,SP21,        REA 235
     3      SP22,SP23,SP24,SP25,SP26,SP27,SP28,SP29,SP30,RP1,RP2,RP3,RP4,       REA 236
     4 RP5,RP6,RP7,RP8,RP9,RP10,RP11,RP12,RP13,RP14,RP15,RP16,RP17,RP18,       REA 237
     5      RP19,RP20,RP21,RP22,RP23,RP24,RP25,RP26,RP27,RP28,RP29,RP30,        REA 238
     6      RP31,RP32,RP33,RP34,RP35,RP36,RP37,RP38,RP39,RP40,RP41,RP42,        REA 239
     7      RP43,RP44,RP45,RP46,RP47,RP48,RP49,RP50,RP51,RP52,RP53,RP54,        REA 240
     8      RP55,RP56,RP57,RP58,RP59,RP60,RP61,RP62,RP63,RP64,RP65,RP66,        REA 241
     9      RP67,RP68,RP69,RP70,RP71,RP72,RP73,RP74,RP75,RP76,RP77,RP78,        REA 242
     $      RP79,RP80,RP81,RP82,RP83,RP84,RP85,RP86,RP87,RP88,RP89,RP90,        REA 243
     $      RP91,RP92                                                           REA 243A
      NAMELIST /TINPUT/KKQ,NNQ,NNKQ,ISEQ,V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,       REA 244
     1V11,V12,V13,V14,V15,V16,V17,V18,V19,V20,V21,V22,V23,V24,V25,V26,          REA 245
     2V27,V28,V29,V30,V31,V32,V33,V34,V35,V36,V37,V38,V39,V40,V41,V42,          REA 246
     3V43,V44,V45,V46,V47,V48,V49,V50,V51,V52,V53,V54,V55,V56,V57,V58,          REA 247
     4V59,V60,V61,V62,V63,V64,V65,V66,V67,V68,V69,V70,V71,V72,V73,V74,          REA 248
     5V75,V76,V77,V78,V79,V80,V81,V82,V83,V84,V85,V86,V87,V88,V89,V90,          REA 249
     6V91,V92,V93,V94,V95,V96,V97,V98,V99,V100,I1,I2,I3,I4,I5,I6,I7,I8,        REA 250
     7I9,I10,I11,I12,I13,I14,I15,I16,I17,I18,I19,I20,I21,I22,I23,I24,          REA 251
     8I25,I26,I27,I28,I29,I30,I31,I32,I33,I34,I35,I36,I37,I38,I39,I40,         REA 252
     9I41,I42,I43,I44,I45,I46,I47,I48,I49,I50,I51,I52,I53,I54,I55,I56,         REA 253
     $I57,I58,I59,I60,I61,I62,I63,I64,I65,I66,I67,I68,I69,I70,I71,I72,         REA 254
     $I72,I73,I74,I75,I76,I77,I78,I79,I80,I81,I82,I83,I84,I85,I86,I87,         REA 255
     $I88,I89,I90,I91,I92,I93,I94,I95,I96,I97,I98,I99,I100,J1,J2,J3,J4,        REA 256
     $J5,J6,J7,J8,J9,J10,J11,J12,J13,J14,J15,J16,J17,J18,J19,J20,J21,          REA 257
     $J22,J23,J24,J25,J26,J27,J28,J29,J30,J31,J32,J33,J34,J35,J36,J37,         REA 258
     $J38,J39,J40,J41,J42,J43,J44,J45,J46,J47,J48,J49,J50,J51,J52,J53,         REA 259
     $J54,J55,J56,J57,J58,J59,J60,J61,J62,J63,J64,J65,J66,J67,J68,J69,         REA 260
     $J70,J71,J72,J73,J74,J75,J76,J77,J78,J79,J80,J81,J82,J83,J84,J85,         REA 261
     $J86,J87,J88,J89,J90,J91,J92,J93,J94,J95,J96,J97,J98,J99,J100,            REA 262
     $      TL,OMEGA1,ASTAR,BSTAR                                               REA 263
      READ (5,960) (ACOM(I),I=1,20)                                             REA 264
C     FIRST 4 CHARACTERS ON THIS CARD ARE USED AS A FACILITY NAME IF            REA 265
C           NOZZLE=0 (NONSTANDARD NOZZLE)                                       REA 266
      READ (5,INPUT)                                                            REA 267
      WRITE (6,970)                                                             REA 268
      WRITE (6,INPUT)                                                           REA 269
      IF (.NOT.READG) GO TO 20                                                  REA 270
      READ (5,EINPUT)                                                           REA 271
      WRITE (6,EINPUT)                                                          REA 272
      DO 10 I=1,20                                                              REA 273
      IF (SPRP(1,I).NE.0.) GO TO 10                                             REA 274
```

```
      SPRP(1,I)=SPNAME(I)                                                REA 275
10    CONTINUE                                                           REA 276
20    CONTINUE                                                           REA 277
      IF (.NOT.READ 3) GO TO 30                                          REA 278
      READ (5,TINPUT)                                                    REA 279
      WRITE (6,TINPUT)                                                   REA 280
30    CONTINUE                                                           REA 281
      ICASE=ICASE+1                                                      REA 282
      WRITE (6,980) IRUN,ICASE,(ACOM(I),I=1,20)                          REA 283
      SETIGS=AAMS.AND.(IGAS.EQ.1.OR.IGAS.EQ.2)                           REA 284
      DO 40 I=1,4                                                        REA 285
40    DUM(I)=0.                                                          REA 286
      IF (ICHAN.GT.0) NPRFLS=2                                           REA 287
      IF (NPRFLS.EQ.1.OR.NPRFLS.EQ.2) GO TO 50                           REA 288
      WRITE (6,990) NPRFLS                                               REA 289
      STOP                                                               REA 290
50    DO 60 I=1,NPRFLS                                                   REA 291
      IF (ICHAN.GT.0) NPROFL(I)=CP(I,ICHAN)                              REA 292
      IPRFL=NPROFL(I)                                                    REA 293
      IF (IPRFL.GT.0) DIAM(I)=ZPRP(1,IPRFL)/1.27                         REA 294
60    CONTINUE                                                           REA 295
      AXISYM=AXIMOD                                                      REA 296
      IF (NPRFLS.EQ.2) AXISYM=.FALSE.                                    REA 297
      IF (ISW2B) 70,80,170                                               REA 298
70    HS=HSTAG/1.8                                                       REA 299
      WRITE (6,1000) HSTAG,FLOW                                          REA 300
      GO TO 90                                                           REA 301
80    WRITE (6,1010) PRESAI,FLOW                                         REA 302
90    IF (NPRFLS.EQ.2) GO TO 110                                         REA 303
      IF (JDIM.NE.0) GO TO 100                                           REA 304
      SMASS=70.3069*FLOW/DIAM(1)                                         REA 305
      GO TO 120                                                          REA 306
100   SMASS=89.5173*FLOW/(DIAM(1))**2                                    REA 307
      GO TO 120                                                          REA 308
110   SMASS=70.3069*FLOW/DIAM(1)/DIAM(2)                                 REA 309
120   IF (.NOT.SETIGS) GO TO 180                                         REA 310
      IF (ISW2B.EQ.0) GO TO 130                                          REA 311
      HWINOV=HSTAG                                                       REA 312
      GO TO 14                                                           REA 313
130   HWINOV=(..8.7*PRESAI/SMASS)**2.52                                  REA 314
140   IF (HWINOV-8000.) 150,150,160                                      REA 315
150   IGAS=2                                                             REA 316
      GO TO 180                                                          REA 317
160   IGAS=1                                                             REA 318
      GO TO 180                                                          REA 319
170   WRITE (6,1020) PRESAI,CTAPI                                        REA 320
      IF (.NOT.SETIGS) GO TO 180                                         REA 321
      IF (CTAPI-6000.) 150,150,160                                       REA 322
180   IUPD=IUPDI                                                         REA 323
      NQS=NQSI                                                           REA 324
      IF (FSTAG.NE.0.) FSTAG=SIGN(ONE,FSTAG)                             REA 325
      RSA=1.                                                             REA 326
      IF (NPRFLS.EQ.2) GO TO 190                                         REA 327
      NBL=1                                                              REA 328
      GO TO 200                                                          REA 329
```

```
190      IF (ICHAN.GT.0) NBL=CP(4,ICHAN)                                    REA 330
200      IF (ISW3B.NE.0) ISW3B=1                                            REA 331
         LIMOUT=ISW7B*ISW3B                                                 REA 332
         INEQV=INEQVI                                                       REA 333
         CTAP=CTAPI                                                         REA 334
         PRESA=PRESAI                                                       REA 335
         CTMXX=CTMXXI                                                       REA 336
         CXMAX=CXMAXI*2.5                                                   REA 337
         MBL=1                                                              REA 338
         IF (NPRFLS.EQ.1) GO TO 210                                         REA 339
         IF (NBL.EQ.1) MBL=2                                                REA 340
210      NTS=0                                                              REA 341
         DO 230 I=1,20                                                      REA 342
         IF (TSDIAM(I).LE.DIAM(MBL)) TSDIAM(I)=1.E30                        REA 343
         IF (TSDIAM(I).GT.9.E19) GO TO 230                                  REA 344
         NTS=NTS+1                                                          REA 345
         IF (NPRFLS.EQ.2) GO TO 220                                         REA 346
         TSAR(NTS)=(TSDIAM(I)/DIAM(1))**(JDI'1+1)                           REA 347
         GO TO 230                                                          REA 348
220      TSAR(NTS)=1.27*TSDIAM(I)                                           REA 349
230      CONTINUE                                                           REA 350
         IF (NANGLE.EQ.C.OR.NRADLE.EQ.0) GO TO 240                          REA 351
         IF (NWX.EQ.0.AND.WXI(1).GT.1.E19) GO TO 240                        REA 352
         IF (NPRFLS.EQ.2) GO TO 240                                         REA 353
         WEDGEM=.TRUE.                                                      REA 354
         GO TO 250                                                          REA 355
240      WEDGEM=.FALSE.                                                     REA 356
250      IF (NOZZLE.GT.0) GO TO 260                                         REA 357
         XZERO=2.54*XZEROI                                                  REA 358
         GO TO 270                                                          REA 359
260      XZERO=ZPRP(2,NOZZLE)                                               REA 360
C        IN THE CASE OF A CHANNEL, THE CODE USES THE XZERO FOR THE FIRST    REA 361
C        PROFILE                                                            REA 362
270      SL=1.                                                              REA 363
         BZERO=BZEROI                                                       REA 364
         TSTOP=TSTOPI                                                       REA 365
         TPRINT=TPRNTI                                                      REA 366
         DELTAX=DELTXI                                                      REA 367
         DELT1=DELT1I                                                       REA 368
         DO 320 J=1,NPRFLS                                                  REA 369
         IPRFL=NPROFL(J)                                                    REA 370
         IF (IPRFL.EQ.C) GO TO 300                                          REA 371
         RTHCM(J)=ZPRP(1,IPRFL)                                             REA 372
         NSECTU(J)=ZPRP(3,IPRFL)                                            REA 373
         NSECTD(J)=ZPRP(4,IPRFL)                                            REA 374
         NSECT(J)=NSECTU(J)+NSECTD(J)                                       REA 375
         IF (J.EQ.1) FACNAM=ZPRP(64,IPRFL)                                  REA 376
         NSECTJ=NSECT(J)                                                    REA 377
         DO 290 I=1,NSECTJ                                                  REA 378
         ISHAPE(I,J)=ZPRP(I+4,IPRFL)                                        REA 379
         IF (I.LT.NSECTJ) ATPI(I,J)=ZPRP(I+16,IPRFL)                        REA 380
         DO 280 L=1,3                                                       REA 381
         LL=3*I+L+24                                                        REA 382
280      PARAM(L,I,J)=ZPRP(LL,IPRFL)                                        REA 383
290      CONTINUE                                                           REA 384
```

```
      GO TO 320                                                          REA 385
  30  NSECTU(J)=NSECTS(1,J)                                              REA 386
      NSECTD(J)=NSECTS(2,J)                                              REA 387
      NSECT(J)=NSECTU(J)+NSECTD(J)                                       REA 388
      IF (J.EQ.1) FACNAM=ACOM(1)                                         REA 389
      RTHCM(J)=1.27*DIAM(J)                                              REA 390
      NSECTJ=NSECT(J)                                                    REA 391
      DO 310 I=1,NSECTJ                                                  REA 392
      DO 310 L=1,3                                                       REA 393
 310  PARAM(L,I,J)=PARAMI(L,I,J)                                         REA 394
 320  CONTINUE                                                           REA 395
      IF (ICHAN.EQ.0) GO TO 330                                          REA 396
      CHANAM=CP(3,ICHAN)                                                 REA 397
      FACNAM=CP(5,ICHAN)                                                 REA 398
      GO TO 340                                                          REA 399
 330  CHANAM=ACOM(2)                                                     REA 400
 340  CONTINUE                                                           REA 401
      IF (NPRFLS.EQ.1) GO TO 360                                         REA 402
      IF (ICHAN.NE.0) GO TO 350                                          REA 403
      WRITE (6,1030) CHANAM,FACNAM,DIAM(1),DIAM(2)                       REA 404
      GO TO 390                                                          REA 405
 350  WRITE (6,1040) CHANAM,ICHAN,FACNAM,DIAM(1),DIAM(2)                 REA 406
      GO TO 390                                                          REA 407
 360  JDP1=JDIM+1                                                        REA 408
      IF (NOZZLE.GT.0) GO TO 370                                         REA 409
      WRITE (6,1050) (WORD(I,JDP1),I=1,3),DIAM(1),WORD2(JDP1),FACNAM     REA 410
      GO TO 380                                                          REA 411
 370  WRITE (6,1060) (WORD(I,JDP1),I=1,3),NOZZLE,DIAM(1),WORD2(JDP1),FACREA 412
     1NAM                                                                REA 413
 380  IF (NOZZLE.EQ.NOZZO.AND.NOZZLE.NE.0) GO TO 440                     REA 414
      GO TO 400                                                          REA 415
 390  IF (ICHAN.EQ.ICHANO.AND.ICHAN.NE.0) GO TO 440                      REA 416
 400  DO 430 J=1,NPRFLS                                                  REA 417
      IF (NPRFLS.EQ.1) GO TO 410                                         REA 418
      WRITE (6,1070) J                                                   REA 419
 410  WRITE (6,1080) RTHCM(J),XZERO,NSECT(J),NSECTU(J)                   REA 420
      WRITE (6,1090)                                                     REA 421
      NSM1=NSECT(J)-1                                                    REA 422
      DO 420 K=1,NSM1                                                    REA 423
      ISH=ISHAPE(K,J)                                                    REA 424
 420  WRITE (6,1100) K,ISH,(SHAPD(I,ISH),I=1,4),ATPI(K,J),(PARAM(L,K,J),REA 425
     1L=1,3)                                                             REA 426
      NSECTJ=NSECT(J)                                                    REA 427
      K=NSECTJ                                                           REA 428
      ISH=ISHAPE(NSECTJ,J)                                               REA 429
      WRITE (6,1110) NSECTJ,ISH,(SHAPD(I,ISH),I=1,4),(PARAM(L,K,J),L=1,3REA 430
     1)                                                                  REA 431
 430  CONTINUE                                                           REA 432
 440  CONTINUE                                                           REA 433
      IF (NEELS.EQ.0) GO TO 460                                          REA 434
      DO 450 I=1,NEELS                                                   REA 435
      IZ=EEPRP(1,I)+0.1                                                  REA 436
      II=IEEP(I)                                                         REA 437
      EPRP(1,II)=ASYM(IZ)                                                REA 438
 450  EPRP(2,II)=EEPRP(2,I)                                              REA 439
```

```
      IGS=IGAS                                                          REA 440
      IF (IGAS.NE.0) GO TO 470                                          REA 441
      WRITE (6,1120)                                                    REA 442
      ISC=ISCI                                                          REA 443
      ISS=ISSI                                                          REA 444
      ISR=ISRI                                                          REA 445
      IC=ICI                                                            REA 446
      GO TO 520                                                         REA 447
  470 IGAS=IABS(IGAS)                                                   REA 448
      WRITE (6,1130) IGAS,GPRP(1,IGAS)                                  REA 449
C     SET UP DESCRIPTION OF GAS IN TERMS OF ELEMENTS, SPECIES, AND      REA 450
C           REACTIONS                                                   REA 451
      ISC=GPRP(2,IGAS)                                                  REA 452
      ISS=GPRP(3,IGAS)                                                  REA 453
      ISR=GPRP(4,IGAS)                                                  REA 454
      IC=GPRP(5,IGAS)                                                   REA 455
      DO 480 I=1,ISC                                                    REA 456
  480 IE(I)=GPRP(I+5,IGAS)                                              REA 457
      NCS=GPRP(120,IGAS)                                                REA 458
      DO 490 I=1,NCS                                                    REA 459
      IF (IGS.LT.0) GO TO 490                                           REA 460
      OPJ(I)=GPRP(I+15,IGAS)                                            REA 461
  490 JCS(I)=GPRP(I+109,IGAS)                                           REA 462
      DO 500 I=1,ISS                                                    REA 463
  500 IS(I)=GPRP(I+25,IGAS)                                             REA 464
      DO 510 I=1,ISR                                                    REA 465
  510 IR(I)=GPRP(I+45,IGAS)                                             REA 466
      ISATOM=GPRP(121,IGAS)                                             REA 467
      ISMOL=GPRP(122,IGAS)                                              REA 468
      INT=GPRP(123,IGAS)                                                REA 469
      LEWIS=GPRP(124,IGAS)                                              REA 470
  520 IF (INT.EQ.0) GO TO 530                                           REA 471
      NT=2                                                              REA 472
      GO TO 540                                                         REA 473
  530 NT=1                                                              REA 474
  540 CONTINUE                                                          REA 475
C     LOOK UP ELEMENTS                                                  REA 476
      DO 550 I=1,ISC                                                    REA 477
      II=IE(I)                                                          REA 478
      ELMENT(I)=EPRP(1,II)                                              REA 479
  550 CMW(I)=EPRP(2,II)                                                 REA 480
C     LOOK UP SPECIES                                                   REA 481
      IAMBIP=1                                                          REA 482
      NFIT=0                                                            REA 483
      DO 680 I=1,ISS                                                    REA 484
      II=IS(I)                                                          REA 485
      DO 560 K=1,NCS                                                    REA 486
      IF (II.EQ.JCS(K)) GO TO 570                                       REA 487
  560 CONTINUE                                                          REA 488
      GO TO 580                                                         REA 489
  570 IJCS(K)=I                                                         REA 490
  580 CONTINUE                                                          REA 491
      IF (II.EQ.ISATOM) JATOM=I                                         REA 492
      IF (II.EQ.ISMOL) JMOL=I                                           REA 493
      HP(I)=SPRP(1,II)                                                  REA 494
```

```
      NELS=SPRP(2,II)+0.1                                                REA 495
      DO 630 K=1,ISC                                                     REA 496
      DO 590 J=1,NELS                                                    REA 497
      IES=SPRP(J+2,II)+0.1                                               REA 498
      IF (IE(K).EQ.IES) GO TO 600                                        REA 499
  590 CONTINUE                                                           REA 500
      LPIJ(I,K)=0                                                        REA 501
      GO TO 630                                                          REA 502
  600 AL=SPRP(J+5,II)                                                    REA 503
      LP=AL                                                              REA 504
      DAL=AL-LP                                                          REA 505
      IF (ABS(DAL).LE.0.5) GO TO 620                                     REA 506
      IF (DAL.GT.0.) GO TO 610                                           REA 507
      LP=LP-1                                                            REA 508
      GO TO 620                                                          REA 509
  610 LP=LP+1                                                            REA 510
  620 LPIJ(I,K)=LP                                                       REA 511
      IF (II.EQ.ISATOM.AND.IES.EQ.1) IAMBIP=2                            REA 512
  630 CONTINUE                                                           REA 513
      TFA(I)=SPRP(9,II)                                                  REA 514
      TFB(I)=SPRP(10,II)                                                 REA 515
      TFC(I)=SPRP(11,II)                                                 REA 516
      TFD(I)=SPRP(12,II)                                                 REA 517
      TFE(I)=SPRP(13,II)                                                 REA 518
      TFK(I)=SPRP(14,II)                                                 REA 519
      SHJAP(I)=SPRP(15,II)                                               REA 520
      ETAJ(I)=SPRP(16,II)                                                REA 521
      SBJ(I)=SPRP(17,II)                                                 REA 522
      THEVP(I)=SPRP(18,II)                                               REA 523
      IF (ETAJ(I).LT.2.9) GO TO 650                                      REA 524
      DO 640 K=1,3                                                       REA 525
  640 THEVE(K+1,I)=SPRP(K+40,II)                                         REA 526
  650 IGM(I)=SPRP(19,II)+0.1                                             REA 527
      IGJ(I)=SPRP(20,II)+0.1                                             REA 528
      IF (IGJ(I).NE.0) NFIT=1                                            REA 529
      IF (ETAJ(I).NE.0..OR.IGJ(I).NE.0) GO TO 660                        REA 530
      WRITE (6,1190) I,II                                                REA 531
      STOP                                                               REA 532
  660 IGMV=IGM(I)                                                        REA 533
      DO 670 L=1,IGMV                                                    REA 534
      GELJ(L,I)=SPRP(L+20,II)                                            REA 535
  670 ELJ(L,I)=SPRP(L+30,II)                                             REA 536
  680 CONTINUE                                                           REA 537
      DO 700 I=1,ISC                                                     REA 538
      CAPQ(I)=0.                                                         REA 539
      DO 700 J=1,NCS                                                     REA 540
      JJ=JCS(J)                                                          REA 541
      NELS=SPRP(2,JJ)+0 1                                                REA 542
      DO 690 K=1,NELS                                                    REA 543
      KK=SPRP(K+2,JJ)+0.1                                                REA 544
      IF (KK.NE.IE(I)) GO TO 690                                         REA 545
      CAPQ(I)=CAPQ(I)+QPJ(J)*SPRP(K+5,JJ)                                REA 546
  690 CONTINUE                                                           REA 547
  700 CONTINUE                                                           REA 548
C     LOOK UP REACTIONS                                                  REA 549
```

```
      DO 780 I=1,ISR                                                    REA 550
      II=IR(I)                                                          REA 551
      CAI(I)=RPRP(1,II)                                                 REA 552
      ETAI(I)=RPRP(2,II)                                                REA 553
      CEACT(I)=RPRP(3,II)                                               REA 554
      QQ(I)=RPRP(4,II)                                                  REA 555
      NRSS=RPRP(5,II)+0.1                                               REA 556
      NPSS=RPRP(6,II)+0.1                                               REA 557
      DO 770 J=1,ISS                                                    REA 558
      DO 710 K=1,NRSS                                                   REA 559
      JRS=RPRP(K+6,II)+0.1                                              REA 560
      IF (JRS.NE.IS(J)) GO TO 710                                       REA 561
      XNUIJ(I,J)=RPRP(K+12,II)                                          REA 562
      GO TO 720                                                         REA 563
  710 CONTINUE                                                          REA 564
      XNUIJ(I,J)=0.                                                     REA 565
  720 CONTINUE                                                          REA 566
      DO 730 K=1,NPSS                                                   REA 567
      JPS=RPRP(K+9,II)+0.1                                              REA 568
      IF (JPS.NE.IS(J)) GO TO 730                                       REA 569
      XNUIJP(I,J)=RPRP(K+15,II)                                         REA 570
      GO TO 740                                                         REA 571
  730 CONTINUE                                                          REA 572
      XNUIJP(I,J)=0.                                                    REA 573
  740 CONTINUE                                                          REA 574
      NK=RPRP(19,II)+0.1                                                REA 575
      DO 750 K=1,NK                                                     REA 576
      ISKUR=RPRP(K+19,II)+0.1                                           REA 577
      IF (ISKUR.EQ.IS(J)) GO TO 760                                     REA 578
  750 CONTINUE                                                          REA 579
      KUR(I,J)=0                                                        REA 580
      GO TO 770                                                         REA 581
  760 KUR(I,J)=1                                                        REA 582
  770 CONTINUE                                                          REA 583
      IF (IGAS.EQ.0.OR.NT.EQ.1) GO TO 780                               REA 584
      KTF(I)=TNEP(I,INT)                                                REA 585
      KTR(I)=TNEP(I+25,INT)                                             REA 586
      ITR(I)=TNEP(I+50,INT)                                             REA 587
      EPAR(1,I)=TNEP(I+75,INT)                                          REA 588
      EPAR(2,I)=TNEP(I+100,INT)                                         REA 589
  780 CONTINUE                                                          REA 590
      IF (NT.EQ.1) GO TO 860                                            REA 591
      IF (IGAS.EQ.0) GO TO 800                                          REA 592
      DO 790 I=1,30                                                     REA 593
      TLIST(I)=TNEP(I+125,INT)                                          REA 594
  790 POM(I)=TNEP(I+155,INT)                                            REA 595
      BPAR=TNEP(186,INT)                                                REA 596
  800 DO 820 I=2,ISS                                                    REA 597
      IF (LPIJ(I,1).EQ.0) GO TO 810                                     REA 598
      ICH(I)=2                                                          REA 599
      GO TO 820                                                         REA 600
  810 ICH(I)=1                                                          REA 601
  820 CONTINUE                                                          REA 602
      DO 850 I=1,ISR                                                    REA 603
      IF (KTF(I).NE.4) GO TO 840                                        REA 604
```

```
      DO 830 J=2,ISS                                                    REA 605
      IF (XNUIJP(I,J).EQ.0.) GO TO 830                                  REA 606
      IPA(I)=J                                                          REA 607
      GO TO 850                                                         REA 608
  830 CONTINUE                                                          REA 609
  840 IPA(I)=0                                                          REA 610
  850 CONTINUE                                                          REA 611
  860 IZERO=0                                                           REA 612
      SUPGO=IGAS.NE.0.AND.IGAS.EQ.IGASO                                 REA 613
      IF (SUPGO) GO TO 870                                              REA 614
      WRITE (6,1140)                                                    REA 615
      WRITE (6,1150)                                                    REA 616
  870 CGMW=0.                                                           REA 617
      DO 890 I=1,NCS                                                    REA 618
      JJ=JCS(I)                                                         REA 619
      NELS=SPRP(2,JJ)                                                   REA 620
      CSMW=0.                                                           REA 621
      DO 880 K=1,NELS                                                   REA 622
      KK=SPRP(K+2,JJ)                                                   REA 623
      ELNAME(K)=EPRP(1,KK)                                              REA 624
      NATOM(K)=SPRP(K+5,JJ)                                             REA 625
      CSMW=CSMW+NATOM(K)*EPRP(2,KK)                                     REA 626
      IF (KK.NE.1) GO TO 880                                            REA 627
      ELNAME(K)=PLUS                                                    REA 628
      NATOM(K)=-NATOM(K)                                                RLA 629
  880 CONTINUE                                                          REA 630
      CGMW=CGMW+CSMW*QPJ(I)                                             REA 631
      IF (SUPGO) GO TO 890                                              REA 632
      WRITE (6,1160) I,SPRP(1,JJ),JJ,QPJ(I),CSMW,(SY1(K),ELNAME(K),SY2(KREA 633
     1),NATOM(K),K=1,NELS)                                              REA 634
  890 CONTINUE                                                          REA 635
      IF (.NOT.SUPGO) WRITE (6,1170) CGMW                               REA 636
      IGASO=IGAS                                                        REA 637
      NOZZO=NOZZLE                                                      REA 638
      ICHANO=ICHAN                                                      REA 639
      IGAS=IGS                                                          REA 640
      IF (XMODP1) 900,900,910                                           REA 641
  900 XMODP1=1.E20                                                      REA 642
      WRITE (6,1180)                                                    REA 643
  910 XMP1=XMODP1*2.54                                                  REA 644
      IF (CXMAXI.GT.XMODP1.AND.NMODPT.GT.1) GO TO 920                   REA 645
      FACMP=1.E10                                                       REA 646
      GO TO 930                                                         REA 647
  920 FNMP=NMODPT-1                                                     REA 648
      EXPMP=1./FNMP                                                     REA 649
      FACMP=(CXMAXI/XMODP1)**EXPMP                                      REA 650
  930 CONTINUE                                                          REA 651
      IF (NRECO.LE.0.OR.ICASE.GT.1) GO TO 950                           REA 652
      DO 940 I=1,NRECO                                                  REA 653
  940 READ (ITPOUT)                                                     REA 654
  950 CONTINUE                                                          REA 655
      RETURN                                                            REA 656
C                                                                       REA 657
CC                                                                      REA 658
C                                                                       REA 659
```

```
   96    FORMAT (20A4)                                                          REA 660
   970   FORMAT (1H1)                                                           REA 661
   980   FORMAT (1H1,55X,20HNATA III CODE OUTPUT/8HORUN NO.,I7,7X,4HCASE,I4REA 662
        1,12H IN THIS JOB,9X,20A4///)                                          REA 663
   990   FORMAT (30HOINVALID INPUT DATA... NPRFLS=,I10)                         REA 664
  1000   FORMAT (21H STAGNATION ENTHALPY=,F8.0,25H BTU/LB, TOTAL MASS FLOW=REA 665
        1,F9.5,7H LB/SEC)                                                      REA 666
  1010   FORMAT (20H RESERVOIR PRESSURE=,F9.4,23H ATM, TOTAL MASS FLOW =,F9REA 667
        1.5,7H LB/SEC)                                                         REA 668
  1020   FORMAT (20H RESERVOIR PRESSURE=,F9.4,28H ATM, RESERVOIR TEMPERATURREA 669
        1E=,F9.2,6H DEG K)                                                     REA 670
  1030   FORMAT (//13H NONSTANDARD ,A4,13H CHANNEL FOR ,A4,2H, ,F7.4,3H BY,REA 671
        1F7.4,12H INCH THROAT)                                                 REA 672
  1040   FORMAT (//1X,A4,30H CHANNEL (STANDARD CHANNEL NO.,I2,5H) FOR,1X,A4REA 673
        1,2H, ,F7.4,3H BY,F7.4,12H INCH THROAT)                                REA 674
  1050   FORMAT (//13H NONSTANDARD ,2A6,A3,8H NOZZLE,,F7.3,13H INCH THROAT REA 675
        1,A4,5H FOR ,A4)                                                       REA 676
  1060   FORMAT (//10H STANDARD ,2A6,A3,11H NOZZLE NO.,I3,1H,,F7.3,13H INCHREA 677
        1 THROAT ,A4,5H FOR ,A4)                                               REA 678
  1070   FORMAT (/60X,11HPROFILE NO.,I2)                                        REA 679
  1080   FORMAT (15HCTHRCAT RADIUS=,F7.3,3H CM,5X,8HINLET AT,F8.3,3H CM,5X,REA 680
        1I2,17H SECTIONS IN FIT,,I2,19H UPSTREAM OF THROAT)                    REA 681
  1090   FORMAT (1HC,3X,1HJ,6X,9HISHAPE(J),15X,5HSHAPE,13X,7HATPI(J),10X,10REA 682
        1HPARAM(1,J),10X,10HPARAM(2,J),10X,10HPARAM(3,J))                      REA 683
  1100   FORMAT (I5,I15,4X,4A4,1P4E20.4)                                        REA 684
  1110   FORMAT (I5,I15,4X,4A4,20X,1P3E20.4)                                    REA 685
  1120   FORMAT (//24H NONSTANDARD GAS MIXTURE)                                 REA 686
  1130   FORMAT (//18H STANDARD GAS NO. ,I2,2H (,A6,1H))                       REA 687
  1140   FORMAT (//56X,12HCOLD SPECIES)                                         REA 688
  1150   FORMAT (1HC,16X,3HNO.,4X,4HNAME,7X,5HINDEX,10X,10HMOLE FRAC.,12X,8REA 689
        1HMOL. WT.,14X,16HCHEMICAL FORMULA)                                    REA 690
  1160   FORMAT (10X,I10.5X,A5,I10,F20.5,F20.4,15X,3(A1,A2,A1,I1,1X))          REA 691
  1170   FORMAT (35HOMEAN MOLECULAR WEIGHT OF COLD GAS=,F9.4///)                REA 692
  1180   FORMAT (61HONEGATIVE OR ZERO VALUES OF XMODP1 NOT ALLOWED. DATA IGREA 693
        1NORED.)                                                               REA 694
  1190   FORMAT (49HONO THERMAL PROPERTY DATA DEFINED FOR SPECIES NO.,I3,30REA 695
        1H IN THE CURRENT GAS MODEL (NO.,I3,31H IN THE MASTER LIST OF SPECIREA 696
        2ES))                                                                  REA 697
        END                                                                    REA 698-
```

```
      SUBROUTINE RESET                                                  RST    1
C     SIMULATION OF UNIVAC 1108 TIMING ROUTINES RESET AND TIME          RST    2
      IN=1                                                              RST    3
      GO TO 10                                                          RST    4
      ENTRY TIME(I)                                                     RST    5
      IN=2                                                              RST    6
10    CALL ACUCPU (ICPU)                                                RST    7
      GO TO (20,30), IN                                                 RST    8
20    IZERO=ICPU                                                        RST    9
      GO TO 40                                                          RST   10
30    I=IZERO-ICPU                                                      RST   11
40    RETURN                                                            RST   12
      END                                                               RST   13-
```

```
      SUBROUTINE RESTMP                                                 RES    1
C     CALCULATES RESERVOIR TEMPERATURE FROM MASS FLOW DATA.             RES    2
      DOUBLE PRECISION AM,BM                                            RES    3
      LOGICAL ERR,SUPGO                                                 RES    4
      REAL ACOM(30),ELMENT(10),HP(20)                                   RES    5
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              RES    6
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             RES    7
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  RES    8
     1      C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  RES    9
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,  RES   10
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  RES   11
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  RES   12
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  RES   13
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,  RES   14
     7   SEN,  SHPG,    SC,    SL, SL64,    SM,    SU,   SU2,  SUMG,  RES   15
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                 RES   16
      COMMON         BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),RES   17
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),RES   18
     2CLN1MC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),RES   19
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),RES   20
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),RES   21
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),RES   22
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) RES   23
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),         RES   24
     1 XNUIJ(64,20),XNUIJP(64,20)                                       RES   25
      COMMON      IC,     IM,   INEQ,  INEQV,     IP,   IRUN,    ISC, ISCP1,  RES   26
     1  ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  RES   27
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,     RES   28
     3 ISW5B, ISW6A, ISW6B,  IUPD,  IZERO,   JJK,    LC,    M1,  NFIT,  RES   29
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        RES   30
      COMMON   IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) RES   31
      COMMON ACOM,ELMENT,HP                                             RES   32
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBRES   33
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                RES   34
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              RES   35
     1     (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),        RES   36
     2     (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),             RES   37
     3     (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                        RES   38
      COMMON /RDMAIN/ HS,SUPGO,MFITER,NOTRAN                            RES   39
      COMMON /MASSFL/ SMASS,CTMXXI,TSTOPI,IS(20)                        RES   40
      COMMON /ERROR/ ERR                                                RES   41
      DATA RNAME /6HRESTMP/                                             RES   42
      DIMENSION AM(3,3),BM(3),F(3),CTSAVE(3)                            RES   43
      NAMELIST /RSTDMP/ N,CTSAVE,F,D1,D2,DCT,DMC,BM,CONST,SM,CMA,DCTAPO RES   44
      NAMELIST/RSTDP2/ N,N2,N3,SMASS,HS,PRESA,CTAP,H,SMP,HO,CTAPP,      RES   45
     1     TLAST,PLAST                                                  RES   46
      DATA TERR,PERR /2*0.0001/                                         RES   47
      DATA RGAS /8.31434E7/                                             RES   48
      IF (ISW5A.NE.0) WRITE (6,200)                                     RES   49
      N=0                                                               RES   50
      IF (ISW2B.LT.0) GO TO 90                                          RES   51
      PCGS=PRESA*1.01325E6                                              RES   52
      CONST=RGAS*(SMASS/PCGS)**2                                        RES   53
C     STANDARD GUESS FOR FIRST POINT                                    RES   54
      CTAP=10000.                                                       RES   55
```

```
10    N=N+1                                                                    RES   56
      IF (N.GT.20) GO TO 160                                                   RES   57
      DO 20 I=1,2                                                              RES   58
      J=3-I                                                                    RES   59
      CTSAVE(J+1)=CTSAVE(J)                                                    RES   60
20    F(J+1)=F(J)                                                              RES   61
      CALL INIT                                                                RES   62
      CALL INTA                                                                RES   63
      IF (ERR) RETURN                                                          RES   64
      CALL NRMAX                                                               RES   65
      IF (ERR) RETURN                                                          RES   66
      CTSAVE(1)=CTAP                                                           RES   67
      F(1)=CMA*SM*SM                                                           RES   68
      IF (ISW5A.NE.0) WRITE (6,RSTOMP)                                         RES   69
      IF (N-2) 30,40,50                                                        RES   70
C     ITERATE TO GET SECOND POINT                                              RES   71
30    CTAP=F(1)/CONST                                                          RES   72
      GO TO 10                                                                 RES   73
C     LINEAR INTERPOLATION TO GET THIRD POINT                                  RES   74
40    DCT=CTSAVE(2)-CTSAVE(1)                                                  RES   75
      D1=(F(1)*CTSAVE(2)-F(2)*CTSAVE(1))/DCT                                   RES   76
      D2=(F(2)-F(1))/DCT                                                       RES   77
      CTAP=-D1/(D2-CONST)                                                      RES   78
      DCTAPO=ABS(CTAP-CTSAVE(1))                                               RES   79
      IF (DCTAPO/ABS(CTAP).LE.TERR) GO TO 190                                  RES   80
      GO TO 10                                                                 RES   81
C     QUADRATIC INTERPOLATION                                                  RES   82
50    DO 60 I=1,3                                                              RES   83
      AM(I,1)=1.00000000                                                       RES   84
      AM(I,2)=CTSAVE(I)                                                        RES   85
      AM(I,3)=CTSAVE(I)*CTSAVE(I)                                              RES   86
60    BM(I)=F(I)                                                               RES   87
      CALL SIMQ (AM,BM,3,KS)                                                   RES   88
      IF (KS.NE.0) GO TO 40                                                    RES   89
      DMC=BM(2)-CONST                                                          RES   90
      IF (BM(3).NE.0.) GO TO 70                                                RES   91
      CTAP=-BM(1)/DMC                                                          RES   92
      GO TO 80                                                                 RES   93
70    CTAP=(-DMC+SIGN(SQRT(DMC*DMC-4.*BM(1)*BM(3)),DMC))*0.5/BM(3)             RES   94
80    DCTAP=ABS(CTAP-CTSAVE(1))                                                RES   95
      IF (DCTAP/ABS(CTAP).LE.TERR) GO TO 190                                   RES   96
      IF (DCTAP.GE.DCTAPO) GO TO 40                                            RES   97
      DCTAPO=DCTAP                                                             RES   98
      GO TO 10                                                                 RES   99
C     ENTHALPY-MASS FLOW OPTION                                                RES  100
90    PRESA=4.8E-3*SMASS*SQRT(HS)                                              RES  101
      HERR=1.E-5*HS                                                            RES  102
      SMERR=1.E-5*SMASS                                                        RES  103
      CTAP=10000.                                                              RES  104
100   N=N+1                                                                    RES  105
      IF (N.GT.20) GO TO 160                                                   RES  106
      TLAST=CTAP                                                               RES  107
      PLAST=PRESA                                                              RES  108
      N2=0                                                                     RES  109
110   N2=N2+1                                                                  RES  110
```

```
      IF (N2.GT.20) GO TO 160                                          RES 111
      HO=H                                                             RES 112
      CALL INIT                                                        RES 113
      CALL INTA                                                        RES 114
      IF (ERR) RETURN                                                  RES 115
      H=CHA*CRP/CMA                                                    RES 116
      IF (ISW5A.NE.0) WRITE (6,RSTDP2)                                 RES 117
      IF (ABS(H-HS).LE.HERR) GO TO 130                                 RES 118
      IF (N2.GT.1) GO TO 120                                           RES 119
      CTAPP=CTAP                                                       RES 120
      CTAP=CTAP*HS/H                                                   RES 121
      GO TO 110                                                        RES 122
120   CTSV=CTAP                                                        RES 123
      CTAP=CTAP+(HS-H)*(CTAPP-CTAP)/(HO-H)                             RES 124
      CTAPP=CTSV                                                       RES 125
      GO TO 110                                                        RES 126
130   N3=0                                                             RES 127
140   N3=N3+1                                                          RES 128
      IF (N3.GT.20) GO TO 160                                          RES 129
      CALL NRMAX                                                       RES 130
      IF (ERR) RETURN                                                  RES 131
      SMP=SM*RHAP*SQRT(RGAS*CTAP/CMA)                                  RES 132
      IF (ISW5A.NE.0) WRITE (6,RSTDP2)                                 RES 133
      IF (ABS(SMP-SMASS).LE.SMERR) GO TO 150                           RES 134
      PRESA=PRESA*SMASS/SMP                                            RES 135
      CALL INTA                                                        RES 136
      IF (ERR) RETURN                                                  RES 137
      GO TO 140                                                        RES 138
150   IF (ABS(CTAP-TLAST).GT.TERR*TLAST) GO TO 100                     RES 139
      IF (ABS(PRESA-PLAST).GT.PERR*PLAST) GO TO 100                    RES 140
      GO TO 190                                                        RES 141
160   WRITE (6,210)                                                    RES 142
      IF (ISW2B.EQ.0) GO TO 170                                        RES 143
      WRITE (6,RSTDP2)                                                 RES 144
      GO TO 180                                                        RES 145
170   WRITE (6,RSTDMP)                                                 RES 146
180   CALL DUMP (RNAME)                                                RES 147
190   RETURN                                                           RES 148
C                                                                      RES 149
C                                                                      RES 150
C                                                                      RES 151
200   FORMAT (26H1TRACE OF RESTMP OPERATION)                           RES 152
210   FORMAT (30H0CONVERGENCE FAILURE IN RESTMP)                       RES 153
      END                                                              RES 154-
```

```
      SUBROUTINE RNKT                                                   RNK    1
      LOGICAL ERR,FAILED,TE                                             RNK    2
      REAL S1,S2,S3,T1,T2,T3,Z                                          RNK    3
      REAL ACOM(30),ELMENT(10),HP(20)                                   RNK    4
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              RNK    5
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             RNK    6
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,   ARBA,   ARBB, BZERO,  RNK    7
     1       C,  CARB,     CH,   CHA,  CLNT,     CM,   CMA,    CRA,   CRP,  RNK    8
     2    CRRB,   CRS,   CSTA,     CT,  CTAP,    CTB,   CTC, CTMAX, CTMXX,  RNK    9
     3     CTP,  CTPL,    CTT,     CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  RNK   10
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,   ENT,  FLUX, HDELX,   PCT,     RNK   11
     5PCTEST,   PRES, PRESA, PRESE,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,     RNK   12
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,     RNK   13
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,     RNK   14
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                   RNK   15
      COMMON          BE(64),    BET(20),  BLBK(31),    CAI(64),CAPXTH(20),RNK   16
     1   CCPJ(20), CEACT(64),    CGI(20),  CGMU(20),    CHI(64),  CHII(20),RNK   17
     2CLN1MC(64), CLNPI(64),    CMW(20),  ETAI(64),   ETAJ(20),   GJA(20),RNK   18
     3    GJB(20),PERTGJ(20),    PGJ(20),    PI(64), PICHI(64),    QM(20),RNK   19
     4     QQ(64),    SAJ(20), SDCHI(64),  SENT(20),    SHJ(20),  SHJA(20),RNK   20
     5   SKIL(20),     SS(20),     TB(30),   TFA(20),    TFB(20),   TFC(20),RNK   21
     6    TFD(20),    TFE(20),    TFK(20),  THEV(20), XMJAT(20),   XNU1(64) RNK   22
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),            RNK   23
     1 XNUIJ(64,20),XNUIJP(64,20)                                         RNK   24
      COMMON       IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  RNK   25
     1   ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,     RNK   26
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,     RNK   27
     3 ISW5B, ISW6A, ISW6B,   IUPD, IZERO,    JJK,     LC,     M1,  NFIT,  RNK   28
     4    NIT,    NNN,    NNS,    NQS,    NQT, NTEST                     RNK   29
      COMMON    IGJ(20),    IGM(20),    ITB(5),    KUR(64,20),   LPIJ(20,10) RNK   30
      COMMON ACOM,ELMENT,HP                                               RNK   31
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBRNK   32
     1J(20),SDGJ(20),SHJAP(20),THEVP(20)                                  RNK   33
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),                RNK   34
     1      (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),         RNK   35
     2      (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),              RNK   36
     3      (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                         RNK   37
      COMMON /ERROR/ ERR                                                  RNK   38
      COMMON /TNERK/ SDTE,CTEB,DCHA,CHB,SDCHA,DQMAX,IFAIL                 RNK   39
      COMMON /CERROR/ FAILED                                              RNK   40
      COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),         RNK   41
     1      KTR(25),ICH(20),IPA(25)                                       RNK   42
      COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE                                 RNK   43
      DIMENSION GJ1(23),GJ2(23),GJ3(23),GJ4(23),F1(23),F2(23),F3(23),     RNK   44
     1      F4(23),X1(23),X2(23),X3(23),X4(23),P(23),SDQ(23),FK(23,4),    RNK   45
     2      GJK(23,4),Y(4)                                                RNK   46
      EQUIVALENCE (Y1,Y(1)), (Y2,Y(2)), (Y3,Y(3)), (Y4,Y(4))              RNK   47
      EQUIVALENCE (SDQ(1),AA(1,1))                                        RNK   48
      EQUIVALENCE (FK(1,1),F1(1)), (FK(1,2),F2(1)), (FK(1,3),F3(1)),      RNK   49
     1      (FK(1,4),F4(1)), (GJK(1,1),GJ1(1)), (GJK(1,2),GJ2(1)),        RNK   50
     2      (GJK(1,3),GJ3(1)), (GJK(1,4),GJ4(1))                          RNK   51
      NAMELIST /RNKDMP/ CTB,CXB,CT,CX,DT,HDELX,DLOGA,DLOGR,SU,AFNX,       RNK   52
     1      PRES,RHO,GJB,DGJ,CTE,DTE,DCHA                                 RNK   53
      NAMELIST /RNKDP2/ SDT,SDTE,SDCHA,SDGJ,F1,F2,F3,F4,P                 RNK   54
      IFAIL=0                                                             RNK   55
```

```
      TE=NT.EQ.1                                                        RNK   56
      LIM=ISSP1                                                         RNK   57
      IF (.NOT.TE) LIM=LIM+2                                            RNK   58
      QDPEB=QDPE                                                        RNK   59
      DQMRK=2.*DQMAX                                                    RNK   60
      IF (ISWSB.EQ.0) GO TO 10                                          RNK   61
      ISTMNT=1                                                          RNK   62
      WRITE (6,410) ISTMNT                                              RNK   63
      WRITE (6,RNKDMP)                                                  RNK   64
   10 DO 20 J=1,ISS                                                     RNK   65
      F1(J)=DGJ(J)                                                      RNK   66
      GJ1(J)=GJB(J)                                                     RNK   67
      GJ(J)=GJB(J)+DGJ(J)*HDELX                                         RNK   68
   20 GJ2(J)=GJ(J)                                                      RNK   69
      F1(ISSP1)=DT                                                      RNK   70
      GJ1(ISSP1)=CTB                                                    RNK   71
      CT=CTB+DT*HDELX                                                   RNK   72
      GJ2(ISSP1)=CT                                                     RNK   73
      IF (TE) GO TO 30                                                  RNK   74
      F1(ISSP2)=DTE                                                     RNK   75
      GJ1(ISSP2)=CTEB                                                   RNK   76
      CTE=CTEB+DTE*HDELX                                                RNK   77
      GJ2(ISSP2)=CTE                                                    RNK   78
      F1(ISSP3)=DCHA                                                    RNK   79
      GJ1(ISSP3)=CHB                                                    RNK   80
      CHA=CHB+DCHA*HDELX                                                RNK   81
      GJ2(ISSP3)=CHA                                                    RNK   82
   30 DO 40 J=1,LIM                                                     RNK   83
      IFAIL=-J                                                          RNK   84
      IF (GJ2(J)) 270,40,40                                             RNK   85
   40 CONTINUE                                                          RNK   86
      IFAIL=0                                                           RNK   87
      CX=CXB+HDELX                                                      RNK   88
      CALL DERIVS                                                       RNK   89
      IF (ERR.OR.FAILED) RETURN                                         RNK   90
      IF (ISWSB.EQ.0) GO TO 50                                          RNK   91
      ISTMNT=2                                                          RNK   92
      WRITE (6,410) ISTMNT                                              RNK   93
      WRITE (6,RNKDMP)                                                  RNK   94
   50 DO 60 J=1,ISS                                                     RNK   95
      F2(J)=DGJ(J)                                                      RNK   96
      GJ(J)=GJB(J)+DGJ(J)*HDELX                                         RNK   97
   60 GJ3(J)=GJ(J)                                                      RNK   98
      F2(ISSP1)=DT                                                      RNK   99
      CT=CTB+DT*HDELX                                                   RNK  100
      GJ3(ISSP1)=CT                                                     RNK  101
      IF (TE) GO TO 70                                                  RNK  102
      F2(ISSP2)=DTE                                                     RNK  103
      CTE=CTEB+DTE*HDELX                                                RNK  104
      GJ3(ISSP2)=CTE                                                    RNK  105
      F2(ISSP3)=DCHA                                                    RNK  106
      CHA=CHB+DCHA*HDELX                                                RNK  107
      GJ3(ISSP3)=CHA                                                    RNK  108
   70 DO 80 J=1,LIM                                                     RNK  109
      IFAIL=-J                                                          RNK  110
```

```
      IF (GJ3(J)) 270,80,80                                                  RNK 111
80    CONTINUE                                                               RNK 112
      IF (TE) GO TO 90                                                       RNK 113
      IFAIL=-ISSP4                                                           RNK 114
      IF (ABS(QDPE-QCPEB).LE.DQMRK.OR.DQMAX.EQ.0.) GO TO 90                  RNK 115
      GO TO 270                                                              RNK 116
90    IFAIL=0                                                                RNK 117
      CALL DERIVS                                                            RNK 118
      IF (ERR.OR.FAILED) RETURN                                              RNK 119
      IF (ISW5B.EQ.0) GO TO 100                                              RNK 120
      ISTMNT=3                                                               RNK 121
      WRITE (6,410) ISTMNT                                                   RNK 122
      WRITE (6,RNKDMP)                                                       RNK 123
100   DO 110 J=1,ISS                                                         RNK 124
110   F3(J)=DGJ(J)                                                           RNK 125
      F3(ISSP1)=DT                                                           RNK 126
      IF (TE) GO TO 120                                                      RNK 127
      F3(ISSP2)=DTE                                                          RNK 128
      F3(ISSP3)=DCHA                                                         RNK 129
120   DO 150 J=1,LIM                                                         RNK 130
      IF (F1(J).EQ.0.0) GO TO 130                                            RNK 131
      X4(J)=F2(J)-F1(J)                                                      RNK 132
      IF (ABS(X4(J)/F1(J)).GE..0001) GO TO 140                               RNK 133
130   P(J)=0.0                                                               RNK 134
      GO TO 150                                                              RNK 135
140   XX=ABS(F3(J)-F2(J))                                                    RNK 136
      IF (XX.EQ.0.0) GO TO 130                                               RNK 137
      XX=ALOG10(XX)                                                          RNK 138
      YY=ALOG10(DABS(X4(J)))                                                 RNK 139
      IF ((XX-YY).GT.38.0) GO TO 130                                         RNK 140
      P(J)=2.0*(F3(J)-F2(J))/X4(J)                                           RNK 141
150   CONTINUE                                                               RNK 142
      DO 210 J=1,LIM                                                         RNK 143
      IF (P(J)) 160,210,210                                                  RNK 144
160   IF (P(J)+1.25) 200,200,170                                             RNK 145
170   Z=3.0                                                                  RNK 146
      S1=0.0                                                                 RNK 147
      S2=0.0                                                                 RNK 148
      S3=0.0                                                                 RNK 149
      T1=1.0                                                                 RNK 150
      T2=0.5                                                                 RNK 151
      T3=1.0/6.0                                                             RNK 152
180   S1=S1+T1                                                               RNK 153
      S2=S2+T2                                                               RNK 154
      S3=S3+T3                                                               RNK 155
      Z=Z+1.0                                                                RNK 156
      T1=T1*P(J)/(Z-2.0)                                                     RNK 157
      T2=T2*P(J)/(Z-1.0)                                                     RNK 158
      T3=T3*P(J)/Z                                                           RNK 159
      IF ((S3+T3)-S3) 180,190,180                                            RNK 160
190   X1(J)=S1                                                               RNK 161
      X2(J)=S2                                                               RNK 162
      X3(J)=S3                                                               RNK 163
      GO TO 210                                                              RNK 164
200   X1(J)=(EXP(P(J))-1.0)/P(J)                                             RNK 165
```

```
      X2(J)=(X1(J)-1.0)/P(J)                                                    RNK 166
      X3(J)=(X2(J)-0.5)/P(J)                                                    RNK 167
  210 CONTINUE                                                                  RNK 168
      DO 240 J=1,LIM                                                            RNK 169
      IF (P(J)) 230,220,220                                                     RNK 170
  220 GJ4(J)=GJ1(J)+F3(J)*DELTAX                                                RNK 171
      IFOR=1                                                                    RNK 172
      GO TO 240                                                                 RNK 173
  230 GJ4(J)=GJ1(J)+DELTAX*(2.0*F3(J)*X2(J)+F1(J)*(X1(J)-2.0*X2(J))-F2(JRNK 174
     1)*P(J)*X2(J))                                                             RNK 175
      IFOR=2                                                                    RNK 176
  240 CONTINUE                                                                  RNK 177
      DO 250 J=1,LIM                                                            RNK 178
      IFAIL=-J-30*IFOR                                                          RNK 179
      IF (GJ4(J)) 270,250,250                                                   RNK 180
  250 CONTINUE                                                                  RNK 181
      IF (TE) GO TO 260                                                         RNK 182
      IFAIL=-ISSP4-30*IFOR                                                      RNK 183
      IF (ABS(QDPE-QDPEB).LE.DQMRK.OR.DQMAX.EQ.0.) GO TO 260                    RNK 184
      GO TO 270                                                                 RNK 185
  260 IFAIL=0                                                                   RNK 186
      GO TO 280                                                                 RNK 187
  270 FAILED=.TRUE.                                                             RNK 188
      GO TO 400                                                                 RNK 189
  280 DO 290 J=1,ISS                                                            RNK 190
  290 GJ(J)=GJ4(J)                                                              RNK 191
      CT=GJ4(ISSP1)                                                             RNK 192
      IF (TE) GO TO 300                                                         RNK 193
      CTE=GJ4(ISSP2)                                                            RNK 194
      CHA=GJ4(ISSP3)                                                            RNK 195
  300 CX=CX8+DELTAX                                                             RNK 196
      CALL DERIVS                                                               RNK 197
      IF (ER .OR.FAILED) RETURN                                                 RNK 198
      IF (IS .EQ.0) GO TO 310                                                   RNK 199
      ISTMNT=                                                                   RNK 200
      WRITE ( ,410) ISTMNT                                                      RNK 201
      WRITE (6,RNKDMP)                                                          RNK 202
  310 DO 320 J=1,ISS                                                            RNK 203
  320 F4(J)=DGJ(J)                                                              RNK 204
      F4(ISSP1)=DT                                                              RNK 205
      IF (TE) GO TO 330                                                         RNK 206
      F4(ISSP2)=DTE                                                             RNK 207
      F4(ISSP3)=DCHA                                                            RNK 208
  330 DO 370 J=1,LIM                                                            RNK 209
      IF (P(J)) 350,340,340                                                     RNK 210
  340 SDQ(J)=(DELTAX/6.0)*(F1(J)+2.0*F2(J)+2.0*F3(J)+F4(J))                     RNK 211
      GO TO 370                                                                 RNK 212
  350 DO 360 K=1,4                                                              RNK 213
  360 Y(K)=FK(J,K)-(P(J)*GJK(J,K))/DELTAX                                       RNK 214
      SDQ(J)=DELTAX*(F1(J)*X1(J)+(-3.0*Y1+2.0*Y2+2.0*Y3-Y4)*X2(J)+4.0*(YRNK 215
     1-Y2-Y3+Y4)*X3(J))                                                         RNK 216
  370 CONTINUE                                                                  RNK 217
      DO 380 J=1,ISS                                                            RNK 218
  380 SDGJ(J)=SDQ(J)                                                            RNK 219
      SDT=SDQ(ISSP1)                                                            RNK 220
```

```
      IF (TE) GO TO 390                                                 RNK 221
      SDTE=SDQ(ISSP2)                                                   RNK 222
      SDCHA=SDQ(ISSP3)                                                  RNK 223
390   IF (ISW58.EQ.0) GO TO 400                                         RNK 224
      ISTMNT=5                                                          RNK 225
      WRITE (6,410) ISTMNT                                              RNK 226
      WRITE (6,RNKDMP)                                                  RNK 227
      WRITE (6,RNKDP2)                                                  RNK 228
400   RETURN                                                            RNK 229
C                                                                       RNK 230
C                                                                       RNK 231
C                                                                       RNK 232
410   FORMAT (18H DUMP AT STATEMENT,I2)                                 RNK 233
      END                                                               RNK 234-
```

```
      SUBROUTINE SHOCK (BMF,DEL,GAM,FL,IERR)                                SHK    1
C     KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC VERSION                    SHK    2
      IMPLICIT REAL*8 (A-H,O-Z)                                             SHK    3
C     HARVEY BUSS ROUTINE - OBLIQUE SHOCK IN A PERFECT GAS                  SHK    4
C                                                                           SHK    5
C     **********DEFINITIONS OF ARGUMENTS**********                         SHK    6
C     BMF=FREE STREAM MACH NUMBER                                           SHK    7
C     DEL=WEDGE HALF ANGLE (DEGREES)                                        SHK    8
C     GAM=RATIO OF SPECIFIC HEATS                                           SHK    9
C     FL(1)=SHOCK ANGLE (RADIANS)                                           SHK   10
C     FL(2)=MACH NUMBER BEHIND SHOCK                                        SHK   11
C     FL(3)=DENSITY RATIO, RHO(BEHIND)/RHO(INFINITY)                        SHK   12
C     FL(4)=STATIC PRESSURE RATIO, P(BEHIND)/P(INFINITY)                    SHK   13
C     FL(5)=STATIC TEMPERATURE RATIO, T(BEHIND)/T(INFINITY)                 SHK   14
C     FL(6)=TOTAL PRESSURE RATIO, PT(BEHIND)/PT(INFINITY)                   SHK   15
C     FL(7)= TSTAG/TINF                                                     SHK   16
C     IERR=1 NORMAL RETURN, IERR=2 ERROR RETURN, IERR=0 DETACHED SHOCK      SHK   17
C                                                                           SHK   18
      DIMENSION FL(7)                                                       SHK   19
      KIT=0                                                                 SHK   20
      DELR=DEL/57.295779                                                    SHK   21
      SD=SIN(DELR)                                                          SHK   22
      CD=COS(DELR)                                                          SHK   23
      BM2=BMF**2                                                            SHK   24
      BM4=BM2**2                                                            SHK   25
      C1=GAM+1.0                                                            SHK   26
      C2=GAM-1.0                                                            SHK   27
      B=-(BM2+2.0)/BM2-GAM*SD**2                                            SHK   28
      C=(2.*BM2+1.)/BM4+(C1**2/4.+C2/BM2)*SD**2                             SHK   29
      D=-CD**2/BM4                                                          SHK   30
      C3=2.*B                                                               SHK   31
      UG=0.                                                                 SHK   32
10    FU=D+UG*(C+UG*(B+UG))                                                 SHK   33
      FPU=C+UG*(C3+3.*UG)                                                   SHK   34
      UN=UG-FU/FPU                                                          SHK   35
      IF (DABS(UG/UN-1.D0)-1.D-8) 50,50,20                                  SHK   36
20    IF (KIT-50) 30,30,40                                                  SHK   37
30    KIT=KIT+1                                                             SHK   38
      UG=UN                                                                 SHK   39
      GO TO 10                                                              SHK   40
40    WRITE (6,100)                                                         SHK   41
      IERR=2                                                                SHK   42
      RETURN                                                                SHK   43
50    ALP=B+UN                                                              SHK   44
      BETA=UN*ALP+C                                                         SHK   45
      DIS=ALP**2-4.*BETA                                                    SHK   46
      IF (DIS.GE.0.) GO TO 60                                               SHK   47
      WRITE (6,110)                                                         SHK   48
      IERR=0                                                                SHK   49
      RETURN                                                                SHK   50
60    DIS=SQRT(DIS)                                                         SHK   51
      U1=(-ALP+DIS)/2.0                                                     SHK   52
      U2=(-ALP-DIS)/2.0                                                     SHK   53
      IF (U1-U2) 70,70,80                                                   SHK   54
70    RT=U1                                                                 SHK   55
```

```
      GO TO 90                                                                  SHK   56
80    RT=U2                                                                     SHK   57
C     80 ZT= SQRT(RT)                                                           SHK   58
90    CONTINUE                                                                  SHK   59
C     FL(1)=ATAN(ZT/SQRT(1.-ZT**2))                                             SHK   60
      V1=BM2*RT                                                                 SHK   61
C     V2=C1**2*BM4*RT-4.0  *(V1-1.0  )*(GAM*V1+1.0  )                           SHK   62
C     V3=(2.0  *GAM*V1-C2)*(C2*V1+2.0)                                          SHK   63
C     FL(2)= SQRT(V2/V3)                                                        SHK   64
C     FL(3)=C1*V1/(C2*V1+2.0)                                                   SHK   65
C     FL(5)=V3/C1**2/V1                                                         SHK   66
      FL(4)=(2.0*GAM*V1-C2)/C1                                                  SHK   67
C     C4=C2/2.0                                                                 SHK   68
C     BME2=FL(2)**2                                                             SHK   69
C     C5=GAM/C2                                                                 SHK   70
C     SMFP=1.0  +C4*BM2                                                         SHK   71
C     SMFP=1.0  +C4*BME2                                                        SHK   72
C     FL(6)=(SMEP/SMFP)**C5*FL(4)                                               SHK   73
C     FL(7)=SMFP                                                                SHK   74
      IERR=1                                                                    SHK   75
      RETURN                                                                    SHK   76
C                                                                               SHK   77
100   FORMAT (32H EXCEEDED 50 ITERATIONS IN SHOCK)                              SHK   78
110   FORMAT (110H ----SHOCK DETACHED IN CLASSICAL WEDGE CALCULATION. UNSHK   79
     1MODIFIED CHENG-KEMP THEORY IS USED FOR REMAINING ANGLES.)                 SHK   80
      END                                                                       SHK   81-
```

```
C     ******************************************************************SIM    1
                                                                         SIM    2
      SUBROUTINE SIMQ                                                    SIM    3
                                                                         SIM    4
      PURPOSE                                                            SIM    5
         OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS,      SIM    6
         AX=B                                                            SIM    7
                                                                         SIM    8
      USAGE                                                              SIM    9
         CALL SIMQ(A,B,N,KS)                                             SIM   10
                                                                         SIM   11
      DESCRIPTION OF PARAMETERS                                          SIM   12
         A - MATRIX OF COEFFICIENTS STORED COLUMNWISE.  THESE ARE        SIM   13
             DESTROYED IN THE COMPUTATION.  THE SIZE OF MATRIX A IS      SIM   14
             N BY N.                                                     SIM   15
         B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N).  THESE ARE         SIM   16
             REPLACED BY FINAL SOLUTION VALUES, VECTOR X.                SIM   17
         N - NUMBER OF EQUATIONS AND VARIABLES.                          SIM   18
         KS - OUTPUT DIGIT                                               SIM   19
              0 FOR A NORMAL SOLUTION                                    SIM   20
              1 FOR A SINGULAR SET OF EQUATIONS                          SIM   21
                                                                         SIM   22
      REMARKS                                                            SIM   23
         MATRIX A MUST BE GENERAL.                                       SIM   24
         IF MATRIX IS SINGULAR , SOLUTION VALUES ARE MEANINGLESS.        SIM   25
         AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX         SIM   26
         INVERSION (MINV) AND MATRIX PRODUCT (GMPRD).                    SIM   27
                                                                         SIM   28
      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                      SIM   29
         NONE                                                            SIM   30
                                                                         SIM   31
      METHOD                                                             SIM   32
         METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL      SIM   33
         DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGINGSIM   34
         ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL          SIM   35
         ELEMENTS.                                                       SIM   36
         THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN            SIM   37
         N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES IS          SIM   38
         CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL SOLUTION          SIM   39
         VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),      SIM   40
         VARIABLE 2 IN B(2)......... VARIABLE N IN B(N).                 SIM   41
         IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.0,          SIM   42
         THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1. THIS      SIM   43
         TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT.     SIM   44
                                                                         SIM   45
C     ******************************************************************SIM   46
                                                                         SIM   47
      SUBROUTINE SIMQ (A,B,N,KS)                                         SIM   48
C     KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC VERSION ***************SIM   49
      IMPLICIT REAL*8 (A-H,O-Z)                                          SIM   50
      DIMENSION A(1),B(1)                                                SIM   51
      DATA TOL /0.D0/                                                    SIM   52
      KS=0                                                               SIM   53
      IF (N.GT.1) GO TO 10                                               SIM   54
C                                                                        SIM   55
```

```
C     SPECIAL CASE N=1                                                  SIM   56
C                                                                       SIM   57
      IF (DABS(A(1)).LE.TOL) GO TO 40                                   SIM   58
      B(1)=B(1)/A(1)                                                    SIM   59
      RETURN                                                            SIM   60
C                                                                       SIM   61
CC       FORWARD SOLUTION                                               SIM   62
C                                                                       SIM   63
10    JJ=-N                                                             SIM   64
      DO 90 J=1,N                                                       SIM   65
      JY=J+1                                                            SIM   66
      JJ=JJ+N+1                                                         SIM   67
      BIGA=0                                                            SIM   68
      IT=JJ-J                                                           SIM   69
      DO 30 I=J,N                                                       SIM   70
C                                                                       SIM   71
CC       SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                       SIM   72
C                                                                       SIM   73
      IJ=IT+I                                                           SIM   74
      IF (DABS(BIGA)-DABS(A(IJ))) 20,30,30                              SIM   75
20    BIGA=A(IJ)                                                        SIM   76
      IMAX=I                                                            SIM   77
30    CONTINUE                                                          SIM   78
C                                                                       SIM   79
CC       TEST FOR PIVCT LESS THAN TOLERANCE (SINGULAR MATRIX)           SIM   80
C                                                                       SIM   81
      IF (DABS(BIGA)-TOL) 40,40,50                                      SIM   82
40    KS=1                                                              SIM   83
      RETURN                                                            SIM   84
C                                                                       SIM   85
CC       INTERCHANGE ROWS IF NECESSARY                                  SIM   86
C                                                                       SIM   87
50    I1=J+N*(J-2)                                                      SIM   88
      IT=IMAX-J                                                         SIM   89
      DO 60 K=J,N                                                       SIM   90
      I1=I1+N                                                           SIM   91
      I2=I1+IT                                                          SIM   92
      SAVE=A(I1)                                                        SIM   93
      A(I1)=A(I2)                                                       SIM   94
      A(I2)=SAVE                                                        SIM   95
C                                                                       SIM   96
CC       DIVIDE EQUATION BY LEADING COEFFICIENT                         SIM   97
C                                                                       SIM   98
60    A(I1)=A(I1)/BIGA                                                  SIM   99
      SAVE=B(IMAX)                                                      SIM  100
      B(IMAX)=B(J)                                                      SIM  101
      B(J)=SAVE/BIGA                                                    SIM  102
C                                                                       SIM  103
CC       ELIMINATE NEXT VARIABLE                                        SIM  104
C                                                                       SIM  105
      IF (J-N) 70,100,70                                                SIM  106
70    IQS=N*(J-1)                                                       SIM  107
      DO 90 IX=JY,N                                                     SIM  108
      IXJ=IQS+IX                                                        SIM  109
      IT=J-IX                                                           SIM  110
```

```
      DO 80 JX=JY,N                                                    SIM 111
      IXJX=N*(JX-1)+IX                                                 SIM 112
      JJX=IXJX+IT                                                      SIM 113
80    A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))                                  SIM 114
90    B(IX)=B(IX)-(B(J)*A(IXJ))                                        SIM 115
C                                                                      SIM 116
C        BACK SOLUTION                                                 SIM 117
C                                                                      SIM 118
100   NY=N-1                                                           SIM 119
      IT=N*N                                                           SIM 120
      DO 110 J=1,NY                                                    SIM 121
      IA=IT-J                                                          SIM 122
      IB=N-J                                                           SIM 123
      IC=N                                                             SIM 124
      DO 110 K=1,J                                                     SIM 125
      B(IB)=B(IB)-A(IA)*B(IC)                                          SIM 126
      IA=IA-N                                                          SIM 127
110   IC=IC-1                                                          SIM 128
      RETURN                                                           SIM 129
      END                                                              SIM 130-
```

```
      SUBROUTINE STUNTS                                                 STU    1
      REAL ACOM(30),ELMENT(10),HP(20)                                   STU    2
      DOUBLE PRECISION AA,CAPX,CDIJ                                     STU    3
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             STU    4
      COMMON         A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,  STU    5
     1       C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,  STU    6
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,    STU    7
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,    STU    8
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,    STU    9
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,    STU   10
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,    STU   11
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,    STU   12
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  STU   13
      COMMON           BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),STU   14
     1  CCPJ(20), CEACT(64),   CGI(20),   CGMU(20),   CHI(64),  CHII(20),STU   15
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),STU   16
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICH (64),    QM(20),STU   17
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),STU   18
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),STU   19
     6   TFD(20),   TFE(20),   TFK(20), THEV(20), XMJAT(20),  XNUI(64)  STU   20
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),          STU   21
     1 XNUIJ(64,20),XNUIJP(64,20)                                       STU   22
      COMMON      IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  STU   23
     1  ISMC,ISMCNR,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,    STU   24
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,    STU   25
     3 ISW5B, ISW6A, ISW6B,  IUPD,  IZERO,   JJK,    LC,     M1,   NFIT,  STU   26
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        STU   27
      COMMON    IGJ(20),   IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10) STU   28
      COMMON ACOM,ELMENT,HP                                             STU   29
      COMMON /SPEC/ SPRP(43,30)                                         STU   30
      COMMON /TEMPRY/ SAVEC(20)                                         STU   31
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                              STU   32
      COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,IIIII(50)        STU   33
      COMMON /TRANV/ VV(5,100)                                          STU   34
      COMMON /TRANJ/ JJQ(5,100)                                         STU   35
      COMMON /TRANI/ IIQ(5,100)                                         STU   36
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400) ,NKQ       STU   37
      COMMON /TRANS8/ NV(15),NN                                         STU   38
      DATA ASTRSK /'******'/                                            STU   39
      DIMENSION DELTT(6),NDT(6),TT(74),TPROP(74,4,2)                    STU   40
      DATA NDT /10,10,14,10,15,15/, DELTT /100.,200.,500.,1000.,2000.,  STU   41
     1     5000./                                                       STU   42
      EQUIVALENCE (TT(1),BETA(1,1)), (TPROP(1,1,1),BETA(1,3))           STU   43
      DO 110 J=1,ISS                                                    STU   44
      H00=SHJA(J)*CRP/1000.                                             STU   45
      LIM1=1                                                            STU   46
      WRITE (6,290) HP(J),H00                                           STU   47
      IF (IGJ(J)) 20,10,20                                              STU   48
   10 WRITE (6,300) (ASTRSK,K=1,6)                                      STU   49
      LIM=1                                                             STU   50
      GO TO 50                                                          STU   51
   20 WRITE (6,310) (ASTRSK,K=1,12)                                     STU   52
      LIM=2                                                             STU   53
      CTMXX=1000.                                                       STU   54
      IF (ETAJ(J).GT.0.) GO TO 50                                       STU   55
```

```
      LIM1=2                                                                STU  56
      CTMXX=-1000.                                                          STU  57
      DO 40 K=1,4                                                           STU  58
      DO 30 I=1,74                                                          STU  59
30    TPROP(I,K,1)=0.                                                       STU  60
40    CONTINUE                                                              STU  61
50    DO 90 M=LIM1,LIM                                                      STU  62
      IT=0                                                                  STU  63
      II=1                                                                  STU  64
      TV=0.                                                                 STU  65
      DO 70 I=1,74                                                          STU  66
      IT=IT+1                                                               STU  67
      IZ=I                                                                  STU  68
      IF (IT.LE.NDT(II)) GO TO 60                                           STU  69
      II=II+1                                                               STU  70
      IT=1                                                                  STU  71
60    TV=TV+DELTT(II)                                                       STU  72
      IF (M.EQ.1) TT(I)=TV                                                  STU  73
      CT=TV/CTAP                                                            STU  74
      CALL THERM                                                            STU  75
      TPROP(I,1,M)=XMJAT(J)-SHJA(J)/CT                                      STU  76
      TPROP(I,2,M)=(SHJ(J)-SHJA(J))*CRP/1000.                               STU  77
      TPROP(I,3,M)=CCPJ(J)*CRA                                              STU  78
      TPROP(I,4,M)=SENT(J)*CRA                                              STU  79
      IF (CT-1.) 70,80,80                                                   STU  80
70    CONTINUE                                                              STU  81
80    CTMXX=-1000.                                                          STU  82
90    CONTINUE                                                              STU  83
      DO 100 I=1,IZ                                                         STU  84
100   WRITE (6,320) TT(I),((TPROP(I,K,M),K=1,4),M=1,LIM)                    STU  85
110   CONTINUE                                                              STU  86
      RETURN                                                                STU  87
      ENTRY STUNT2                                                          STU  88
      WRITE (6,330)                                                         STU  89
      WRITE (6,340)                                                         STU  90
      WRITE (6,350)                                                         STU  91
      DO 150 L=1,NNKQ                                                       STU  92
      I=ISEQ(L)                                                             STU  93
      NNQI=NNQ(I)                                                           STU  94
      DO 140 J=1,NNQI                                                       STU  95
      IIQJ=IIQ(J,I)                                                         STU  96
      JJQJ=JJQ(J,I)                                                         STU  97
      IF (J-1) 120,120,130                                                  STU  98
120   WRITE (6,360) L,I,KKQ(I),(VV(K,I),K=1,5),SPRP(1,IIQJ),SPRP(1,JJQJ)    STU  99
      GO TO 140                                                             STU 100
130   WRITE (6,370) SPRP(1,IIQJ),SPRP(1,JJQJ)                               STU 101
140   CONTINUE                                                              STU 102
150   CONTINUE                                                              STU 103
      WRITE (6,330)                                                         STU 104
      WRITE (6,380)                                                         STU 105
      WRITE (6,350)                                                         STU 106
      MV2=1                                                                 STU 107
      LQ2=0                                                                 STU 108
      DO 190 L=1,NKQ                                                        STU 109
      KK=KQ(L)                                                              STU 110
```

```
      MV1=MV2+1                                                          STU 111
      MV2=MV2+NV(KK)                                                     STU 112
      LQ1=LQ2+1                                                          STU 113
      LQ2=NQ(L)                                                          STU 114
      WRITE (6,390) L,KK,(V(I),I=MV1,MV2)                                STU 115
      IF (LQ2.LT.LQ1) GO TO 190                                          STU 116
      DO 180 LQ=LQ1,LQ2                                                  STU 117
      I=IQ(LQ)                                                           STU 118
      J=JQ(LQ)                                                           STU 119
      IF (LQ-LQ1) 160,160,170                                            STU 120
  160 WRITE (6,400) HP(I),HP(J)                                          STU 121
      GO TO 180                                                          STU 122
  170 WRITE (6,370) HP(I),HP(J)                                          STU 123
  180 CONTINUE                                                           STU 124
  190 CONTINUE                                                           STU 125
      IF (ISW1B.GT.0) GO TO 240                                          STU 126
      WRITE (6,250)                                                      STU 127
      ICARD=0                                                            STU 128
      DO 200 I=1,ISS                                                     STU 129
  200 SAVEC(I)=GJA(I)*CMA                                                STU 130
      DO 230 I=1,ISS                                                     STU 131
      DO 220 J=I,ISS                                                     STU 132
      WRITE (6,260)                                                      STU 133
      DO 210 K=1,30                                                      STU 134
      TV=1000*K                                                          STU 135
      IF (TV.GT.1.00001*CTAP) GO TO 220                                  STU 136
      ICARD=ICARD+1                                                      STU 137
      CALL TRANSP (TV,PRESA)                                             STU 138
      WRITE (6,270) TV,(Q(L,I,J),L=1,3),I,J,HP(I),HP(J),ICARD            STU 139
      IF (ISW1B.NE.-1) GO TO 210                                         STU 140
      PUNCH 270, TV,(Q(L,I,J),L=1,3),I,J,HP(I),HP(J),ICARD               STU 141
  210 CONTINUE                                                           STU 142
  220 CONTINUE                                                           STU 143
  230 CONTINUE                                                           STU 144
      WRITE (6,280)                                                      STU 145
  240 RETURN                                                             STU 146
C                                                                        STU 147
C                                                                        STU 148
C                                                                        STU 149
  250  FORMAT (1H1,38X,56HAVERAGED PAIR CROSS-SECTIONS AS FUNCTIONS OF TESTU 150
     1MPERATURE)                                                         STU 151
  260  FORMAT (///2X,5HTEMP.,8X,4HQ(1),8X,4HQ(2),8X,4HQ(3),5X,7HINDICES,5STU 152
     1X,7HSPECIES,8X,5HCOUNT)                                            STU 153
  270  FORMAT (F7.0,1P3E12.3,5X,I3,1H-,I3,3X,/4,1H-,A4,5X,I8)            STU 154
  280  FORMAT (///10BH NOTE - CHARGED SPECIES CROSS SECTIONS AT ALL TEMPESTU 155
     1RATURES ARE BASED ON ELECTRON MOLE FRACTION IN RESERVOIR.)         STU 156
  290  FORMAT (1H1,40X,22HTHERMAL PROPERTIES OF ,A4,6H (H00=,F10.3,11H KCSTU 157
     1AL/MOLE))                                                          STU 158
  300  FORMAT (1H0,10X,3A6,18H**PHYSICAL MODEL**,3A6/1H0,8X,1HT,2X,12H(MUSTU 159
     10-H00)/RT,9X,5HH-H00,12X,2HCP,12X,2HS0/29X,9HKCAL/MOLE,2X,12HCAL/MSTU 160
     20LE-DEG,2X,12HCAL/MOLE-DEG)                                        STU 161
  310  FORMAT (1H0,11X,3A6,18H**PHYSICAL MODEL**,3A6,12X,3A6,18H****THERMSTU 162
     10 FIT****,3A6/1H0,8X,1HT,2X,12H(MU0-H00)/RT,9X,5HH-H00,12X,2HCP,12STU 163
     2X,2HS0,12X,12H(MU0-H00)/RT,9X,5HH-H00,12X,2HCP,12X,2HS0/29X,9HKCALSTU 164
     3/MOLE,2X,12HCAL/MOLE-DEG,2X,12HCAL/MOLE-DEG,29X,9HKCAL/MOLE,2X,12HSTU 165
```

```
     4CAL/MOLE-DEG,2X,12HCAL/MOLE-DEG)                                                  STU 166
320   FORMAT (F10.0,4F14.3,10X,4F14.3)                                                  STU 167
330   FORMAT (1H1,48X,28HTRANSPORT CROSS SECTION DATA)                                  STU 168
340   FORMAT (60X,7H(INPUT))                                                            STU 169
350   FORMAT (1H0,5X,4HSTEP,5X,5HINDEX,4X,6HOPTION,10X,5HVV(1),10X,5HVV(STU 170
     12),10X,5HVV(3),10X,5HVV(4),10X,5HVV(5),9X,11HINTERACTION)                        STU 171
360   FORMAT (3I10,1P5E15.3,9X,A4,3H - ,A4)                                             STU 172
370   FORMAT (114X,A4,3H - ,A4)                                                         STU 173
380   FORMAT (59X,8H(EDITED))                                                           STU 174
390   FORMAT (I10,I20,1P5E15.3)                                                         STU 175
400   FORMAT (1H+,113X,A4,3H - ,A4)                                                     STU 176
      END                                                                               STU 177-
```

```
      SUBROUTINE THERM                                                  THE    1
C     SPECIES THERMAL PROPERTIES                                        THE    2
      LOGICAL MIX,F1                                                    THE    3
      REAL ACOM(30),ELMENT(10),HP(20)                                   THE    4
      DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ                              THE    5
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             THE    6
      COMMON       A, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO, THE    7
     1      C,  CARB,    CH,   CHA,  CLNT,     CM,   CMA,   CRA,   CRP, THE    8
     2   CRRB,   CRS,  CSTA,    CT,  CTAP,    CTB,   CTC, CTMAX, CTMXX, THE    9
     3    CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  THE   10
     4 DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,  THE   11
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHO,  RHAP,   RHO,  RHOB,  THE   12
     6RHOBAR,  RHOC,  RHOP,  RHPL, RHTH,ROBARA,ROBARP,  SCPG,   SDT,   THE   13
     7   SEN,  SHPG,    SC,    SL,  SL64,    SM,    SU,   SU2,  SUMG,  THE   14
     8  TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                  THE   15
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),THE   16
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),THE   17
     2CLNMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),THE   18
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),THE   19
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),THE   20
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),THE   21
     6   TFD(20),   TFF(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) THE   22
      COMMON           BETA(64,20),   ELJ(10,20),  GELJ(10,20),         THE   23
     1 XNUIJ(64,20),XNUIJP(64,20)                                       THE   24
      COMMON      IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCPI,  THE   25
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,  THE   26
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,   THE   27
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,  THE   28
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                        THE   29
      COMMON  IGJ(20),  IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10)   THE   30
      COMMON ACOM,ELMENT,HP                                             THE   31
      DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBTHE   32
     J(20),SDGJ(20),SHJAP(20),THEVP(20)                                 THE   33
      EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),              THE   34
     1     (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),        THE   35
     2     (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),             THE   36
     3     (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))                        THE   37
      COMMON /POLYAT/ THEVE(4,20)                                       THE   38
C     COMPUTE ALL THERMAL PROPERTIES                                    THE   39
      E1=.FALSE.                                                        THE   40
      GO TO 10                                                          THE   41
      ENTRY THERM1                                                      THE   42
C     COMPUTE SPECIFIC HEATS ONLY                                       THE   43
      E1=.TRUE.                                                         THE   44
10    CLGT=ALOG(CT)                                                     THE   45
      CTP=CT*CTAP                                                       THE   46
      CLNT=ALOG(CTP)                                                    THE   47
      DO 320 J=1,ISS                                                    THE   48
      SZ1=0.                                                            THE   49
      SZ2=0.                                                            THE   50
      IF (ETAJ(J).LE.0.) GO TO 30                                       THE   51
      IF (IGJ(J)) 20,30,20                                              THE   52
20    ZT=CTAP*(CT-CTMXX)                                                THE   53
      IF (ABS(ZT).GE.500.) GO TO 40                                     THE   54
      IF (CTMXX.LE.0.) GO TO 40                                         THE   55
```

```
      MIX=.TRUE.                                                        THE  56
      WTTF=0.001*(ZT+500.)                                              THE  57
      WTHO=1.-WTTF                                                      THE  58
      GO TO 50                                                          THE  59
30    MIX=.FALSE.                                                       THE  60
      IF (ETAJ(J)) 50,50,70                                             THE  61
40    MIX=.FALSE.                                                       THE  62
      IF (CT-CTMXX) 70,50,50                                            THE  63
50    IF (E1) GO TO 60                                                  THE  64
C                                                                       THE  65
CC    THERMO FIT                                                        THE  66
C                                                                       THE  67
      XMJAT(J)=TFA(J)*(1.-CLNT)-TFK(J)-CT*(TFB(J)+CT*(TFC(J)/2.+CT*(TFD(THE  68
     1J)/3.+CT*TFE(J)/4.)))+SHJA(J)/CT                                  THE  69
      SHJ(J)=SHJA(J)+CT*(TFA(J)+CT*(TFB(J)+CT*(TFC(J)+CT*(TFD(J)+CT*TFE(THE  70
     1J))))                                                             THE  71
60    CCPJ(J)=TFA(J)+CT*(2.0*TFB(J)+CT*(3.0*TFC(J)+CT*(4.0*TFD(J)+CT*(5.THE  72
     10*TFE(J)))))                                                      THE  73
      IF (.NOT.MIX) GO TO 300                                           THE  74
      XMTF=XMJAT(J)                                                     THE  75
      SHTF=SHJ(J)                                                       THE  76
      CCPTF=CCPJ(J)                                                     THE  77
      STF=SHTF/CT-XMTF                                                  THE  78
C                                                                       THE  79
CC    PHYSICAL MODEL                                                    THE  80
C                                                                       THE  81
      CCPJ(J)=0.                                                        THE  82
70    IF (E1) GO TO 80                                                  THE  83
      SHJ(J)=0.                                                         THE  84
      XMJAT(J)=0.                                                       THE  85
80    IF (ETAJ(J).LT.3.) GO TO 150                                      THE  86
C     TRANSLATIONAL, ROTATIONAL, AND VIBRATIONAL CONTRIBUTIONS FOR      THE  87
C          LINEAR TRIATOMIC SPECIES                                     THE  88
      Z1=3.5*CLNT                                                       THE  89
      DO 140 K=1,4                                                      THE  90
      IF (INEQV) 90,100,90                                              THE  91
90    Z3=THEVE(K,J)                                                     THE  92
      GO TO 110                                                         THE  93
100   Z3=THEVE(K,J)/CT                                                  THE  94
      IF (Z3.GT.88.) GO TO 140                                          THE  95
110   Z4=EXP(Z3)                                                        THE  96
      Z5=Z4-1.                                                          THE  97
      Z6=THEVE(K,J)/Z5                                                  THE  98
      IF (INEQV.NE.IZERO) GO TO 120                                     THE  99
      Z7=Z3/Z5                                                          THE 100
      GO TO 130                                                         THE 101
120   Z7=0.                                                             THE 102
      SZ2=SZ2+Z6/CT                                                     THE 103
130   Z8=ALOG(Z4/Z5)                                                    THE 104
      Z9=Z7*(Z4*Z7)                                                     THE 105
      CCPJ(J)=CCPJ(J)+Z9                                                THE 106
      IF (E1) GO TO 140                                                 THE 107
      XMJAT(J)=XMJAT(J)-Z8                                              THE 108
      SHJ(J)=SHJ(J)+Z6                                                  THE 109
140   CONTINUE                                                          THE 110
```

```
      CCPJ(J)=CCPJ(J)+3.5                                                THE 111
      IF (E1) GO TO 240                                                  THE 112
      SZ1=SHJ(J)                                                         THE 113
      XMJAT(J)=XMJAT(J)-SAJ(J)-Z1+SHJA(J)/CT+(1.5+ETAJ(J))*CTPL          THE 114
      SHJ(J)=SHJ(J)+3.5*CT+SHJA(J)                                       THE 115
      GO TO 240                                                          THE 116
C     TRANSLATIONAL, ROTATIONAL, AND VIBRATIONAL CONTRIBUTIONS FOR       THE 117
C           MONATOMIC AND DIATOMIC SPECIES                               THE 118
150   Z1=ETAJ(J)-1.0                                                     THE 119
      Z2=Z1+2.5                                                          THE 120
      IF (E1) GO TO 160                                                  THE 121
      XMJAT(J)=0.                                                        THE 122
160   CCPJ(J)=0.                                                         THE 123
      IF (INEQV) 170,180,170                                             THE 124
170   Z3=THEV(J)                                                         THE 125
      GO TO 190                                                          THE 126
180   Z3=THEV(J)/CT                                                      THE 127
      IF (Z3.GT.88.) GO TO 230                                           THE 128
190   Z4=EXP(Z3)                                                         THE 129
      Z5=Z4-1.                                                           THE 130
      IF (Z1) 200,230,200                                                THE 131
200   Z6=THEV(J)/Z5                                                      THE 132
      IF (INEQV.NE.IZERO) GO TO 210                                      THE 133
      Z7=Z3/Z5                                                           THE 134
      GO TO 220                                                          THE 135
210   Z7=0.0                                                             THE 136
220   Z8=ALOG(Z4/Z5)                                                     THE 137
      Z9=Z7*(Z4*Z7)                                                      THE 138
      CCPJ(J)=Z1*Z9                                                      THE 139
      IF (E1) GO TO 230                                                  THE 140
      XMJAT(J)=-(Z1*Z8)                                                  THE 141
230   CCPJ(J)=CCPJ(J)+Z2                                                 THE 142
      IF (E1) GO TO 240                                                  THE 143
      SHJ(J)=Z2*CT+SHJA(J)+Z1*Z6                                         THE 144
      XMJAT(J)=XMJAT(J)+SHJA(J)/CT-(SAJ(J)+Z2*CLGT)                      THE 145
240   IF (IGM(J).EQ.0) GO TO 290                                         THE 146
C     ELECTRONIC CONTRIBUTIONS (ALL SPECIES)                             THE 147
      S1=0.0                                                             THE 148
      S2=0.0                                                             THE 149
      S3=0.0                                                             THE 150
      N=IGM(J)                                                           THE 151
      DO 250 M=1,N                                                       THE 152
      XX=GELJ(M,J)*EXP(-ELJ(M,J)/CT)                                     THE 153
      S1=S1+XX                                                           THE 154
      X   XX*ELJ(M,J)                                                    THE 155
      S   S2+XY                                                          THE 156
      X   XX*ELJ(M,J)                                                    THE 157
250   S3=S3+XX                                                           THE 158
      IF (E1) GO TO 260                                                  THE 159
      XMJAT(J)=XMJAT(J)-ALOG(S1)                                         THE 160
      SHJ(J)=SHJ(J)+S2/S1                                                THE 161
260   IF (S3.GT.1.D-37) GO TO 270                                        THE 162
      XX=0.                                                              THE 163
      GO TO 280                                                          THE 164
270   XX=S2*S2                                                           THE 165
```

```
280     CCPJ(J)=CCPJ(J)+(S1*S3-WW)/((S1*CT)**2)                          THE 166
290     IF (INDOV.EQ.0) GO TO 300                                        THE 167
        IF (ETAJ(J).GT.2.1) GO TO 300                                    THE 168
        IF (E1) GO TO 310                                                THE 169
C       ENTROPY CALCULATION                                              THE 170
        SENT(J)=Z2+(SHJA(J)+S2/S1)/CT+Z1*Z6-XMJAT(J)                     THE 171
        GO TO 310                                                        THE 172
  0     IF (E1) GO TO 310                                                THE 173
        SENT(J)=SHJ(J)/CT-XMJAT(J)                                       THE 174
        IF (INDOV.N..0) SENT(J)=SENT(J)+SZ1-SZ2                          THE 175
310     IF (.NOT.MIX) GO TO 320                                          THE 176
C       MIX PHYSICAL MODEL AND THERMO FIT RESULTS IN TRANSITION          THE 177
C            TEMPERATURE RANGE                                           THE 178
        CCPJ(J)=WTHO*CCPJ(J)+WTTF*CCPTF                                  THE 179
        IF (E1) GO TO 320                                                THE 180
        XMJA(J)=WTHO*XMJAT(J)+WTTF*XMTF                                  THE 181
        SHJ(J)=WTHO*SHJ(J)+WTTF*SHTF                                     THE 182
        SENT(J)=WTHO*SENT(J)+WTTF*STF                                    THE 183
320     CONTINUE                                                         THE 184
        RETURN                                                           THE 185
        END                                                              THE 186
```

```
      SUBROUTINE THROAT                                                 THR    1
      REAL ACOM(30),ELMENT(10),HP(20)                                   THR    2
      LOGICAL ERR                                                       THR    3
      DOUBLE PRECISION AA,CAPX,CDIJ                                     THR    4
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                             THR    5
      COMMON        A, AFNTS,   AFNX, AMACH,     AR,   ARBA,   ARBB, BZERO,  THR    6
     1      C,   CARB,     CH,    CHA,   CLNT,     CM,    CMA,    CRA,   CRP,  THR    7
     2   CRRO,    CRS,   CSTA,     CT,   CTAP,    CTB,    CTC,  CTMAX, CTMXX,  THR    8
     3    CTP,   CTPL,    CTT,     CX,    CXB,  CXMAX,DATEST,DBTEST, DELT1,   THR    9
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,    ENT,   FLUX,  HDELX,    PCT,   THR   10
     5PCTEST,   PRES,  PRESA, PRESB,PRESTH,   PRHC,   RHAP,    RHO,   RHOB,   THR   11
     6RHOBAR,   RHOC,   RHOP,   RHPL,   RHTH,ROBARA,ROBARP,   SCPG,    SDT,   THR   12
     7    SEN,   SHPG,     SC,     SL,   SL64,     SM,     SU,    SU2,   SUMG,  THR   13
     8   TEST,  TESTR,TPRINT, TSTOP,     UP,     ZP,    ZPA                 THR   14
      COMMON          BE(64),    BET(20),   BLBK(31),    CAI(64),CAPXTH(20),THR   15
     1  CCPJ(20),  CEACT(64),    CGI(20),   CGMU(20),    CHI(64),   CHII(20),THR   16
     2CLNIMC(64), CLNPI(64),    CMW(20),   ETAI(64),   ETAJ(20),    GJA(20),THR   17
     3    GJB(20),PERTGJ(20),    PGJ(20),     PI(64),  PICHI(64),    QM(20),THR   18
     4     QQ(64),    SAJ(20), S CHI(64),   SENT(20),    SHJ(20),   SHJA(20),THR   19
     5   SKIL(20),     SS(20),     TB(30),    TFA(20),    TFB(20),    TFC(20),THR   20
     6    TFD(20),    TFE(20),    TFK(20),   THEV(20), XMJAT(20),   XNUI(64) THR   21
      COMMON          BETA(64,20),    ELJ(10,20),   GFLJ(10,20),            THR   22
     1 XNUIJ(64,20),XNUIJP(64,20)                                       THR   23
      COMMON       IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  THR   24
     1   ISMC,ISMCNP,    ISR,    ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,     THR   25
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,        THR   26
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,    JJK,     LC,     M1,   NFIT,   THR   27
     4    NI,     NNN,    NNS,    NQS,    NQT, NTEST                     THR   28
      COMMON    IGJ(20),   IGM(20),    ITB(5),   KUR(64,20),  LPIJ(20,10) THR   29
      COMMON ACOM,ELMENT,HP                                             THR   30
      COMMON /ERROR/ ERR                                                THR   31
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  THR   32
     1    SW,RO,JDIM,IPOINT                                             THR   33
      COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),         THR   34
     1    NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL                   THR   35
      COMMON /NEQ/ QMOST(2),DDELBL(2)                                   THR   36
      COMMON /AEGEOM/ SQRTA,S1,S2                                       THR   37
      COMMON /THRT/ RSA                                                 THR   38
      DATA ONE /1./                                                     THR   39
      IUPD=0                                                            THR   40
      CALL GEOMAR (CX,S1,S2)                                            THR   41
      IF (ERR) RETURN                                                   THR   42
      IF (ISW3B,EQ,0) GO TO 10                                          THR   43
      CALL AGSOLN (AFNX,DELBL,ONE,AG,XX)                                THR   44
      IF (ERR) RETURN                                                   THR   45
      GO TO 20                                                          THR   46
   10 AG=AFNX                                                           THR   47
   20 RSA=RSA*AG/S1                                                     THR   48
      WRITE (6,30) CX,AMACH,AFNX,DLOGA,S1,S2,RSA,(DELBL(I),I=1,NPRFLS)  THR   49
      RETURN                                                            THR   50
C                                                                       THR   51
CC                                                                      THR   52
C                                                                       THR   53
   30 FORMAT (//24H ////////////// THROAT  CX=,1PE10,3,4X,6HAMACH=,E10,3,5XTHR   54
     1,5HAFNX=,E10,3,4X,6HDLOGA=,E10,3/21X,3HS1=,E10,3,7X,3HS2=,E10,3,6XTHR   55
```

```
THR  56
THR  57-
```

```
2.4HRSA=,E10.3,4X,6HDELEL=,E10.3,5X,E10.3//)
END
```

```
      SUBROUTINE TRANSP (TTAB,P)                                          TRA    1
C     SUBROUTINE TRANSP HAS GENERAL SUPERVISION OF ALL TRANSPORT          TRA    2
C     PROPERTY CALCULATIONS IN NATA CODE.                                 TRA    3
      LOGICAL ERR                                                         TRA    4
      REAL ACOM(30),ELMENT(10),HP(20)                                     TRA    5
      DOUBLE PRECISION AA,CAPX,CDIJ                                       TRA    6
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                               TRA    7
      COMMON      AZ, AFNTS,  AFNX, AMACH,     AR,  ARBA,  ARBB, BZERO,   TRA    8
     1     CZ,  CARB,     CH,   CHA,  CLNT,     CM,   CMA,   CRA,   CRP,  TRA    9
     2   CRBB,   CRS,   CSTA,     CT,  CTAP,    CTB,   CTC, CTMAX, CTMXX,  TRA   10
     3    CTP,  CTPL,    CTT,     CX,   CXB, CXMAX,DATEST,DBTEST, DELT1,  TRA   11
     4 DELT2,DELTAX, DLOGA, DLOGR,     DT,    ENT,  FLUX, HDELX,    PCT,  TRA   12
     5PCTEST,  PRES, PRESA, PRESR,PRESTH,   PRHC,  RHAP,    RHO,   RHOB,  TRA   13
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,   SCPG,    SDT,  TRA   14
     7    SEN,  SHPG,     SC,    SL,  SL64,     SM,    SU,   SU2,   SUMG,  TRA   15
     8   TEST, TESTB,TPRINT, TSTOP,    UP,     ZP,   ZPA                  TRA   16
      COMMON           BE(64),    BET(20),  BLBK(31),    CAI(64),CAPXTH(20),TRA   17
     1  CCPJ(20), CFACT(64),    CGI(20),   CGMU(20),    CHI(64),  CHII(20),TRA   18
     2CLNIMC(64), CLNPI(64),    CMW(20),   ETAI(64),   ETAJ(20),    GJA(20),TRA   19
     3    GJP(20),PERTGJ(20),    PGJ(20),     PI(64), PICHI(64),    QM(20),TRA   20
     4     QQ(64),    SAJ(20), SDCHI(64),   SENT(20),    SHJ(20),  SHJA(20),TRA   21
     5   SKIL(20),     SS(20),     TB(30),    TFA(20),    TFB(20),   TFC(20),TRA   22
     6    TFD(20),    TFE(20),    TFK(20),   THEV(20), XMJAT(20),   XNUI(64) TRA   23
      COMMON           BETA(64,20),    ELJ(10,20),  GFLJ(10,20),             TRA   24
     1 XNUIJ(64,20),XNUIJP(64,20)                                         TRA   25
      COMMON      IC,     IM,   INEQ, INEQV,     IP,   IRUN,    ISC, ISCP1,  TRA   26
     1   ISMC,ISMCNR,    ISR,    ISS, ISSNR,  ISSP1,  ISSP2,  ISSP3, ISSP4,  TRA   27
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,         TRA   28
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,    JJK,     LC,     M1,   NFIT,   TRA   29
     4    NIT,    NNN,    NNS,    NQS,    NQT, NTEST                      TRA   30
      COMMON    IGJ(20),    IGM(20),    IT8(5),    KUR(64,20),  LPIJ(20,10) TRA   31
      COMMON ACOM,ELMENT,HP                                               TRA   32
      COMMON /TRPROP/ VISC,PRF,SIGMA,FLEWIS                               TRA   33
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                                TRA   34
      COMMON /TRANS3/ B(20,2),BR(20,2),A(2),X(20),DH(20)                  TRA   35
      COMMON /TRANS5/ N,IELEC,ID1,ID2                                     TRA   36
      COMMON /BDTR/ ISWBD                                                 TRA   37
      COMMON /TEMPRY/ SAVEC(20)                                           TRA   38
      COMMON /BOUND/ LEWIS,IAMBIP                                         TRA   39
      COMMON /ERROR/ ERR                                                  TRA   40
      DIMENSION ZM1(20),C(20,20),ZK(2),CSAVE(20)                          TRA   41
      GO TO 30                                                            TRA   42
C     ENTRY TRANSX OF TRANSP DEFINES SPECIES AND INITIALIZES DATA FOR     TRA   43
C     TRANSPORT CALCULATIONS.                                             TRA   44
      ENTRY TRANSX                                                        TRA   45
C     N=ISS IS THE NUMBER OF SPECIES TO BE USED IN THE TRANSPORT          TRA   46
C     CALCULATIONS.                                                       TRA   47
      N=ISS                                                               TRA   48
      CALL XSECT                                                          TRA   49
      IF (ERR) RETURN                                                     TRA   50
C     CGI IS THE SPECIES MOLECULAR WEIGHT.                                TRA   51
      DO 10 I=1,N                                                         TRA   52
      ZM1(I)=2.8567*CGI(I)                                                TRA   53
10    ZM2(I)=.032064*CGI(I)                                               TRA   54
      DO 20 I=1,N                                                         TRA   55
```

```
      DO 20 J=I,N                                                       TRA   56
      Q(1,I,J)=ZM1(I)+ZM1(J)                                            TRA   57
      C(J,I)=(ZM1(I)-ZM1(J))/Q(1,I,J)                                   TRA   58
20    C(I,J)=SQRT(ZM1(I)*ZM1(J)/Q(1,I,J))                               TRA   59
      RETURN                                                            TRA   60
C     ENTRY TRANSP CALCULATES TRANSPORT PROPERTIES FOR TEMPERATURE T,   TRA   61
C     PRESSURE P, AND MOLE FRACTIONS X(J),                              TRA   62
C     NOTE - THE MOLE FRACTIONS ARE NOT NECESSARILY IN EQUILIBRIUM AT   TRA   63
C            THE TEMPERATURE T.                                         TRA   64
30    T=TTAB                                                            TRA   65
C     SAVE CT AND CCPJ(J) FOR LATER USE.                                TRA   66
      CTSAVE=CT                                                         TRA   67
      CT=T/CTAP                                                         TRA   68
      DO 40 J=1,N                                                       TRA   69
40    CSAVE(J)=CCPJ(J)                                                  TRA   70
C     CALCULATE SPECIES SPECIFIC HEATS AT TEMPERATURE T.                TRA   71
      CALL THERM1                                                       TRA   72
      CPTOT=0.                                                          TRA   73
      CMSAVE=0.                                                         TRA   74
      DO 50 J=1,N                                                       TRA   75
      X(J)=SAVEC(J)                                                     TRA   76
      DH(J)=CCPJ(J)-2.5                                                 TRA   77
      CPTOT=CCPJ(J)*X(J)+CPTOT                                          TRA   78
      CMSAVE=CGI(J)*X(J)+CMSAVE                                         TRA   79
50    CCPJ(J)=CSAVE(J)                                                  TRA   80
      CT=CTSAVE                                                         TRA   81
C     COMPUTE COLLISION CROSS SECTIONS.  STORE IN Q ARRAY.              TRA   82
      CALL PUTQIN (X(1)*P)                                              TRA   83
C     COMPUTE MATRIX ELEMENTS FOR TRANSPORT CALCULATIONS.  STORE IN Q   TRA   84
C     ARRAY.                                                            TRA   85
      IF (ISW18.LT.0) RETURN                                            TRA   86
      DO 60 I=1,N                                                       TRA   87
      DO 60 J=I,N                                                       TRA   88
      Q(1,I,J)=C(I,J)*Q(1,I,J)                                          TRA   89
60    Q(2,I,J)=C(I,J)*Q(2,I,J)                                          TRA   90
      S=SQRT(T)                                                         TRA   91
      SSIG=1.3451E5/S                                                   TRA   92
      FLEWIS=CPTOT/Q(1,ID1,ID2)                                         TRA   93
      IF (N.EQ.1) GO TO 80                                              TRA   94
      DO 70 J=2,N                                                       TRA   95
      Q(3,J,J)=Q(2,J,J)                                                 TRA   96
      J1=J-1                                                            TRA   97
      DO 70 I=1,J1                                                      TRA   98
      Q(3,I,J)=3.4375*Q(1,I,J)-.75*C(I,J)*Q(3,I,J)                      TRA   99
      Q(1,J,I)=(1.-C(J,I)**2)*(Q(3,I,J)-Q(2,I,J))                       TRA  100
      Q(2,J,I)=C(J,I)*(3.75*Q(1,I,J)-Q(3,I,J))                          TRA  101
      Q(3,J,I)=Q(3,I,J)-Q(2,J,I)-Q(1,J,I)                               TRA  102
      Q(3,I,J)=Q(3,I,J)+Q(2,J,I)-Q(1,J,I)                               TRA  103
70    Q(2,J,I)=(1.6666667*Q(1,I,J)-Q(2,I,J))/(ZM2(I)+ZM2(J))            TRA  104
80    Q(3,1,1)=Q(2,1,1)                                                 TRA  105
C     COMPUTE TRANSPORT PROPERTIES.  VISC=VISCOSITY, PRF=FROZEN PRANDTL TRA  106
C     NUMBER, SIGMA=ELECTRICAL CONDUCTIVITY, FLEWIS=ATOM-MOLECULE LEWIS TRA  107
C     NUMBER.                                                           TRA  108
      DO 90 I=1,N                                                       TRA  109
      B(I,1)=X(I)*Q(1,I,I)                                              TRA  110
```

```
 90     B(I,2)=X(I)*Q(2,I,I)                                              TRA  111
        CALL ELCOND (SIGMA,SSIG)                                          TRA  112
        CALL KINT (ZKINT)                                                 TRA  113
        CALL KANDMU (ZK)                                                  TRA  114
        ZKINT=ZK(1)+ZKINT                                                 TRA  115
        VISC=S*ZK(2)*1.D-3                                                TRA  116
        FLEWIS=FLEWIS/ZKINT*IAMBIP                                        TRA  117
        PRF=8.3167*CPTOT*ZK(2)/ZKINT/CMSAVE                               TRA  118
        IF (ISW8B.EQ.0) RETURN                                            TRA  119
C       PRINT TRANSPORT PROPERTY VALUES ONLY IF ISW8B.NE.0.               TRA  120
        WRITE (6,100)                                                     TRA  121
        WRITE (6,110) T,P,VISC,PRF,SIGMA,FLEWIS,CPTOT,CMSAVE             TRA  122
        WRITE (6,110) (X(J),J=1,N)                                        TRA  123
        RETURN                                                            TRA  124
C                                                                         TRA  125
C                                                                         TRA  126
100     FORMAT (1H )                                                      TRA  127
110     FORMAT (1P12E11.3)                                                TRA  128
        END                                                               TRA  129
                                                                          TRA  130-
```

```
      SUBROUTINE WEDGE                                                  WED    1
      DOUBLE PRECISION ZETA,CAPGAM,ZZP,ZZPP,Z,AMD,AND,GAMD,FL           WED    2
      LOGICAL WEDGEM,AXISYM,FINAL                                       WED    3
      COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,     WED    4
     1     NWX,NANGLE,NRADLE,WEDGEM,AXISYM,ISW9B                        WED    5
      COMMON /WMOD/ TSF                                                 WED    6
      COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),  WED    7
     1     SW,RC,JDIM,IPCINT                                            WED    8
      COMMON /TRPROP/ VISC,PR,SIGMA,FLEWIS                              WED    9
      COMMON /OUTPUT/ FVOUT(35),GJMF(20)                                WED   10
      EQUIVALENCE (GAMMA,FVOUT(10)), (AM,FVOUT(8)), (VMUINF,FVOUT(14)), WED   11
     1     (TINF,FVOUT(2)), (RHOINF,FVOUT(5)),(UINF,FVOUT(7)),         WED   12
     2     (PINF,FVOUT(4)), (REPF,FVOUT(12)), (GMW,FVOUT(13))          WED   13
      DIMENSION XW(10),PWW(10),QWW(10),DELSTW(10),YS(10),ZTA(10),UN(2)  WED   14
      DIMENSION FL(7), APR(10), AST(2), ASTQ(10),CH(10)                 WED   15
      DIMENSION QFM(10)                                                 WED   16
      DIMENSION GFAC(2),GFAK(2)                                         WED   17
      DATA UN /'  ','UN'/, RNAME/5HWEDGE/, ZERO /0./, ANMIN /1.E-4/     WED   18
      DATA PI /3.1415927/, AST /' ','*'/, ALMIN /0.01/                  WED   19
      DATA           AE /1./, RGC /8.314E7/, SQTPI /1.7724539/          WED   20
      ITH=1                                                             WED   21
      IAMAX=100                                                         WED   22
      ISW=IABS(ISW9B)                                                   WED   23
      HO=FVOUT(3)/1.8+0.5*(FVOUT(7)*30.48)**2/4.186E7                   WED   24
      HW=CPWALL*TWEDGE                                                  WED   25
C     HO AND HW ARE IN CAL/GM                                           WED   26
      HRATIO=HW/HO                                                      WED   27
      TRATIO=TWEDGE/TINF                                                WED   28
      S=AM*SQRT(0.5*GAMMA)                                              WED   29
      RU2=RHOINF*UINF**2                                                WED   30
      TREF=TSF*(1.+3.*HRATIO)/6.                                        WED   31
      CALL TRANSP (TREF,PINF)                                           WED   32
      C=VISC/VMUINF*TINF/TREF*0.067197                                  WED   33
C     THE NUMERICAL FACTOR CONVERTS VISC FROM GM/CM-SEC TO LBM/FT-SEC   WED   34
      TAU=0.664+1.73*HRATIO                                             WED   35
      A=(GAMMA+1.)/2.                                                   WED   36
      A2=A*A                                                            WED   37
      A4=A2*A2                                                          WED   38
      A5=A4*A                                                           WED   39
      A7=A2*A5                                                          WED   40
      G2=GAMMA*GAMMA                                                    WED   41
      G3=GAMMA*G2                                                       WED   42
      WK2=WK*WK                                                         WED   43
      WK3=WK2*WK                                                        WED   44
      WK4=WK2*WK2                                                       WED   45
      EPS=(GAMMA-1.)/(GAMMA+1.)                                         WED   46
      EPS2=EPS*EPS                                                      WED   47
      QFMC1=0.03886*PINF*SQRT(RGC*TINF/(2.*PI*GMW))*AE                  WED   48
      QFMC2=S**2+(GAMMA-0.5*(GAMMA+1.)*TRATIO)/(GAMMA-1.)               WED   49
     /FAC=16.*A7/G3*EPS2/WK4                                            WED   50
      GFC=WK/(2.*A2*EPS)                                                WED   51
      AM2=AM*AM                                                         WED   52
      PFAC=A5/GAMMA*4.*EPS2*AM2/WK2                                     WED   53
      PFAC=G2/A5*WK3/8./EPS                                             WED   54
      CHOFAC=0.332*A*EPS                                                WED   55
```

```
      YSFAC=0.125*G2*WK3/A4/EPS                                          WED   56
      AM=533.*C*AM2/REPF                                                 WED   57
      XS=12.*C*(EPS*TAU*AM*AM2)**2/REPF                                  WED   58
10    WRITE (6,260) UN(ITH)                                              WED   59
      WRITE (6,270) XM                                                   WED   60
      IF (ITH.EQ.2) WRITE (6,280) XS                                     WED   61
      ITHS=ITH                                                           WED   62
      GFAC(1)=GFC/AM                                                     WED   63
      GFAC(2)=GAMMA*GFC                                                  WED   64
      DO 240 IR=1,NRADLE                                                 WED   65
      TN=2.*RADLE(IR)                                                    WED   66
      IF (TN.EQ.0.) TN=1.E-6                                             WED   67
      EN=AM*SQRT(C/(REPF*TN/12.))                                        WED   68
      TAUN=TAU*EN                                                        WED   69
      TAUN2=TAUN*TAUN                                                    WED   70
      TAUN3=TAUN*TAUN2                                                   WED   71
      TAUN4=TAUN2*TAUN2                                                  WED   72
      TAUN6=TAUN3*TAUN3                                                  WED   73
      CHD=CHDFAC*TAUN*EN*TN                                              WED   74
      DPEAK=DFAC*TN/TAUN4                                                WED   75
      ZPEAK=ZFAC*TAUN6/TN                                                WED   76
      YSPEAK=YSFAC*TN/TAUN4                                              WED   77
      GPEAK(1)=GFAC(1)/TAUN2                                             WED   78
      GPEAK(2)=GFAC(2)/TAUN2                                             WED   79
      PPEAK=PFAC*TAUN4                                                   WED   80
      OMEGA=1./PPEAK                                                     WED   81
      DO 230 IA=1,NANGLE                                                 WED   82
      IF (IR.GT.1.OR.ITH.EQ.2) GO TO 30                                  WED   83
      AMD=AM                                                             WED   84
      AND=AMAX1(ANGLE(     ANMIN)                                        WED   85
      GAMD=GAMMA                                                         WED   86
      CALL SHOCK (AMD,AND,GAMD,FL,IERR)                                  WED   87
      IF (IERR.NE.0) GO TO 20                                            WED   88
      IAMAX=IA                                                           WED   89
      GO TO 30                                                           WED   90
20    APR(IA)=FL(4)                                                      WED   91
      IF (IERR.EQ.1) GO TO 30                                            WED   92
      CALL DUMP (RNAME)                                                  WED   93
      RETURN                                                             WED   94
30    IF (IA.GE.IAMAX) ITH=2                                             WED   95
      FINAL=.FALSE.                                                      WED   96
      ALPHA=ANGLE(IA)/57.2958                                            WED   97
      IFM=0                                                              WED   98
      IF (ITH.EQ.1) GO TO 40                                             WED   99
      CAPGAM=GPEAK(2)*ALPHA                                              WED  100
      GO TO 50                                                           WED  101
40    CAPGAM=GPEAK(1)*SQRT(GAMMA/A*AMAX1(APR(IA)-1.,ZERO))               WED  102
      IF (IR.GT.1) GO TO 50                                              WED  103
      SSA=S*SIN(AMAX1(ALPHA,ALMIN))                                      WED  104
      SSASQ=SSA**2                                                       WED  105
      EXPV=EXP(-SSASQ)                                                   WED  106
      ERFV=1.+ERF(SSA)                                                   WED  107
      QFM(IA)=QFMC1*(QFMC2*(EXPV+SQTPI*SSA*ERFV)-0.5*EXPV)               WED  108
50    IT=2                                                               WED  109
      NM1=1                                                              WED  110
```

```
      NW2=1                                                             WED 111
      ITAB=1                                                            WED 112
60    IT=IT+1                                                           WED 113
      IF (NW1.GT.NWX) GO TO 70                                          WED 114
      WXV1=WX1+(NW1-1)*DWX                                              WED 115
      GO TO 80                                                          WED 116
70    WXV1=1.E30                                                        WED 117
80    IF (NW2.GT.20) GO TO 90                                           WED 118
      WXV2=WX1(NW2)                                                     WED 119
      GO TO 100                                                         WED 120
90    WXV2=1.E30                                                        WED 121
100   IF (WXV1-WXV2) 110,110,120                                        WED 122
110   IF (WXV1.GT.1.E19) GO TO 130                                      WED 123
      XW(IT)=WXV1                                                       WED 124
      NW1=NW1+1                                                         WED 125
      GO TO 140                                                         WED 126
120   IF (WXV2.GT.1.E19) GO TO 130                                      WED 127
      XW(IT)=WXV2                                                       WED 128
      NW2=NW2+1                                                         WED 129
      GO TO 140                                                         WED 130
130   IF (IT.LE.1) GO TO 230                                            WED 131
      IT=IT-1                                                           WED 132
      FINAL=.TRUE.                                                      WED 133
      GO TO 120                                                         WED 134
140   XC=XW(IT)*COS(ALPHA)                                              WED 135
      ZETA=ZETA*XC                                                      WED 136
      ZTA(IT)=ZETA                                                      WED 137
      CALL WESOLN (ZETA,CAPGAM,ZZP,ZZPP,Z)                              WED 138
      ZZPS=ZZP                                                          WED 139
      IF (ITH.EQ.1) GO TO 150                                           WED 140
C     CHENG-KEMP THEORY                                                 WED 141
      YS(IT)=YSPAK*Z                                                    WED 142
      PRW(IT)=PINF*PPAK*ZZPP                                            WED 143
      SQT=SQRT(ZZPS)                                                    WED 144
      DELSTW(IT)=DPAK*SQT/ZZPP                                          WED 145
      GO TO 160                                                         WED 146
C     MODIFIED CHENG-KEMP THEORY                                        WED 147
150   SQT=SQRT(ZZPS+OMEGA*ZTA(IT))                                      WED 148
      DELSTW(IT)=DPAK*SQT/(ZZPP+OMEGA)                                  WED 149
      PRATIO=PPAK*ZZPP+1.                                               WED 150
      PRW(IT)=PINF*PRATIO                                               WED 151
      YS(IT)=YSPAK*(1.+SQT)/(ZZPP+OMEGA)+A*XC*TAN(ALPHA)                WED 152
160   CH(IT)=CHO/DELSTW(IT)                                             WED 153
      QWW(IT)=CH(IT)*RHOINF*UINF*(HO-HW)*1.8                            WED 154
      ASTQ(IT)=AST(1)                                                   WED 155
      IF (ITH.EQ.2) GO TO 170                                           WED 156
      IF (QWW(IT).LT.QFM(IA)) GO TO 170                                 WED 157
      IFM=1                                                             WED 158
      QWW(IT)=QFM(IA)                                                   WED 159
      ASTQ(IT)=AST(2)                                                   WED 160
170   IF (IT.LT.10) GO TO 60                                            WED 161
180   IF (ITAB.EQ.1) GO TO 190                                          WED 162
      WRITE (6,290)                                                     WED 163
      GO TO 200                                                         WED 164
190   CPGV=CAPGAM                                                       WED 165
```

```
      WRITE (6,300) RADLE(IR),ANGLE(IA),CPGM,   EGA                          WED 166
      WRITE (6,310) (XW(J),J=1,IT)                                           WED 167
200   WRITE (6,320) (PWW(J),J=1,IT)                                          WED 168
      WRITE (6,330) (QWW(J),ASTQ(J),J=1,IT)                                  WED 169
      WRITE (6,340) (CH(J),J=1,IT)                                           WED 170
      WRITE (6,350) (DELSTW(J),J=1,IT)                                       WED 171
      IF (IS .EQ.0) GO TO 220                                                WED 172
      IF ( S  EQ.2) GO TO 210                                                WED 173
      WRITE (6,360) (YS(J),J=1,IT)                                           WED 174
      IF (IS9.EQ.1) GO TO 220                                                WED 175
210   WRITE (6,370) (ZTA(J),J=1,IT)                                          WED 176
220   IF (IPM.NE.0) WRITE (6,380)                                            WED 177
      IF (FINAL) GO TO 230                                                   WED 178
      IT=0                                                                   WED 179
      ITAB=ITAB+1                                                            WED 180
      GO TO 60                                                               WED 181
230   CONTINUE                                                               WED 182
      ITH=ITHS                                                               WED 183
240   CONTINUE                                                               WED 184
      IF (ITH.EQ.2.OR.IS9.EQ.ISW9B) GO TO 250                                WED 185
      ITH=2                                                                  WED 186
      GO TO 10                                                               WED 187
250   RETURN                                                                 WED 188
C                                                                            WED 189
C                                                                            WED 190
C                                                                            WED 191
260   FORMAT (1HC,43X,46H**********CONDITIONS ON WEDGE MODELS**********/WED 192
     152X,A2,26HMODIFIED CHENG-KEMP THEORY)                                  WED 193
270   FORMAT (45H MERGING EFFECTS SIGNIFICANT FOR XW LESS THAN,1PE10.2,8WED 194
     1H INCHES.)                                                            WED 195
280   FORMAT (65H STRONG-INTERACTION APPROXIMATION BREAKS DOWN FOR XW GRWED 196
     1EATER THAN,1PE10.2,8H INCHES.)                                        WED 197
290   FORMAT (1H0)                                                           WED 198
300   FORMAT (31H0*********LEADING-EDGE RADIUS =,F6.3,25H INCH,  ANGLE OWED 199
     1F ATTACK =,F5.1,26H DEGREES,  CAPITAL GAMMA =,1PE10.2,9H, OMEGA =,WED 200
     2E10.2,9H*********)                                                    WED 201
310   FORMAT (12H XW (INCHES),9X,1H=,1P10E11.2)                              WED 202
320   FORMAT (10H PWW (ATM),11X,1H=,1P10E11.2)                               WED 203
330   FORMAT (22H QWW (BTU/SQ FT-SEC) =,1X,10(1PE10.2,A1))                  WED 204
340   FORMAT (15H STANTON NUMBER,6X,1H=,1P10E11.2)                           WED 205
350   FORMAT (14H DELSTW (INCH),7X,1H=,1P10E11.2)                            WED 206
360   FORMAT (12H YS (INCHES),9X,1H=,1P10E11.2)                              WED 207
370   FORMAT (5H ZETA,16X,1H=,1P10E11.2)                                     WED 208
380   FORMAT (5X,22H*  FREE-MOLECULE LIMIT)                                  WED 209
      END                                                                    WED 210-
```

```
      SUBROUTINE WESOLN (ZETA,CAPGAM,ZZP,ZZPP)                          WES    1
C     SOLUTION OF THE CHENG EQUATION FOR FLOW OVER A WEDGE              WES    2
      IMPLICIT REAL*8 (A-H,O-Z)                                         WES    3
C     KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC 1103 VERSION           WES    4
      LOGICAL CALLED                                                    WES    5
      DATA CALLED /.FALSE./                                             WES    6
      DATA B0,B1/1.81712059D0,1.73205080D0/                             WES    7
      NAMELIST/WEDMP/ XL,SQTL,Q1,Q2,Q3,Q4,Q5,ZT,  DLDZT                 WES    8
      IF (CALLED) GO TO 10                                              WES    9
      CALLED=.TRUE.                                                     WES   10
      C1=1.00                                                           WES   11
      C3=1.00/3.00                                                      WES   12
      C229=20.00/9.00                                                   WES   13
      C9=9.00                                                           WES   14
      C463=46.00/3.00                                                   WES   15
      C103=10.00/3.00                                                   WES   16
      C4=4.00                                                           WES   17
      C32=1.500                                                         WES   18
      C2=2.00                                                           WES   19
      C769=76.00/9.00                                                   WES   20
10    P1=ZETA**C3                                                       WES   21
      P2=DSQRT(ZETA)                                                    WES   22
      IF (ZETA.GT.1.D-8) GO TO 20                                       WES   23
C     ROGER'S ANALYTICAL SOLUTION FOR SMALL ZETA                        WES   24
      P12=P1**2                                                         WES   25
      Z=1.08096*P12+0.50869*P1*P2-0.0249*ZETA                           WES   26
      XL=1.81711*P1+1.25975*P2+0.14712*P12                              WES   27
      DLDZT=0.6057/P12+0.62988/P2+0.09808/P1                            WES   28
      GO TO 50                                                          WES   29
C     ITERATIVE SOLUTION BASED ON CHENG/KEMP EXACT SOLUTION FOR GAMMA=0.WES   30
20    XL=B0*P1+B1*P2                                                    WES   31
      N=0                                                               WES   32
30    IF (N.GT.20) GO TO 40                                             WES   33
      SQTL=DSQRT(XL)                                                    WES   34
      Q1=C1+SQTL                                                        WES   35
      Q2=Q1*Q1                                                          WES   36
      Q3=Q1*Q2                                                          WES   37
      Q4=Q2*Q2                                                          WES   38
      Q5=DLOG(Q1)                                                       WES   39
      ZT=C3*Q4-C229*Q3+C9*Q2-C463*Q1+Q5*(C103-C4*SQTL+C2*Q5)+C769       WES   40
      Z=C2*(SQTL-XL/C2+C3*XL**C32-Q5)                                   WES   41
      DLDZT=Q1/Z                                                        WES   42
      IF (DABS(ZT-ZETA)/ZETA.LE.1.D-4) GO TO 50                         WES   43
      XL=XL-(ZT-ZETA)*DLDZT                                             WES   44
      N=N+1                                                             WES   45
      GO TO 30                                                          WES   46
40    WRITE (6,60) ZETA,CAPGAM                                          WES   47
      WRITE (6,WEDMP)                                                   WES   48
50    CGSQ=CAPGAM**2                                                    WES   49
      ZZP=XL+CGSQ*ZETA                                                  WES   50
      ZZPP=DLDZT+CGSQ                                                   WES   51
      Z=DSQRT(Z**2+(CAPGAM*ZETA)**2)                                    WES   52
      RETURN                                                            WES   53
C                                                                       WES   54
60    FORMAT (30H0TOO MANY ITERATIONS IN WESOLN,10X,5HZETA=,1PE15.8,10X,WES   55
```

```
      WES  56
      WES  57-
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      THCAPGAM=.E15.8)
      END
```

```
      SUBROUTINE XSECT                                                      XSE   1
C     SUBROUTINE XSECT EDITS INPUT CROSS SECTION DATA FOR USE IN            XSE   2
C     TRANSPORT CALCULATIONS.                                               XSE   3
      LOGICAL ERR                                                           XSE   4
      REAL ACOM(30),ELMENT(10),HP(20)                                       XSE   5
      DOUBLE PRECISION AA,CAPX,CDIJ                                         XSE   6
      COMMON AA(22,24),CDIJ(20,10),CAPX(20)                                 XSE   7
      COMMON       A, AFNTS,  AFNX, AMACH,    AR,  ARBA,  ARBB, BZERO,     XSE   8
     1       C,  CARB,    CH,   CHA,  CLNT,    CM,   CMA,   CRA,   CRP,    XSE   9
     2    CRRB,   CRS,  CSTA,    CT,  CTAP,   CTB,   CTC, CTMAX, CTMXX,    XSE  10
     3     CTP,  CTPL,   CTT,    CX,   CXB, CXMAX,DATFST,DBTEST, DELT1,    XSE  11
     4  DELT2,DELTAX, DLOGA, DLOGR,    DT,   ENT,  FLUX, HDELX,   PCT,    XSE  12
     5PCTEST,  PRES, PRESA, PRESB,PRESTH,  PRHC,  RHAP,   RHO,  RHOB,    XSE  13
     6RHOBAR,  RHOC,  RHOP,  RHPL,  RHTH,ROBARA,ROBARP,  SCPG,   SDT,    XSE  14
     7    SEN,  SHPG,    SC,    SL,  SL64,    SM,   SU,   SU2,  SUMG,    XSE  15
     8   TEST, TESTB,TPRINT, TSTOP,    UP,    ZP,   ZPA                    XSE  16
      COMMON          BE(64),   BET(20),  BLBK(31),   CAI(64),CAPXTH(20),XSE  17
     1  CCPJ(20), CEACT(64),   CGI(20),  CGMU(20),   CHI(64),  CHII(20),XSE  18
     2CLNIMC(64), CLNPI(64),   CMW(20),  ETAI(64),  ETAJ(20),   GJA(20),XSE  19
     3   GJB(20),PERTGJ(20),   PGJ(20),    PI(64), PICHI(64),    QM(20),XSE  20
     4    QQ(64),   SAJ(20), SDCHI(64),  SENT(20),   SHJ(20),  SHJA(20),XSE  21
     5  SKIL(20),    SS(20),    TB(30),   TFA(20),   TFB(20),   TFC(20),XSE  22
     6   TFD(20),   TFE(20),   TFK(20),  THEV(20), XMJAT(20),  XNUI(64) XSE  23
      COMMON          BETA(64,20),   ELJ(10,20),  GELJ(10,20),              XSE  24
     1 XNUIJ(64,20),XNUIJP(64,20)                                           XSE  25
      COMMON       IC,    IM,  INEQ, INEQV,    IP,  IRUN,   ISC, ISCP1,   XSE  26
     1  ISMC,ISMCNR,   ISR,   ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,     XSE  27
     2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,     XSE  28
     3 ISW5B, ISW6A, ISW6B,  IUPD, IZERO,   JJK,    LC,    M1,  NFIT,    XSE  29
     4   NIT,   NNN,   NNS,   NQS,   NQT, NTEST                            XSE  30
      COMMON   IGJ(20),  IGM(20),   ITB(5),   KUR(64,20),  LPIJ(20,10)     XSE  31
      COMMON ACOM,ELMENT,HP                                                 XSE  32
      DIMENSION VCOUL(2,2,2,2)                                              XSE  33
      DIMENSION IZ(20)                                                      XSE  34
      COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)                                  XSE  35
      COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,I(50)                XSE  36
      COMMON /TRANI/ IIQ(5,100)                                             XSE  37
      COMMON /TRANJ/ JJQ(5,100)                                             XSE  38
      COMMON /TRANV/ VV(5,100)                                              XSE  39
      COMMON /TRANS5/       LSS,IELEC,ID1,ID2                               XSE  40
      COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ            XSE  41
      COMMON /TRANS8/ NV(15),N                                              XSE  42
      COMMON /LN/ IATOM,IMOL,JATOM,JMOL                                     XSE  43
      COMMON /MASSFL/ SMASS,CTMXXI,TSTOPI,IS(20)                            XSE  44
      COMMON /ERROR/ ERR                                                    XSE  45
      NAMELIST /XSDMP/ T,Q,ZM2,I,LSS,IELEC,ID1,ID2,V,KQ,NQ,IQ,JQ,          XSE  46
     1     NKQ,NV,N                                                         XSE  47
      DATA VCOUL/.994,1.71,3.41,2.01,3.41,2.01,16.,1.71,.994,1.62,2*0.,     XSE  48
     1 3.41,1.00,0.,.0./                                                    XSE  49
      DATA IZ /20*0./                                                       XSE  50
      N=ISS                                                                 XSE  51
      IELEC=IS(1)                                                           XSE  52
      IF (IELEC.NE.1) IELEC=0                                               XSE  53
C     FIND CORRESPONDENCE BETWEEN INPUT SPECIES AND CROSS SECTION DATA.     XSE  54
      JJ=50                                                                 XSE  55
```

```
      DO 30 K=1,50                                                      XSE  56
      J=51-K                                                            XSE  57
      DO 10 L=1,N                                                       XSE  58
      IF (IS(L)-J) 10,20,10                                             XSE  59
10    CONTINUE                                                          XSE  60
      I(J)=JJ                                                           XSE  61
      JJ=JJ-1                                                           XSE  62
      GO TO 30                                                          XSE  63
20    I(J)=L                                                            XSE  64
30    CONTINUE                                                          XSE  65
C     FIND INDICES OF SPECIES TO BE USED IN LEWIS NUMBER CALCULATION.   XSE  66
      ID1=MIN0(IATOM,IMOL)                                              XSE  67
      ID2=MAX0(IATOM,IMOL)                                              XSE  68
      CALL CXSECT (ID1,ID2)                                             XSE  69
C     RE-INDEX CROSS SECTION DATA TO CORRESPOND TO ORDER OF INPUT       XSE  70
C     SPECIES.  SET Q(1,L,J)=1.0 FOR ALL PAIRS OF SPECIES FOR WHICH     XSE  71
C     CROSS SECTION DATA ARE PROVIDED.                                  XSE  72
40    DO 50 J=1,N                                                       XSE  73
      DO 50 L=1,N                                                       XSE  74
50    Q(1,J,L)=0.                                                       XSE  75
      Q(1,2,1)=1.                                                       XSE  76
      NKQ=MIN0(NNKQ,99)                                                 XSE  77
      LQ=0                                                              XSE  78
      V(1)=1.                                                           XSE  79
      KV=1                                                              XSE  80
      KIJ=0                                                             XSE  81
      DO 90 K=1,NKQ                                                     XSE  82
      M=ISEQ(K)                                                         XSE  83
      NK=NKQ(M)                                                         XSE  84
      KQ(K)=NK                                                          XSE  85
      NK=NV(NK)                                                         XSE  86
      IF (NK.EQ.0) GO TO 70                                             XSE  87
      DO 60 KM=1,NK                                                     XSE  88
      KV=KV+1                                                           XSE  89
60    V(KV)=V(KM,M)                                                     XSE  90
70    CONTINUE                                                          XSE  91
      NK=NNQ(M)                                                         XSE  92
      LQ=LQ+NK                                                          XSE  93
      NQ(K)=LQ                                                          XSE  94
      IF (NK.EQ.0) GO TO 90                                             XSE  95
      DO 80 KM=1,NK                                                     XSE  96
      KIJ=KIJ+1                                                         XSE  97
      L=IQ(KM,M)                                                        XSE  98
      J=JQ(KM,M)                                                        XSE  99
      CALL CXSECT (L,J)                                                 XSE 100
      IF (J.LE.N.AND.L.LE.N) Q(1,L,J)=1.                                XSE 101
      IQ(KIJ)=L                                                         XSE 102
80    JQ(KIJ)=J                                                         XSE 103
90    CONTINUE                                                          XSE 104
C     FOR KQ=9 OR 12, THE V-ARRAY DEPENDS ON THE INDEXING OF THE        XSE 105
C     SPECIES.  THESE DATA MUST ALSO BE RE-INDEXED.                     XSE 106
      L=0                                                               XSE 107
100   CALL CXSECT (L,VMV,LL,J)                                          XSE 108
      IF (L.EQ.0) GO TO 110                                             XSE 109
      CALL CXSECT (LL,J)                                                XSE 110
```

```
      IF (J.LE.N.AND.LL.LE.N) Q(1,LL,J)=1.                                  XSE  111
      GO TO 100                                                             XSE  112
C     EDIT CROSS SECTION DATA TO DELETE SPECIES NOT REQUIRED FOR            XSE  113
C     TRANSPORT CALCULATIONS.                                               XSE  114
110   LQ1=1                                                                 XSE  115
C     FOR KQ=9 OR 12, IT MAY BE NECESSARY TO COMPUTE CROSS SECTIONS FOR     XSE  116
C     SPECIES NOT INCLUDED IN TRANSPORT CALCULATIONS.  STATEMENTS 120 TO    XSE  117
C     140 CHECK FOR THIS POSSIBILITY. IF SUCH CROSS SECTIONS ARE FOUND,     XSE  118
C     STATEMENTS 140 TO 300 REVISE THE DATA TO ELIMINATE THEM FROM THE      XSE  119
C     CALCULATIONS, STATEMENTS 300 THROUGH 380 THEN PROCEED TO EDIT THE     XSE  120
C     CROSS SECTION DATA TO REMOVE SPECIES NOT REQUIRED FOR THE             XSE  121
C     TRANSPORT CALCULATIONS.                                               XSE  122
120   CALL BXSECT (L,MV,LL,J)                                               XSE  123
      IF (L.EQ.0) GO TO 300                                                 XSE  124
      IF (J.LE.N.AND.LL.LE.N) GO TO 120                                     XSE  125
      IF (L.GT.1) LQ1=NQ(L-1)+1                                             XSE  126
      LQ2=NQ(L)                                                             XSE  127
      IF (LQ1.GT.LQ2) GO TO 120                                             XSE  128
      DO 130 LQ=LQ1,LQ2                                                     XSE  129
      IF (JQ(LQ).LE.N.AND.IQ(LQ).LE.N) GO TO 140                            XSE  130
130   CONTINUE                                                              XSE  131
      GO TO 120                                                             XSE  132
C     CROSS SECTIONS ARE REQUIRED FOR THE PAIR (LL,J).  FIND UNUSED         XSE  133
C     STORAGE LOCATION FOR THEM.                                            XSE  134
140   DO 150 II=2,N                                                         XSE  135
      II1=II-1                                                              XSE  136
      DO 150 JJ=1,II1                                                       XSE  137
      IF (Q(1,II,JJ).EQ.0.) GO TO 190                                       XSE  138
150   CONTINUE                                                              XSE  139
C     NO STORAGE SPACE AVAILABLE.  ADD AN ADDITIONAL SPECIES TO CROSS       XSE  140
C     SECTION COMPUTATIONS.                                                 XSE  141
      IF (N.EQ.20) GO TO 540                                                XSE  142
      N=N+1                                                                 XSE  143
      JJ=MAX0(J,LL)                                                         XSE  144
      DO 160 J=1,50                                                         XSE  145
      IF (I(J).EQ.N) GO TO 170                                              XSE  146
160   CONTINUE                                                              XSE  147
170   DO 180 K=1,50                                                         XSE  148
      IF (I(K).NE.JJ) GO TO 180                                             XSE  149
      I(J)=JJ                                                               XSE  150
      I(K)=N                                                                XSE  151
      GO TO 40                                                              XSE  152
180   CONTINUE                                                              XSE  153
      GO TO 540                                                             XSE  154
C     STORE CROSS SECTIONS FOR PAIR (LL,J) IN LOCATION (II,JJ).             XSE  155
190   Q(1,II,JJ)=1.                                                         XSE  156
C     IT IS FIRST NECESSARY TO CONVERT ANY KQ=10 STEPS FOR THE PAIR TO      XSE  157
C     KQ=12.                                                                XSE  158
      NSV=NKQ                                                               XSE  159
      L=0                                                                   XSE  160
      LQ1=0                                                                 XSE  161
200   L=L+1                                                                 XSE  162
      IF (KQ(L).NE.10) GO TO 270                                            XSE  163
      LQ2=NQ(L)                                                             XSE  164
      IF (L.GT.1) LQ1=NQ(L-1)                                               XSE  165
```

```
      IF (LQ1.GE.LQ2) GO TO 270                                          XSE 166
      LQ=LQ1                                                             XSE 167
  210 LQ=LQ+1                                                            XSE 168
      IV=IQ(LQ)                                                          XSE 169
      JV=JQ(LQ)                                                          XSE 170
      IF ((LL.NE.J.AND.(IV.NE.LL.OR.JV.NE.J)).OR.(IV.NE.LL.AND.JV.NE.J))XSE 171
     1 GO TO 260                                                         XSE 172
C     CONVERSION TO KQ=12 MUST BE MADE FOR CURRENT STEP.                 XSE 173
C     FIND LOCATION MM AT WHICH NEW DATA ARE TO BE INSERTED IN V-ARRAY.  XSE 174
      NKQ=L                                                              XSE 175
      LV=0                                                               XSE 176
  220 CALL BXSECT (LV,MM,K,L)                                            XSE 177
      IF (LV.NE.0) GO TO 220                                             XSE 178
C     REVISE CROSS SECTION DATA                                          XSE 179
      MV=MMV                                                             XSE 180
      DO 230 K=MM,MMV                                                    XSE 181
      V(MV+4)=V(MV)                                                      XSE 182
  230 MV=MV-1                                                            XSE 183
      V(MV+2)=IV                                                         XSE 184
      V(MV+3)=V(MV+2)                                                    XSE 185
      V(MV+4)=JV                                                         XSE 186
      V(MV+5)=V(MV+4)                                                    XSE 187
      MMV=MMV+4                                                          XSE 188
      LQ1=LQ1+1                                                          XSE 189
      LV=NSV                                                             XSE 190
      DO 240 K=NKQ,NSV                                                   XSE 191
      KQ(LV+1)=KQ(LV)                                                    XSE 192
      NQ(LV+1)=NQ(LV)                                                    XSE 193
  240 LV=LV-1                                                            XSE 194
      NSV=NSV+1                                                          XSE 195
      NQ(LV+1)=LQ1                                                       XSE 196
      L=LV+2                                                             XSE 197
      LV=LQ                                                              XSE 198
      DO 250 K=LQ1,LQ                                                    XSE 199
      LV=LV-1                                                            XSE 200
      IQ(LV+1)=IQ(LV)                                                    XSE 201
  250 JQ(LV+1)=JQ(LV)                                                    XSE 202
      IQ(LQ1)=IV                                                         XSE 203
      JQ(LQ1)=JV                                                         XSE 204
      KQ(NKQ)=12                                                         XSE 205
  260 CONTINUE                                                           XSE 206
      IF (LQ.LT.LQ2) GO TO 210                                           XSE 207
  270 CONTINUE                                                           XSE 208
      IF (L.LT.NSV) GO TO 200                                            XSE 209
      NKQ=NSV                                                            XSE 210
C     CONVERSION OF KQ=10 TO KQ=12 COMPLETED.  NOW REVISE DATA TO CHANGE XSE 211
C     PAIR (LL,J) TO (II,JJ).                                            XSE 212
      LQ2=NQ(NKQ)                                                        XSE 213
      DO 280 LQ=1,LQ2                                                    XSE 214
      IF (IQ(LQ).NE.LL.OR.JQ(LQ).NE.J) GO TO 280                         XSE 215
      IQ(LQ)=II                                                          XSE 216
      JQ(LQ)=JJ                                                          XSE 217
  280 CONTINUE                                                           XSE 218
      L=0                                                                XSE 219
  290 CALL BXSECT (L,MV,IV,JV)                                           XSE 220
```

```
      IF (L.EQ.0) GO TO 110                                              XSE 221
      IF (IV.NE.LL.OR.JV.NE.J) GO TO 290                                 XSE 222
      IV=II                                                              XSE 223
      JV=JJ                                                              XSE 224
      GO TO 290                                                          XSE 225
C     EDIT CROSS SECTION DATA TO ELIMINATE SPECIES NOT REQUIRED IN       XSE 226
C     COMPUTATIONS.                                                      XSE 227
300   DO 310 LQ=1,400                                                    XSE 228
      IF (IQ(LQ).GT.N.OR.JQ(LQ).GT.N) IQ(LQ)=0                           XSE 229
310   CONTINUE                                                           XSE 230
      NEWL=0                                                             XSE 231
      NEWLQ=0                                                            XSE 232
      NEWMV=1                                                            XSE 233
      MV1=2                                                              XSE 234
      NSV=NKQ                                                            XSE 235
      LQ1=1                                                              XSE 236
      KQ(100)=0                                                          XSE 237
      DO 370 L=1,NSV                                                     XSE 238
      K=KQ(L)                                                            XSE 239
      LQ2=NQ(L)                                                          XSE 240
      IF (LQ2.LT.LQ1) GO TO 330                                          XSE 241
      DO 320 LQ=LQ1,LQ2                                                  XSE 242
      IF (IQ(LQ).NE.0) NEWLQ=NEWLQ+1                                     XSE 243
320   CONTINUE                                                           XSE 244
330   LQ1=LQ2+1                                                          XSE 245
      IF (K.EQ.10) GO TO 350                                             XSE 246
      IF (NEWLQ.EQ.NQ(NEWL)) GO TO 360                                   XSE 247
      MV2=MV1+NV(K)-1                                                    XSE 248
      DO 340 MV=MV1,MV2                                                  XSE 249
      NEWMV=NEWMV+1                                                      XSE 250
340   V(NEWMV)=V(MV)                                                     XSE 251
350   NEWL=NEWL+1                                                        XSE 252
      KQ(NEWL)=K                                                         XSE 253
      NQ(NEWL)=NEWLQ                                                     XSE 254
      GO TO 370                                                          XSE 255
360   NKQ=NKQ-1                                                          XSE 256
370   MV1=MV1+NV(K)                                                      XSE 257
      LL=0                                                               XSE 258
      DO 380 LQ=1,400                                                    XSE 259
      IF (IQ(LQ).EQ.0) GO TO 380                                         XSE 260
      LL=LL+1                                                            XSE 261
      IQ(LL)=IQ(LQ)                                                      XSE 262
      JQ(LL)=JQ(LQ)                                                      XSE 263
380   CONTINUE                                                           XSE 264
C     COMPUTE UNSPECIFIED NEUTRAL-NEUTRAL CROSS SECTIONS FROM EMPIRICAL  XSE 265
C     MIXING LAW.  COMPUTATIONS ARE PERFORMED AT LAST SPECIFIED STEP     XSE 266
C     WITH KQ=10, OR, IF NO STEPS WITH KQ=10 ARE SPECIFIED, AFTER ALL    XSE 267
C     SPECIFIED CROSS SECTIONS HAVE BEEN COMPUTED.                       XSE 268
      LL=NKQ+1                                                           XSE 269
      DO 390 L=1,NKQ                                                     XSE 270
      IF (KQ(L).EQ.10) LL=L                                              XSE 271
390   CONTINUE                                                           XSE 272
      IF (LL.LE.NKQ) GO TO 400                                           XSE 273
      KQ(LL)=10                                                          XSE 274
      NQ(LL)=NQ(LL-1)                                                    XSE 275
```

```
      NKQ=NKQ+1                                                          XSE 276
400   NN=0                                                               XSE 277
      DO 420 L=1,ISS                                                     XSE 278
      IF (IELEC.EQ.0) GO TO 410                                          XSE 279
      IZ(L)=LPIJ(L,1)                                                    XSE 280
      IF (IZ(L).NE.0) GO TO 420                                          XSE 281
410   NN=NN+1                                                            XSE 282
      I(NN)=L                                                            XSE 283
      IF (Q(I,L,L).EQ.0.) GO TO 540                                      XSE 284
420   CONTINUE                                                           XSE 285
      CALL AXSECT (LL,NN)                                                XSE 286
C     FOR IONIZED SPECIES, USE EFFECTIVE COULOMB CROSS SECTIONS CHOSEN   XSE 287
C     TO FIT TRANSPORT CALCULATIONS OF SPITZER,                          XSE 288
      IF (LL.EQ.NKQ.AND.NQ(LL).EQ.NQ(LL-1)) GO TO 430                    XSE 289
      NKQ=NKQ+1                                                          XSE 290
      NQ(NKQ)=NQ(NKQ-1)                                                  XSE 291
430   DO 480 J=1,2                                                       XSE 292
      IE=IELEC                                                           XSE 293
      KK=J                                                               XSE 294
      DO 470 JJ=1,2                                                      XSE 295
440   NN=0                                                               XSE 296
      DO 450 L=1,ISS                                                     XSE 297
      II=IABS(IZ(L))                                                     XSE 298
      IF (L.EQ.IE.OR.(II.NE.J.AND.II.NE.KK)) GO TO 450                   XSE 299
      NN=NN+1                                                            XSE 300
      I(NN)=L                                                            XSE 301
450   CONTINUE                                                           XSE 302
      CALL AXSECT (NKQ,NN)                                               XSE 303
      IF (NQ(NKQ).EQ.NQ(NKQ-1)) GO TO 460                                XSE 304
      KQ(NKQ)=2                                                          XSE 305
      NKQ=NKQ+1                                                          XSE 306
      NQ(NKQ)=NQ(NKQ-1)                                                  XSE 307
      V(NEWMV+1)=VCOUL(1,KK,J,JJ)                                        XSE 308
      V(NEWMV+2)=VCOUL(2,KK,J,JJ)                                        XSE 309
      NEWMV=NEWMV+2                                                      XSE 310
460   IF (KK.LE.1) GO TO 470                                             XSE 311
      KK=KK-1                                                            XSE 312
      GO TO 440                                                          XSE 313
470   IE=0                                                               XSE 314
480   CONTINUE                                                           XSE 315
      NN=0                                                               XSE 316
      DO 490 L=1,ISS                                                     XSE 317
      IF (IZ(L).EQ.0) GO TO 490                                          XSE 318
      NN=NN+1                                                            XSE 319
      I(NN)=L                                                            XSE 320
490   CONTINUE                                                           XSE 321
      CALL AXSECT (NKQ,NN)                                               XSE 322
      IF (NQ(NKQ).NE.NQ(NKQ-1)) GO TO 540                                XSE 323
C     SET UNSPECIFIED ION-NEUTRAL CROSS SECTIONS EQUAL TO TYPICAL VALUE  XSE 324
C     ESTIMATED FOR N-O+ INTERACTION.                                    XSE 325
      DO 500 L=1,ISS                                                     XSE 326
500   I(L)=L                                                             XSE 327
      CALL AXSECT (NKQ,ISS)                                              XSE 328
      IF (NQ(NKQ-1).NE.NQ(NKQ)) GO TO 510                                XSE 329
      NKQ=NKQ-1                                                          XSE 330
```

```
      GO TO 520                                                          XSE 331
510   K2(NKQ)=6                                                          XSE 332
      V(NEWMV+1)=996.                                                    XSE 333
      V(NEWMV+2)=5.                                                      XSE 334
520   CONTINUE                                                           XSE 335
      IF (NKQ.GT.100) GO TO 540                                          XSE 336
530   RETURN                                                             XSE 337
C     INSUFFICIENT CROSS SECTION DATA AVAILABLE.  TERMINATE CASE.        XSE 338
540   WRITE (6,550)                                                      XSE 339
      ERR=.TRUE.                                                         XSE 340
      WRITE (6,XSDMP)                                                    XSE 341
      GO TO 530                                                          XSE 342
C                                                                        XSE 343
C                                                                        XSE 344
C                                                                        XSE 345
550   FORMAT (/116HOTRANSPORT PROPERTIES OF DESIRED MIXTURE CANNOT BE CAXSE 346
     1LCULATED FROM AVAILABLE DATA.  REVISE CROSS SECTION INPUT DATA.)   XSE 347
      END                                                                XSE 348-
```

APPENDIX

PROGRAMMER'S MANUAL FOR THE NOZFIT CODE

The present appendix documents the programming of the NOZFIT code, a small computer program for generating NATA-type curvefits to nozzle profiles. A user's manual for NOZFIT is presented in Appendix D of Volume II of this report (ref. 2).

Section A.1 of this appendix analyzes the NOZFIT main program and the subroutines used to produce plots of nozzle profile curvefits. Section A.2 is a glossary of Fortran symbols for NOZFIT. Finally, a source listing of NOZFIT and its associated subroutines is presented in Section A.3.

A.1 Analysis of NOZFIT Routines

This section discusses and explains the main program and each subroutine of the NOZFIT code. The purpose of this exposition is to provide an entry into the coding for programmers who wish to analyze errors, make corrections, or introduce modifications into the code.

A.1.1 NOZFIT Main Program

NOZFIT consists of two main sections. The first section computes the parameters of the profile fit from the input data (see Appendix D of Volume II); the second produces printed, punched, and plotted output.

(1) Computation of profile fit parameters. -- The parameters* of the curvefit to the nozzle or channel profile are calculated in the loop DØ 140, L = 1, 2. For L = 1, the parameters of the upstream sections are determined; for L = 2, the downstream sections are treated. For each region, the calculations start at the throat and proceed away from the throat. The index K enumerates the profile sections for

---

*The profile fit parameters and the analytical forms in which they appear are defined in Section 4.3 of Volume I (ref. 1).

each region in the order in which they are determined. The index J labels the sections for the entire nozzle, both upstream and downstream, in sequence starting at the upstream end. Finally, JP indexes the downstream boundaries of the profile sections in sequence starting at the upstream end.

The algorithm used to determine the parameters for each section depends upon the input "condition" index, ICØND(J).

a. ICØND(J) = 1

In each region (upstream or downstream), the first section treated is the one adjacent to the throat. This section is required to be a circular arc convex toward the axis (ISHAPE(J) = 2), and must be specified as a throat section (ICØND(J) = 1). For such a throat section, dy/dx must be zero at x = 0. Thus, equation I(122) shows that the abscissa P_2 of the circle center must be zero.

$$P_2 = 0 \tag{178}$$

The circle radius is equal to PAR(1,J):

$$P_3 = 2.54 \cdot \text{PAR}(1,J) \tag{179}$$

where the numerical factor converts from the inch unit used in the input to the centimeter unit used in the output. Finally, the ordinate of the circle center is equal to the throat radius plus the circle radius (see figure 28):

$$P_1 = P_3 + \frac{1}{2} \cdot 2.54 \cdot \text{DTH} \tag{180}$$

where DTH is the throat diameter in inches.

b. ICØND(J) = 2

For the condition ICØND(J) = 2, the current profile section is assumed to be straight (ISHAPE(J) = 1) and tangent to an adjacent circular section nearer the throat. The index of the section nearer the throat is denoted by INDEX. The profile parameters for this section are already known, because of the order which the sections are treated.

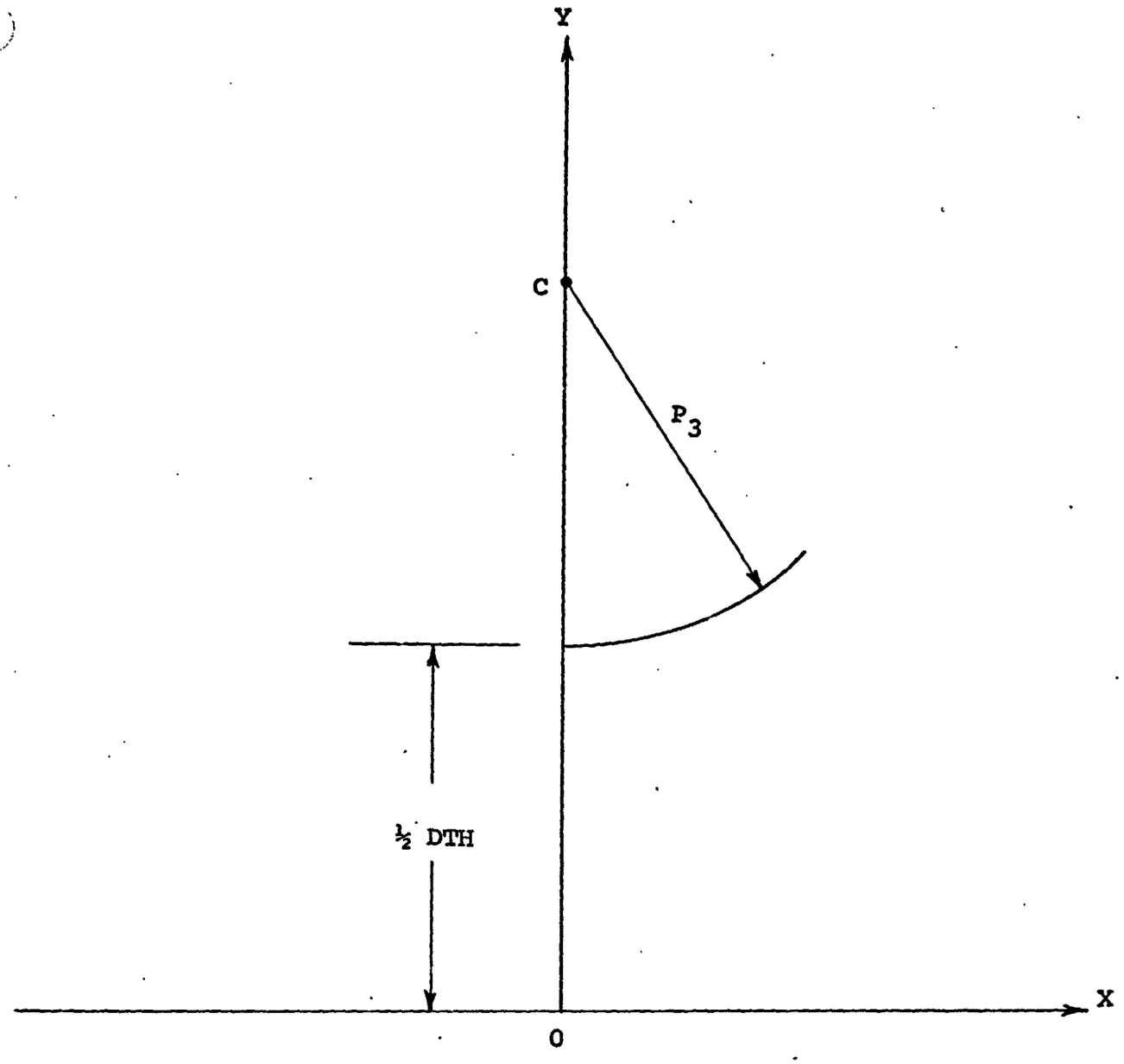


Figure 28. Geometry of the Condition ICØND(J) = 1

Since the sections can be either upstream or downstream of the throat, and since the adjacent section INDEX can be either convex (ISHAPE(INDEX) = 2) or concave (ISHAPE(INDEX) = 3) toward the axis, there are four cases to be considered, as illustrated in figure 29. These cases are all treated in one set of formulas with the aid of the following symbols:

$$S_1 = \begin{cases} -1 & \text{upstream} \\ +1 & \text{downstream} \end{cases} \qquad (181)$$

$$S_2 = \begin{cases} -1 & \text{for circle bottom, ISHAPE(INDEX) = 2} \\ +1 & \text{for circle top, ISHAPE(INDEX) = 3} \end{cases} \qquad (182)$$

The symbol S_1 has the same sign as the slope of the current, straight section; S_2 is negative if the circle center C is above the line passing through the straight section, positive if the center is below.

The point of tangency of the straight and circular sections is marked by a "1" in each part of figure 29. The coordinates of this point are

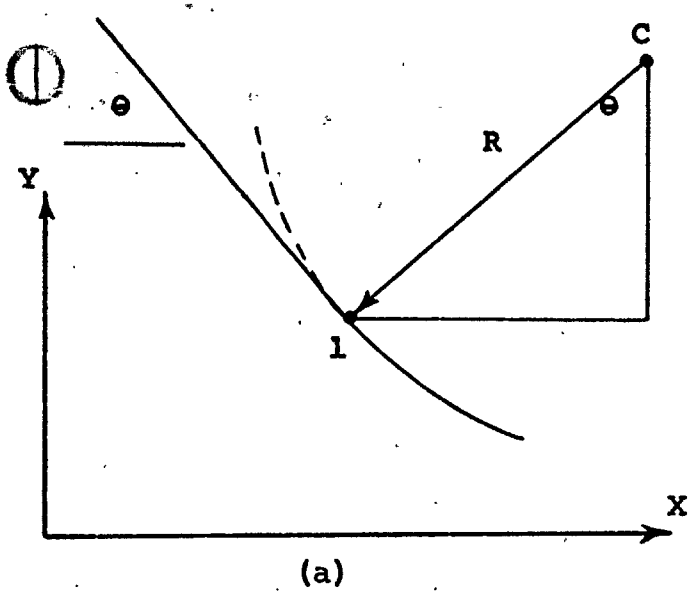
$$X_1 = X_c - S_1 S_2 R \sin \Theta \qquad (183a)$$

$$Y_1 = Y_c + S_2 R \cos \Theta \qquad (183b)$$

Here R = PARAM(3,INDEX) is the circle radius, and Θ = PAR(1,J) is the angle of inclination of the straight section to the nozzle axis, taken positive. It is easy to show that the internal angle of the triangle at the vertex C is equal to Θ. Also X_c and Y_c are the coordinates of the circle center in the adjacent circular section nearer the throat:

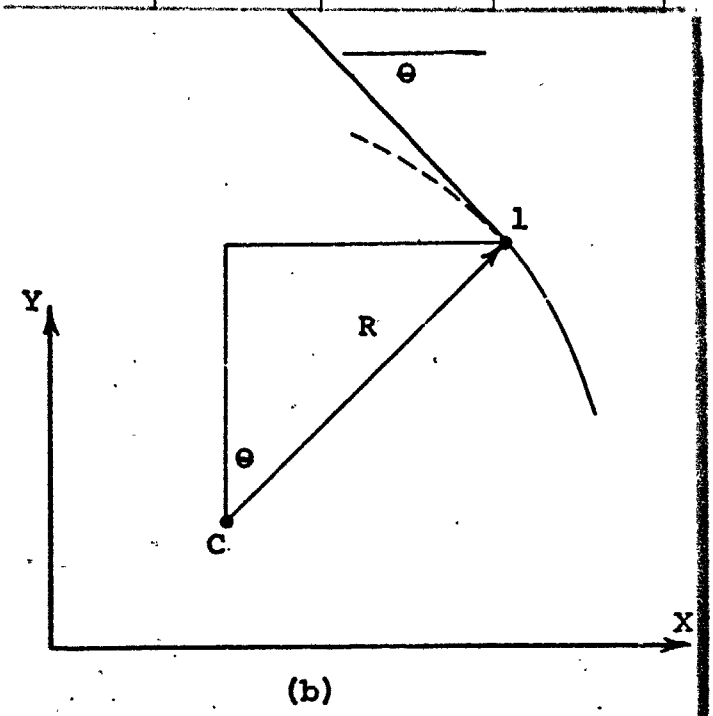
$$X_c = \text{PARAM (2,INDEX)} \qquad (183c)$$

$$Y_c = \text{PARAM (1,INDEX)} \qquad (183d)$$



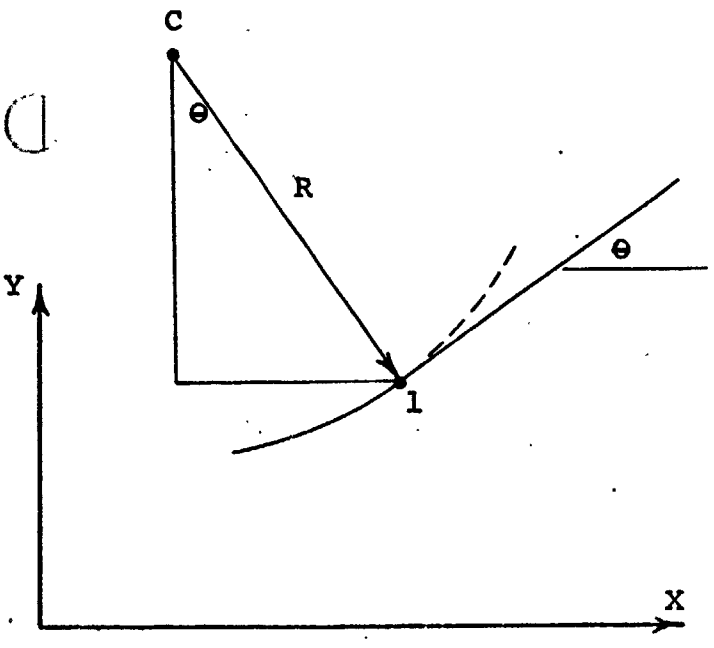
(a)

Upstream, ISHAPE(INDEX) = 2



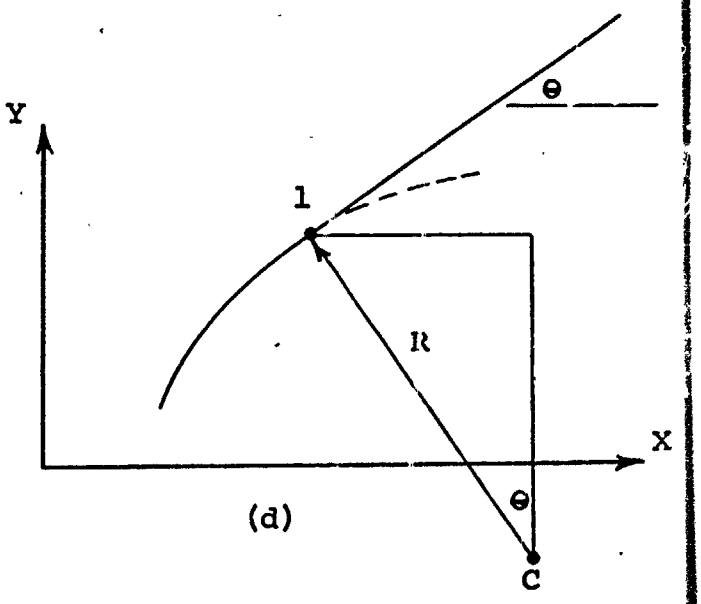
(b)

Upstream, ISHAPE(INDEX) = 3



(c)

Downstream, ISHAPE(INDEX) = 2



(d)

Downstream, ISHAPE(INDEX) = 3

Figure 29. Geometry of the Condition ICØND(J) = 2

The equation for the straight section is

$$Y - Y_1 = S_1 \tan \Theta \cdot (X - X_1) \quad (184)$$

Thus, from equation I(121),

$$P_1 = Y_1 - S_1 X_1 \tan \Theta \quad (185)$$

$$P_2 = S_1 \tan \Theta \quad (186)$$

c. ICØND(J) = 3

For the condition ICØND(J) = 3, the current profile section is assumed to be circular and tangent to an adjoining straight section which is nc rer the throat. The abscissa of the point of tangency is assumed to be PAR(2,J) (inches). As illustrated in figure 30, there are again four cases to be considered. As in the preceding case, the quantities S_1 and S_2, equations (181) and (182), are used to treat all four cases in a single set of formulas.

In the present case, the coordinates of the point of tangency ("1" in figure 30) are known, since

$$X_1 = 2.54 \cdot PAR(2,J) \quad (187)$$

and Y_1 can be calculated from equation I(121) using the known parameter values for the adjoining straight section nearer the throat. The basic problem is to determine the coordinates (X_c, Y_c) of the circle center. Equations (183) are again applicable, and give

$$X_c = X_1 + S_1 S_2 R \sin \Theta \quad (188a)$$

$$Y_c = Y_1 - S_2 R \cos \Theta \quad (188b)$$

From equations I(122) and I(123), $P_2 = X_c$ and $P_1 = Y_c$. The radius P_3 is

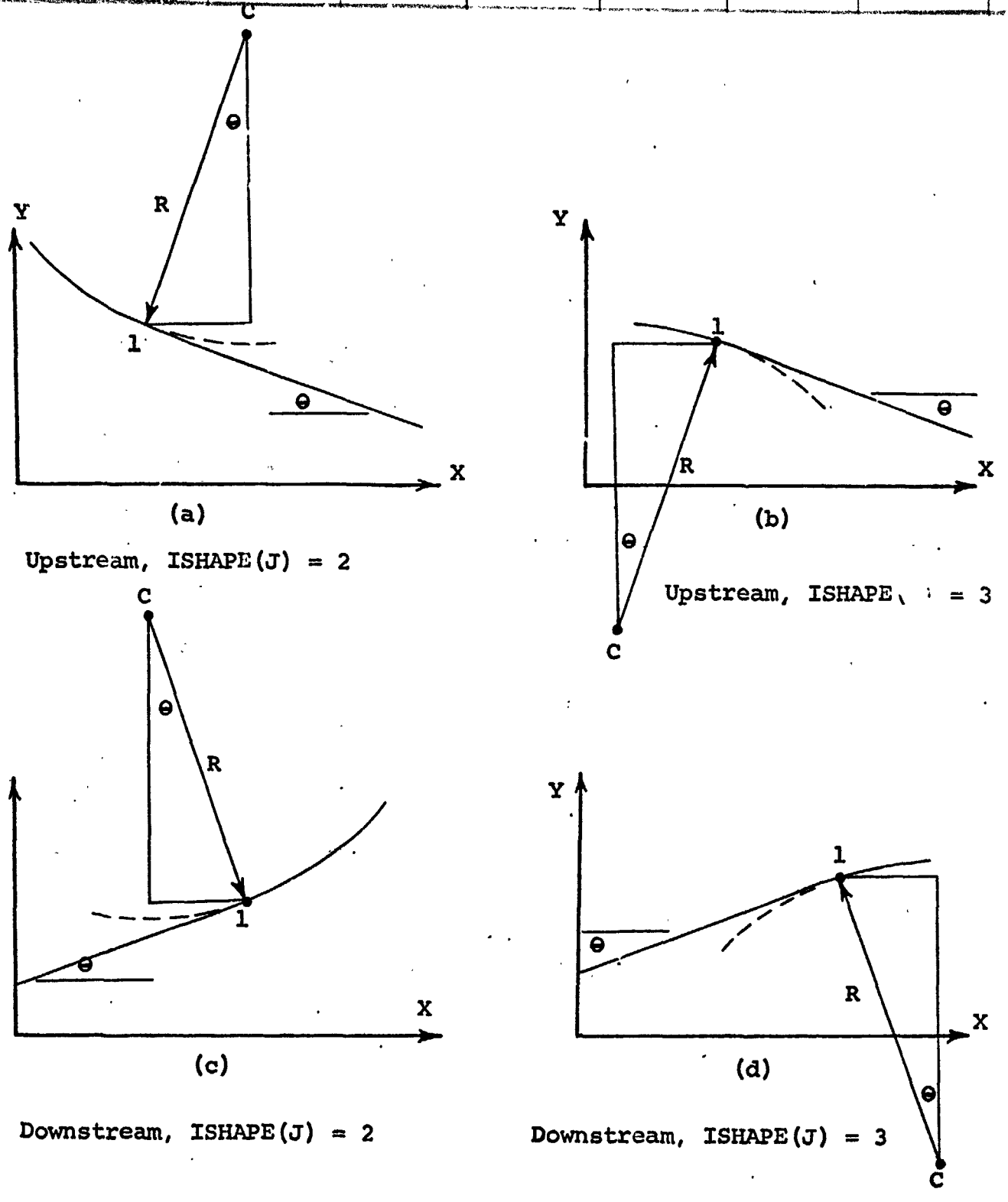


Figure 30. Geometry of the Condition ICØND(J) = 3

$$P_3 = 2.54 \cdot PAR(1,J) \qquad (189)$$

d. ICØND(J) = 4

The condition ICØND(J) = 4 allows the determination of a circular section which is tangent to two adjacent straight sections (J+1 and J-1). Figure 31 illustrates the four cases which must be considered. The straight section nearer the throat has the index INDEX; the one farther from the throat has the index INDX2. The angles θ_1 = PAR(1,INDEX) and θ_2 = PAR(1,INDX2) are known and the abscissa X_1 = 2.54 · PAR(2,J) of the intersection of the two lines is given. Since the curvefit parameters for the section INDEX have already been determined, the ordinate Y_1 of the intersection can be calculated from eq. I(121).

Since the circular section to be determined is tangent to both lines, its center C lies on the bisector of the angle between the two lines. The length of this bisector, from the intersection point "1" to the circle center "C", is denoted by L (or XL in the Fortran). Also, the angle between the bisector and either of the lines is denoted by ϕ. This angle is given by

$$\phi = \frac{1}{2} (\pi - |\theta_2 - \theta_1|) \qquad (190)$$

since $(\pi - |\theta_2 - \theta_1|)$ is the included angle between the lines. Then, from either of the right triangles with a vertex at C,

$$L = R/\sin \phi \qquad (191)$$

For ISHAPE(J) = 2 (parts (a) and (c) of figure 31), the angle C1A is equal to $\phi - \theta_1$. For ISHAPE(J) = 3 (parts (b) and (d) of figure 31), angle C1A is equal to $\phi - \theta_2$. Hence, for ISHAPE(J) = 2,

$$X_c = X_1 - S_1 L \cos (\phi - \theta_1) \qquad (192a)$$

$$Y_c = Y_1 + L \sin (\phi - \theta_1) \qquad (192b)$$

For ISHAPE(J) = 3,

$$X_c = X_1 + S_1 L \cos (\phi - \theta_2) \qquad (193a)$$

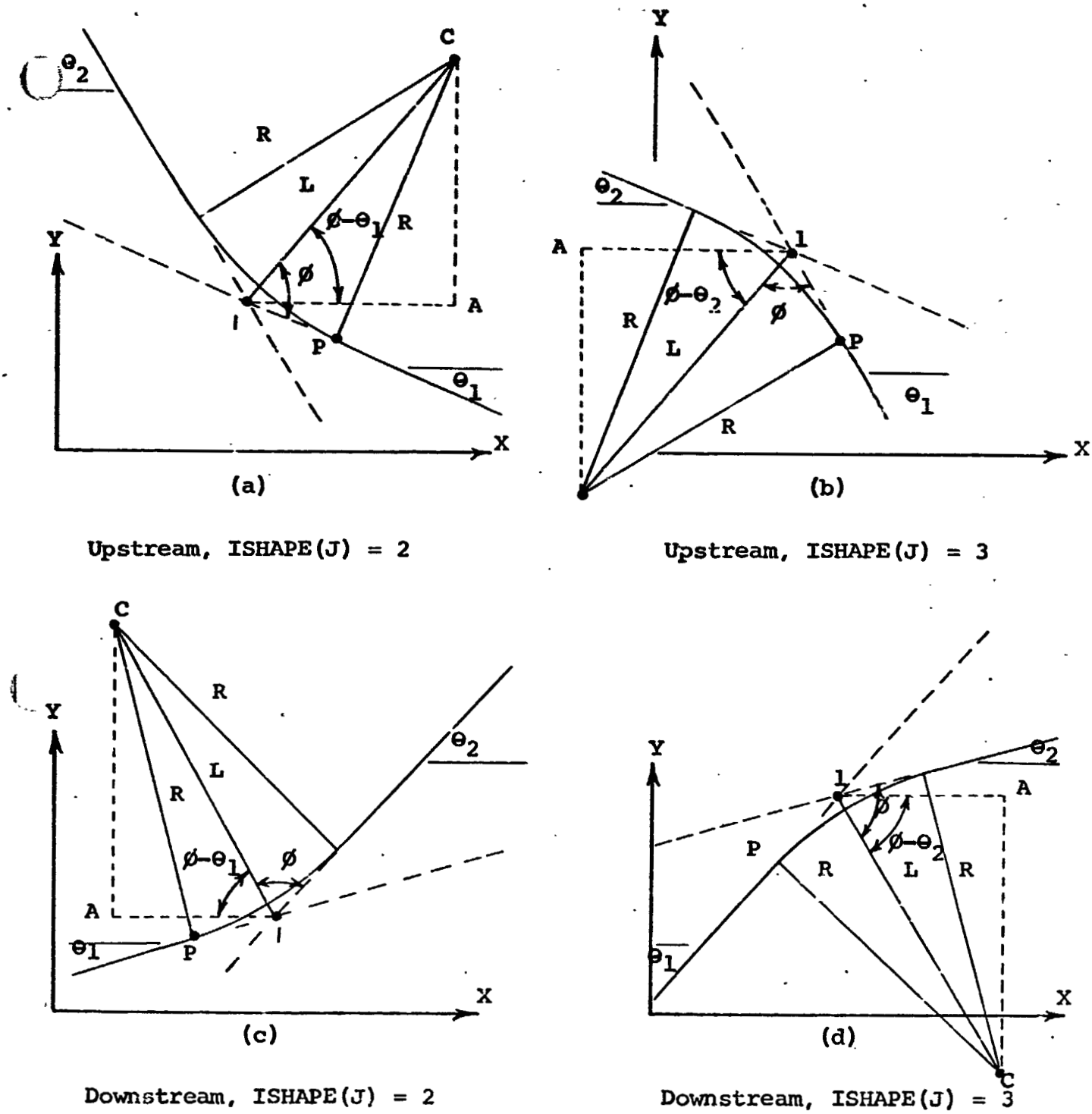


Figure 31. Geometry of the Condition ICOND(J) = 4

$$Y_c = Y_1 - L \sin (\phi - \theta_2) \qquad (193b)$$

For either case, the distance from point "1" to the junction point "P" between the circular section and the straight section nearer the throat is equal to L cos ϕ. The projection of this distance onto the X-axis is L cos $\phi \cdot \cos \theta_1$. Hence, the abscissa of the junction point is

$$X_p = X_1 - S_1 L \cos \phi \cos \theta_1 \qquad (194)$$

It is clear from figure 31 that ISHAPE(J) can be equal to 2 only if $\theta_2 > \theta_1$ and ISHAPE(J) can be equal to 3 only if $\theta_1 > \theta_2$. If the user inputs a value for ISHAPE(J) violating these conditions, an error message is written and execution is terminated.

(2) Input and Output. -- NATA and its auxiliary programs are designed for operation on either a UNIVAC 1108 or an IBM System 360. So far as NØZFIT is concerned, the principal difference between the two computer systems is the word length. The UNIVAC 1108 contains 36-bit words which can store 6 alphanumeric characters. The IBM 360 contains 32- and 64-bit words which can store 4 or 8 alphanumeric characters, respectively. The plot routines used at NASA/JSC on the UNIVAC 1108 and at Avco Systems Division on the IBM 360/75 assume that arrays of alphanumeric information are close-packed (i.e., with no unused characters). To provide correct handling of the information on the "header" card*, NØZFIT contains read and write statements based on both the A6 format (for the UNIVAC) and the A4 format (for the IBM 360). The correct input/output statements for alphanumeric information are selected by reference to an indicator IWØRDL, which is set in subroutine FRAMEB (see below). Two versions of FRAMEB are available, one for use on the UNIVAC 1108 at the Johnson Space Center, the other for use on the IBM 360/75 at Avco. Apart from the differences in this one subroutine, NØZFIT

*See discussion of NØZFIT inputs in Appendix D of Volume II (ref. 2).

and its plot subroutines are fully compatible with both computer systems.

The coding of the printed and punched output of the profile curvefit parameters is straightforward. In the profile calculation (statements 240 to 310), the abscissa X is varied by increments of 0.03 inch from X = XSTART to X = XSTART + 6 inches. At each X, the index K of the profile section in which X lies is determined by finding the first K for which the downstream boundary ATP(K) is greater than X. The ordinate Y is then computed from equation I(121), I(122), or I(123) according as ISHAPE(K) is equal to 1, 2, or 3. The result of this calculation is in centimeters, and is converted into inches for output.

Finally, if the indicator PLØTS is .TRUE., the profile is plotted by a call to subroutine GRAPH. After the final case in the job (ENDJOB = .TRUE.), the entry FRAMB2 of subroutine FRAMEB is called. If the program is being run on the Avco IBM 360/75, FRAMB2 calls a library routine which closes the plot data set.

A.1.2 Subroutine FRAMEB

Subroutine FRAMEB contains all of the irreconcilable differences between the UNIVAC 1108 and IBM 360 versions of NØZFIT. Two versions of this routine are provided (Section A.3), one for each type of computer system.

The IBM 360 version sets the indicator IWØRDL to 2, to induce NØZFIT to use A4 formats in reading and writing alphanumeric information. It also calls the Avco library routine IDFRMV to produce an identification frame preceding the plots of nozzle profiles, if the control variable PLØTS is .TRUE.. When called through the entry FRAMB2, it calls the Avco library routine PLTND to empty the plot buffer and close the plotting data set.

The UNIVAC 1108 version of FRAMEB sets IWØRDL to 1, to induce NØZFIT to use A6 formats in reading and writing alphanumeric information. It performs no other function.

A plot identification frame is produced automatically on the 1108 computers at NASA/JSC when plots are requested on the JØB card, and the data set is also closed automatically.

A.1.3 Subroutine GRAPH

GRAPH is a general-purpose plotting routine which was developed for use in the auxiliary programs to the NATA code. It uses several lower level plot routines programmed by North American Aviation (NAA) as software for the S-C 4020 high-speed microfilm recorder. The capabilities of GRAPH may be summarized as follows:

(1) It produces Cartesian graphs of Y versus X from data stored in arrays.

(2) The data points can be displayed by up to 10 different plot symbols, or not displayed.

(3) The data points can be connected by lines, or not connected. The lines can be continuous, dashed (with user-controlled dash length), dotted, or traced by a string of Charactron symbols.

(4) Different data sets can be plotted on the same frame. The points from the different data sets can be represented by different plot symbols and/or connected by lines of different types.

(5) A parameter value can be associated with each data set. The curves for different data sets can be labelled with numerical labels and a table displaying the labels and the associated parameter values can be shown in the margin to the right of the graph. Up to three distinct parameters, each with up to 10 values, can be included in a single plot.

The arguments in the calling sequence of GRAPH are defined in the Glossary of Fortran Symbols (Section A.2).

If a table of parameter values is to be printed in the frame, the first call to GRAPH must be preceded by a call to the NAA routine SETMIV (references 3, 17) to provide the required space in the right margin:

```
CALL SETMIV (30, 125, 24, 24)
```

(1) Grid generation. -- The first part of GRAPH, down to statement 110, is executed only when the routine is called with NF $\neq$ 0 to establish the grid for a new frame. In this case, the counter JCURVE for data sets plotted and the counter NPL for parameter tables are initialized to zero, and the NAA routine SMXYV (references 3, 17) is called to establish the logarithmic or linear mode for each axis as indicated by the GRAPH arguments LX, LY. Then the NAA routine DXDYV (references 3, 17) is called for each axis to determine the arguments DX, DY, NNN, MMM, III, JJJ, NXX, NYY of subroutine GRID1V. If either axis is to be used as a logarithmic scale (LX $\neq$ 0 or LY $\neq$ 0), then the corresponding call to DXDYV is skipped, and DX and DY are set to 1 in accordance with instructions in references 3 and 17.

Beginning at statement 50, the parameters NXY(1) and NXY(2), which determine the numbers of characters to be displayed in the labels of vertical and horizontal grid lines, are established. The corresponding values NXX and NYY provided by DXDYV were found to be unsatisfactory in some applications. For each axis, if $\log_{10}$ of the maximum absolute value to be plotted is greater than 5 or less than -3, the grid-line labels are printed in scientific notation, with three significant digits, by setting NXY(I) = -3. Otherwise, a fixed-point format is used, with a number of characters determined from the magnitude of the maximum value to be plotted.

Next, the GRIV1V arguments III and JJJ are reversed in sign to force placement of the grid line labels outside

of the grid area. Also, if numerical labels for the individual curves are to be displayed, the lower limits XL and YL are decreased slightly, and the upper limit YU is increased slightly,to allow space for the labels within the grid area. Then GRIDlV is called to draw the grid and to establish scaling factors for subsequent use in plotting. Finally, the counter ICURVE for curves in the first family (i.e., for various values of the first parameter, if any) is initialized to zero.

(2) Plotting of a data set. -- The second part of GRAPH plots an individual data set, that is, the data contained in the arrays X and Y for a single call to GRAPH. This section of the subroutine extends from statement 110 down to statement 190.

First, the counters ICURVE and JCURVE are each incremented by 1. If ICURVE is equal to 1 after this increase, the counter NPL for parameter tables is also increased by 1. Note that ICURVE is equal to 1 at this point if this is the first call to GRAPH for a new frame, or if ICURVE was set to zero in the calling routine before the current call to GRAPH. Setting ICURVE to 0 in the calling routine is a signal to GRAPH that a new family of curves with a new parameter is being started.

Next, the NAA routines NXV and NYV are used to initialize IX1 and IY1 to the raster coordinates corresponding to X(1) and Y(1), respectively. Then the plot symbol to be used in displaying the data points is selected on the basis of the arguments NC(1) and NC(2). If NC(1) is negative, the symbol index NSYMBL is set equal to ICURVE, so that a new symbol from the standard table in PØINTB is used for the points on each successive curve in a family of curves. If NC(1) is positive, NSYMBL is set equal to NC(1); this option allows direct user control of the plot symbol. For NC(1) = 0, the individual points are not plotted.

The DØ loop from statement 140 to statement 180 plots the data points and draws the connecting lines, if any. First, subroutine PØINTB (Section A.1.4 of this appendix) is called to plot the current (Ith) data point. Then I2 is set equal to I+1. If LINE(1) is nonzero, subroutine LINEB (Section A.1.5 of this appendix) is called to draw the line segment from the Ith to the I2th point. The type of line to be drawn (continuous, dashed, etc.) is determined by the argument LINE in the call to LINEB. Then IX1 and IX2 are set equal to the raster coordinates corresponding to X(I2) and Y(I2), respectively, using the NAA function routines NXV and NYV, for use in calling PØINTB the next time around the loop.

When these plotting operations have been completed, if JCURVE is equal to 1, the NAA routine CHSIZV is called with arguments (3,3) to restore the normal character size.* Then subroutine RITE2V (references 3, 17) is called to write the abscissa and ordinate labels ABSL and ØRDL and the plot title TITL in the bottom, left, and top margins of the plot, respectively.

(3) Parameter tables and lables. -- If the argument LL is zero, then parameter tables and curve labelling are not to be used, and a RETURN is executed at statement 190 in GRAPH. If LL is 1, the parameter tables are produced but the curve labelling is skipped. If LL is neither 0 nor 1, then the curves are labelled also. To implement the labelling, the code calls the NAA routines NXV and NYV to determine the raster coordinates, NXV1 and NYV1, of the first point on the curve. It then computes the raster coordinates, IX1 and IY1, of a point which lies 16 raster units away from the first point of the curve in a direction opposite to that in which the second point lies. It then writes the curve label (CURVE = ICURVE) into the plot by calling the NAA subroutine LABLV (references 3, 17).

*A nonstandard character size is set in PØINTB by calling CHSIZV(3,2), to obtain plot symbols whose height and width are approximately equal.

The tables of parameter values are produced by the coding from statement 200 to statement 300. For ICURVE = 1, the code first writes the table heading. The heading consists of a sample of the type of line used for the current family of curves, and a two-line printed specification of the nature of the current parameter. The sample line is produced by calling the entry LINEBR of subroutine LINEB (discussed below). It is horizontal and lies at the raster ordinate IY2. The printed heading is produced by calling the NAA routine PRINTV. Its first line lies 15 raster units below the sample line, and its second line is an additional 20 raster units lower. The tables for different parameters (families of curves) have a vertical separation of 300 raster units.

The body of the parameter table consists of rows each containing the numerical label for the curve (equal to CURVE), a sample of the plot symbol used, and the parameter value. The numerical label is omitted for LL = 1. When it is included, it is produced by calling LABLV. The sample plot symbol is produced by calling PØINTB, and the parameter value by another call to LABLV. The arguments NCHAR and NDMAX of LABLV are determined as functions of IPVM, which is $\log_{10}$ (PVAL) rounded down in magnitude to the nearest integer. If IPVM is greater than 4 or less than -2, NCHAR is set to -3 to give scientific notation with three significant figures. Otherwise, fixed-point notation is used, and NCHAR and NDMAX are adjusted to provide the required number of characters and positions to the left of the decimal point.

A.1.4 Subroutine LINEB

Subroutine LINEB draws a straight line between two specified points (X1, Y1), (X2, Y2). When it is called through its entry LINEBR, the two points are specified in raster coordinates (MX1, MY1), (MX2, MY2). The argument LLINE is a three-element array whose values control the type of line produced and its darkness, as defined in the Glossary of Fortran Symbols (Section A.2).

The raster coordinates of the two points to be connected by a line are denoted, within LINEB, by (NX1, NY1), (NX2, NY2). If entry was through LINEBR, these values are set equal to MX1, etc. If entry was through LINEB, the raster coordinates are determined by calling the NAA subroutines NXV and NYV.

For LINE ≡ LLINE(1) = 1, the points are connected with a continuous line by calling NAA subroutine LINEV. This call is executed IDARK times to obtain the desired darkness.

For LINE > 1, beginning in statement 50, the sine (SINA) and cosine (CØSA) of the angle between the vector from point 1 to point 2 and the positive X-axis are computed. Then, for LINE = 2, a dashed line is drawn from point 1 to point 2, with the dash length in millimeters* being approximately equal to IPAR ≡ LLINE(3). The distance DEL from the beginning of one dash to the beginning of the next is calculated, in raster units, as 10.8 x IPAR. The numerical coefficient in this formula is obtained by noting that the frame width is 1024 raster units and is 190 mm in hard copy. Thus, 1 mm on the hard copy is equivalent to 5.4 raster units, the dash length HDEL is equal to 5.4 x IPAR, and DEL is twice as large, allowing for the gap between two successive dashes. The number of dashes to be used in the line joining the two points is thus NP = DIST/DEL, where DIST is the separation of the points in raster units. If the two points are too close together in comparison with the specified dash length, integer arithmetic gives NP = 0. This possibility should be borne in mind by the user of GRAPH.

LINEB next computes the increments IDX and IDY in raster coordinates between the beginning and end of a single dash, using the previously determined trigonometric

*This description of the dash length is applicable to hard copy from the Calcomp 890 CRT Plotter at Avco, or to enlargements of microfilm plots produced on the SD 4060 at NASA/JSC to a frame size of 7.5 x 7.5 inches.

functions CØSA and SINA. It then calls the NAA subroutine LINE2V (references 3, 17) NP times to draw the dashed line joining the end points specified in the arguments. In the DØ loop, DØ 70 I = 1, NP, the quantities VX1 and VY1 represent the desired starting points for each dash in raster coordinates. These quantities are calculated using floating point arithmetic to avoid the loss of accuracy possible in integer arithmetic, and are rounded down to obtain the integer values NX1, NY1 used in the calls to LINE2V.

For LINE = 3, the line joining the specified end points is produced by plotting a string of Charactron symbols. The symbol to be used is specified by IPAR, which is used as the index in the POINTV character table (references 3, 17). The following list is excerpted from references 3 and 17.

IPAR	Character
0	.
1	O
2	×
4	Y
5	+
6	*
10	H
15	⊗
22	Θ
25	I

The separation of successive points, DEL, is set to 5 raster units if IPAR = 0 (i.e., if the plotting character is a dot), and is set to 15 raster units otherwise. The number of characters to be used in tracing the line is then NP = DIST/DEL. The string of characters is then produced by calling the four-argument version of PØINTV (references 3, 17) NP times, in the loop DØ120 I = 1, NP. The minus sign on the argument IPAR of PØINTV suppresses the plotting of a central dot in each character.

A.1.5 Subroutine POINTB

This routine plots a point at the raster coordinates (IX, IY), using a plot symbol selected from a table ICHAR(I) of ten standard symbols. The index NSYMBL in the calling sequence specifies which symbol is to be used. The symbols are as follows:

I	Symbol
1	X
2	+
3	O
4	*
5	H
6	C
7	Ↄ
8	Y
9	⅄
10	I

These symbols are produced by calling the NAA subroutine RITE2V (references 3, 17). The inversions for I = 7 and 9, and the 90-degree rotation for I = 10, are accomplished using the argument K(NSYMBL) of RITE2V. Before RITE2V is called, a call is made to CHSIZV (references 3, 17) to make the height of the symbols nearly equal to their width instead of 50 percent greater, as is normally the case. The argument IDARK of PØINTB and RITE2V controls the number of times the symbol is overdrawn and thus its darkness in the plot.

A.2 Glossary of Fortran Symbols for NOZFIT

This section defines the Fortran symbols used in the NOZFIT main program and in its associated subroutines. The variables are listed in alphanumeric order within each routine, and the routines themselves are in alphabetic order. There are no common variables.

A.2.1 Main Program

ABSL(I)	Array containing alphanumeric abscissa label for plots
ACØN	Conversion factor (180/π) from radians to degrees
ATP(JP)	Axial coordinate at the downstream boundary of the JPth profile section (cm)
CARDS	Logical control; .FALSE. value suppresses punched card output
CØMMNT(I)	Hollerith descriptive data
CTHETA	Cos (THETA)
DTH	Throat diameter (inch)
DX	Increment in X for calculation of profile (inch)
ENDJØB	Logical control; .TRUE. value terminates the job at the end of the current case
FISHAP(I)	Floating point value representing ISHAPE(I) in punched-card output
FNSECT(I)	NSECTS(I) + 0.1 for card output
I	DØ index
IC	ICØND(J) for the Jth profile section
ICØND(J)	Index specifying the type of condition to be used in determining the Jth profile section (see input discussion, Appendix D of Volume II, ref. 2 or Section A.1.1 of the present Appendix)
ICURVE	Unused argument of subroutine GRAPH
INCR(L)	0 for L = 1; 1 for L = 2

INDEX	Index of the profile section adjacent to the current (Jth) section and nearer the throat
INDX2	Index of the profile section adjacent to the current (Jth) section and farther from the throat
ISGN(L)	1 for L = 1; -1 for L = 2
ISH	ISHAPE(INDEX)
ISHAPE(J)	Indicator for shape of Jth profile section (see input discussion, Appendix D of Volume II)
ISJ	ISHAPE(K)
IWØRDL	Indicator for number of characters per word on the type of computer system being used: IWØRDL = 1 6-character word (UNIVAC 1108) IWØRDL = 2 4-character word (IBM 360)
J	Profile section index
JP	Index of profile section boundaries
J1	2K - 1; first J value on the Kth card containing PARAM(I,J) data in the punched output
J2	J1 + 1; second J value on the Kth card containing PARAM(I,J) data in the punched output
K	DØ index; profile section index in calculation of profile from curvefit
KP	Continuation card counter punched in column 6 of output cards

L	Indicator for upstream profile sections (L = 1) and downstream sections (L = 2)
LINE(I)	Argument of subroutine GRAPH (see below)
LL	Argument of subroutine GRAPH (see below)
NC(I)	Argument of subroutine GRAPH (see below)
NØZZLE	Profile index for use in NATA
NS	NSECTS(L) in computation of profile fit parameters; total number of sections in coding of output
NSM1	NS-1
NSECTS(L)	Number of profile sections upstream (L = 1) and downstream (L = 2) of the throat
ØRDL(I)	Alphanumeric label for ordinate in plot
PAR(I,J)	Input parameter values for the Jth profile section; see discussion of inputs, Appendix D of Volume II
PARAM(I,J)	Parameters of curvefit for Jth profile section; see discussion of outputs, Appendix D of Volume II
PHI	0.5 (π - |THETA2-THETA1|); see discussion of ICØND = 4 in Section A.1.1.
PI	3.14159265
PLABL(I)	Argument of subroutine GRAPH (see below)
PLØTS	Logical control, .FALSE. value suppresses plot output
PVAL	Argument of subroutine GRAPH (see below)

R	Circle radius for a circular-arc section (cm)
RTHCM	Throat radius in centimeters
STHETA	Sin (THETA)
S1	-1.0 for L = 1, +1.0 for L = 2
S2	0.0 for ISHAPE(J) = 1; -1.0 for ISHAPE(J) = 2; +1.0 for ISHAPE(J) = 3
S2ARR(ISH)	Array containing S2 values for the three shapes
THETA	Angle of inclination of a straight section to the nozzle axis (radians)
THETA1	For ICØND = 4, inclination angle of the adjacent section closer to the throat
THETA2	For ICØND = 4, inclination angle of the adjacent section farther from the throat
TITLE(I)	Title for plots (Hollerith)
TTHETA	Tan(THETA)
X	Axial coordinate (cm)
XC	X-coordinate of the circle center for the adjacent section nearer the throat
XINCH(I)	Axial coordinate (inches)
XL	R/sin(PHI); see discussion of ICØND = 4 in Section A.1.1
XLØ	Lower limit of abscissa in plot
XSTART	Left limit on X for calculating nozzle profile (inch)
XUP	Upper limit of abscissa in plot

XZERØ	Axial coordinate of start of boundary layer, for use in NATA (cm)
XZERØI	XZERØ expressed in inches
X1	Intermediate X values in calculations of parameters for ICØND = 2, 3, and 4
Y	Profile ordinate (cm)
YC	Y-coordinate of the circle center for the adjacent section nearer the throat
YINCH(I)	Profile ordinate (inches)
YLØ	Lower limit on ordinate in plot
YUP	Upper limit on ordinate in plot
Y1	Intermediate Y values in calculations of parameters for ICØND = 2, 3, and 4

A.2.2 Subroutine FRAMEB

BIN	Alphanumeric bin number for plot identification frame (Avco)
IWØRDL	Defined above (Main program)
PLØTS	Defined above (Main program)
XMEMØ	Alphanumeric memo number for plot identification (Avco)
XNAME	Alphanumeric submitter name for plot identification frame (Avco)

A.2.3 Subroutine GRAPH

ABSL	Array containing 48-character label for abscissa scale
CURVE	ICURVE expressed as a floating-point value
DC	Parameter controlling the spacing of grid lines on linear scales; DC = 16 is a normal value. The spacing increases or decreases with the value of DC
DX	Floating point data increment at which vertical grid lines are displayed
DY	Floating point data increment at which horizontal grid lines are displayed
DYL	Amount by which YL is decreased and YU increased to allow space for numerical labels of curves
HYP	Distance (in raster units) between the first two points of a curve
I	ØØ index
ICURVE	Counter for the number of curves plotted on the same frame. If this argument is set to zero by the calling program, a new parameter label for a new series of curves is printed in the margin; this feature is used only when two or more families of curves are to be plotted in the same frame.
IDXV	Interval in raster units between the abscissas of the first two points in a curve
IDYV	Interval in raster units between the ordinates of the first two points of a curve

IERR	Error indicator in subroutine DXDYV (references 3, 17); if a nonzero value of IERR is returned, grid generation is impossible with the given upper and lower limits on X and Y
III	Argument of subroutine GRIDlV (references 3, 17). Every IIIth vertical line of the grid is labelled with a numerical value; a negative III forces placing the labels outside the grid
IMAG(I)	For I = 1 and 2, $\log_{10}$ of the largest numerical value to be plotted on the abscissa and ordinate, respectively
IPVM	Log_{10} of the absolute magnitude of a parameter value (PVAL)
IXl	Abscissa of the location where a curve label is to be printed (raster units)
IY	Ordinate of the location where a parameter value is to be printed in the margin (raster units)
IYl	Ordinate of the location where a curve label is to be printed (raster units)
IY2	Ordinate of the location where a sample of the type of line used for a family of curves is drawn in the margin (raster units)
I2	I + 1
J	DO index
JCURVE	Counter for the total number of curves plotted in a given frame

JJJ	Argument of subroutine GRID1V (references 3, 17). Every JJJth horizontal line of the grid is labelled with a numerical value; a negative JJJ forces placing the labels outside the grid
LINE(I)	Control parameters for lines joining data points, defined as follows:
LINE(1)	0 Do not join points with lines 1 Join points with continuous lines 2 Join points with dashed lines 3 Join points with strings of characters
LINE(2)	Number of strikes for line; use 2 for dark lines
LINE(3)	Ignored if LINE(1) is 0 or 1 Length of dashes in mm if LINE(1) = 2 Index of character in POINTV table if LINE(1) = 3 (see Procedures Manual, Computation and Analysis Division, MSC, p. 5.c.7.5.1.127) (Use LINE(3) = 0 with LINE(1) = 3 to produce dotted lines)
LL	Control parameter for labelling of curves: LL = 0 Suppress labelling of curves and printing of parameter values in table at right of grid LL = 1 Print parameter values and plot symbols but do not label curves LL = 2 Label curves and print parameter values and plot symbols
LX	Argument of subroutine SMXYV (references 3, 17); 0 sets linear mode for abscissa, nonzero log mode

LY	Argument of subroutine SMXYV (references 3, 17); 0 sets linear mode for ordinate, nonzero log mode
MMM	Argument of subroutine GRIDlV (references 3, 17). Every MMMth horizontal grid line is retraced for emphasis
NC(I)	For I = 1 and 2, control parameters for plotting points:
	NC(1) < 0 Use standard table of 10 characters for plotting points, a new character for each successive call to GRAPH for the same family of curves in the same frame (see subroutine PØINTB)
	NC(1) = 0 Supress plotting of points
	NC(1) > 0 Use the NC(1)th character in the standard table (see PØINTB)
	NC(2) Number of strikes for each symbol plotted; use 1 for light characters, 2 for dark
NCHAR	Argument of subroutine LABLV (see references 3, 17); number of characters to be displayed, including leading blanks and the decimal point, if any
NCl	NC(1)
NC2	NC(2)
NDMAX	Argument of subroutine LABLV (see references 3, 17); maximum number of characters to be displayed to the left of the decimal point

NF	Control parameter for frame advance: NF = 0 Plot data on same frame NF = 1 Start a new frame, establish a grid, and plot the data on the new frame
NLAST	Argument of subroutine RITE2V (references 3, 17); not used in GRAPH
NNN	Argument of subroutine GRID1V (references 3, 17). Every NNNth vertical grid line is retraced for emphasis
NP	Number of data points to be plotted
NPL	Counter for families of curves plotted in the same frame
NSYMBL	Index of point symbol in standard table (see PØINTB)
NXV1	Abscissa of first point in raster coordinates
NXX	Argument of subroutine DXDYV (references 3, 17); not used in GRAPH
NXY(I)	For I = 1 and 2, the number of characters to be displayed in the labels of vertical and horizontal grid lines; arguments of subroutine GRID1V (references 3, 17)
NYV1	Ordinate of first point in raster coordinates
NYY	Argument of subroutine DXDYV (references 3, 17); not used in GRAPH
ØRDL(I)	Alphanumeric label for ordinate
PLABL(I,J)	For J = 1, 2, a two-line alphanumeric label for the parameter

PVAL	Parameter value
TITL(I)	Alphanumeric title
X(I)	Array containing the abscissas of the data points
XL	Lower limit on X for all data to be plotted in a given frame
XU	Upper limit on X for all data to be plotted in a given frame
XYMAG(I)	For I = 1 or 2, the largest absolute magnitude abscissa or ordinate value to be plotted in a given frame
Y(I)	Array containing the ordinates of the data points
YL	Lower limit on Y for all data to be plotted in a given frame
YU	Upper limit on Y for all data to be plotted in a given frame

A.2.4 Subroutine LINEB

CØSA	Cosine of the angle between the vector to be drawn and the positive X-axis
DEL	Twice the dash length in raster units
DIST	Distance between the beginning and end of the vector in raster units
HDEL	Dash length in raster units
I	DØ index over the dashed or plot symbols used to draw the vector
IDARK	Number of times lines or symbols are drawn

IDX	Increment in the abscissa for a dash (raster units)
IDY	Increment in the ordinate for a dash (raster units)
IPAR	Length of dashes in millimeters for LINE = 2; integer (NS) for selecting the plot symbol from the table in subroutine PØINTV (references 3, 17) for LINE = 3
J	DØ index for double-strike plotting to obtain dark lines or symbols
LINE	Control parameter for type of line: LINE = 1 Continuous line LINE = 2 Dashed line LINE = 3 String of Charactron symbols
LLINE(I)	LLINE(1) = LINE LLINE(2) = IDARK LLINE(3) = IPAR
MX1	Abscissa of the start point in raster units
MX2	Abscissa of the end point in raster units
MY1	Ordinate of the start point in raster units
MY2	Ordinate of the end point in raster units
NP	Number of dashes in vector
NX1	Abscissa of the start point in raster units
NX2	Abscissa of the end point in raster units
NY1	Ordinate of the start point in raster units
NY2	Ordinate of the end point in raster units

RASTER	Logical indicator, .TRUE. for entry through LINEBR, .FALSE. for entry through LINEB
SINA	Sine of the angle between the vector to be drawn and the positive X-axis
SQ	Square of the distance between the start and end points of the vector (raster units)
VX1	Floating point value of NX1
VY1	Floating point value of NY1
X1	Abscissa of the start point
X2	Abscissa of the end point
Y1	Ordinate of the start point
Y2	Ordinate of the end point

A.2.5 Subroutine PØINTB

CHAR(I)	Array of characters for plotting points
IDARK	Argument of RITE2V controlling darkness of plotted symbol (see references 3, 17)
IX	Abscissa of location where point is plotted (raster coordinate)
IY	Ordinate of location where point is plotted (raster coordinate)
K(I)	Argument of RITE2V controlling orientation of plotted symbol; used to present each of the symbols "H", "C", and "Y" in two orientations
NLAST	Argument of RITE2V (references 3, 17); not used in PØINTB

NSYMBL	Index used to select plotting symbol from CHAR array

A.3 Listing of NØZFIT

Presented below are source program listings for the main program of NØZFIT, and for subroutines FRAMEB, GRAPH, LINEB, and PØINTB. Listings are provided for both the IBM (Avco) and UNIVAC (NASA/JSC) versions of FRAMEB.

```
C    4540N  PROGRAM FOR DETERMINING AREA R   ,0 PARAMETERS FOR NATA          NOZ    1
      LOGICAL ENDJOB,CARDS,PLOTS                                             NOZ    2
      DIMENSION ISHAPE(12),PARAM(3,12),ATP(11),PLABL(6)                      NOZ    3
      DIMENSION FNSECT(2),FISHAP(12)                                         NOZ    4
      DATA PI /3.14159265/, CARDS,PLOTS /2*.TRUE./                           NOZ    5
      DIMENSION PAR(2,12),ICOND(12),NSECTS(2)                                NOZ    6
      DIMENSION INCR(2), S2ARR(3)                                            NOZ    7
      NAMELIST /INPUT/  NSECTS,ISHAPE,PAR,ICOND,DTH,ENDJOB,XSTART,           NOZ    8
     1      XZEROI,NOZZLE,CARDS,PLOTS                                        NOZ    9
C                                                                            NOZ   10
CCC   DEFINITIONS OF INPUTS                                                  NOZ   11
CCC                                                                          NOZ   12
CCC   NSECTS(1)    NUMBER OF UPSTREAM SECTIONS IN NOZZLE PROFILE             NOZ   13
CCC   NSECTS(2)    NUMBER OF DOWNSTREAM SECTIONS IN NOZZLE PROFILE           NOZ   14
CCC   NOTE-- NSECTS(1) + NSECTS(2) MUST BE .LE. 10                           NOZ   15
CCC   ISHAPE(J)    INDEX SPECIFYING SHAPE OF J-TH SECTION (COUNTING FROM     NOZ   16
CCC        UPSTREAM INLET)                                                   NOZ   17
CCC                ISHAPE(J)=1   STRAIGHT SECTION                          NOZ   18
CCC                ISHAPE(J)=2   CIRCLE BOTTOM                             NOZ   19
CCC                ISHAPE(J)=3   CIRCLE TOP                                NOZ   20
CCC   PAR(I,J)     PARAMETER VALUES FOR J-TH SECTION                         NOZ   21
CCC                FOR ISHAPE(J)=1, PAR(1,J)=ANGLE OF INCLINATION TO         NOZ   22
CCC                      NOZZLE AXIS IN DEGREES (POSITIVE VALUE)             NOZ   23
CCC                FOR ISHAPE(J)=2 OR 3, PAR(1,J)=CIRCLE RADIUS IN INCHES    NOZ   24
C     DTH          THROAT DIAMETER IN INCHES                                 NOZ   25
      ACON=180./PI                                                           NOZ   26
C     ICOND(J)     INDEX SPECIFYING CONDITION DEFINING J-TH SECTION          NOZ   27
CCC                ICOND(J)=1 THROAT CONDITION (Y=1.27*DTH AND DY/DX=0       NOZ   28
CCC                      AT X=0)                                             NOZ   29
CCC                ICOND(J)=2 STRAIGHT SECTION (J) IS TANGENT TO ADJACENT    NOZ   30
CCC                      CIRCULAR SECTION NEARER THE THROAT                  NOZ   31
CCC                ICOND(J)=3 CIRCULAR SECTION (J) IS TANGENT TO ADJACENT    NOZ   32
CCC                      STRAIGHT SECTION (NEARER THE THROAT)                NOZ   33
CCC                      AT X=2.54*PAR(2,J)                                  NOZ   34
CCC                ICOND(J)=4 CIRCULAR SECTION (J) IS USED TO BREAK A        NOZ   35
CCC                      SHARP ANGLE BETWEEN TWO ADJOINING STRAIGHT          NOZ   36
CCC                      SECTIONS WHICH INTERSECT AT X=2.54*PAR(2,J)         NOZ   37
CCC   XSTART       LEFT LIMIT ON X FOR CALCULATING NOZZLE PROFILE (INCH)     NOZ   38
CCC   XZEROI       INLET POSITION IN INCHES ABOVE THE THROAT (NEGATIVE)       NOZ   39
CCC   NOZZLE       NOZZLE INDEX FOR NATA                                     NOZ   40
CCC   CARDS        LOGICAL CONTROL - FALSE VALUE SUPPRESSES PUNCH OUTPUT     NOZ   41
CCC   PLOTS        LOGICAL CONTROL - FALSE VALUE SUPPRESSES PLOT              NOZ   42
CCC   ENDJOB       LOGICAL CONTROL FOR JOB TERMINATION                       NOZ   43
C                                                                            NOZ   44
      DIMENSION ORDL(12),ABSL(12),NC(2),LINE(3)                              NOZ   45
      DIMENSION COMMNT(18),XINCH(201),YINCH(201),TITLE(12)                   NOZ   46
      DATA NSECTS,ISHAPE,ICOND /26*0/, PAR /24*0./                           NOZ   47
      DATA ABSL /4*' ',' X',' ','INCH',6*' '/                                NOZ   48
      DATA ORDL /4*' ',' Y',' ','INCH',6*' '/                                NOZ   49
      EQUIVALENCE(COMMNT(1),TITLE(1))                                        NOZ   50
      DATA ENDJOB /.TRUE./, INCR /0,1/                                       NOZ   51
      DATA S2ARR /0.,-1.,1./                                                 NOZ   52
      CALL FRAMEB (IWORDL,PLOTS)                                             NOZ   53
10    IF (IWORDL.EQ.1) GO TO 20                                              NOZ   54
      READ (5,330) (COMMNT(I),I=1,18)                                        NOZ   55
```

```
      GO TO 30                                                          NOZ   56
20    READ (5,340) (COMMNT(I),I=1,12)                                   NOZ   57
C     FIRST 4 CHARACTERS ON THIS CARD SHOULD BE NAME OF FACILITY FOR USENOZ   58
C           IDENTIFICATION IN NATA (DCA, EOS, ETC.)                     NOZ   59
30    READ (5,INPUT)                                                   NOZ   60
      WRITE (6,INPUT)                                                  NOZ   61
C     ZERO ARRAYS                                                      NOZ   62
      DO 40 I=1,12                                                     NOZ   63
      IF (I.NE.12) ATP(I)=0.                                           NOZ   64
      DO 40 J=1,3                                                      NOZ   65
40    PARAM(J,I)=0.                                                    NOZ   66
C                                                                      NOZ   67
C     COMPUTE PROFILE FIT PARAMETERS                                   NOZ   68
C                                                                      NOZ   69
      DO 140 L=1,2                                                     NOZ   70
C     L=1 UPSTREAM                                                     NOZ   71
C     L=2 DOWNSTREAM                                                   NOZ   72
      NS=NSECTS(L)                                                     NOZ   73
      DO 130 K=1,NS                                                    NOZ   74
      IF (L.EQ.2) GO TO 50                                             NOZ   75
      S1=-1.                                                           NOZ   76
      J=NS-K+1                                                         NOZ   77
      ISGN=1                                                           NOZ   78
      GO TO 60                                                         NOZ   79
50    J=NSECTS(1)+K                                                    NOZ   80
      ISGN=-1                                                          NOZ   81
      S1=1.                                                            NOZ   82
60    IC=ICOND(J)                                                      NOZ   83
      JP=J-INCR(L)                                                     NOZ   84
      INDEX=J+ISGN                                                     NOZ   85
      GO TO (70,80,90,100), IC                                         NOZ   86
70    IF (ISHAPE(J).NE.2) GO TO 120                                    NOZ   87
C     THROAT SECTION (ICOND=1)                                         NOZ   88
      PARAM(3,J)=2.54*PAR(1,J)                                         NOZ   89
      PARAM(2,J)=0.                                                    NOZ   90
      PARAM(1,J)=PARAM(3,J)+1.27*DTH                                   NOZ   91
      ATP(JP)=0.                                                       NOZ   92
      GO TO 130                                                        NOZ   93
80    IF (ISHAPE(J).NE.1) GO TO 120                                    NOZ   94
      IF (ISHAPE(INDEX).NE.2.AND.ISHAPE(INDEX).NE.3) GO TO 120         NOZ   95
C     CONDITION ICOND=2                                                NOZ   96
      R=PARAM(3,INDEX)                                                 NOZ   97
      THETA=PAR(1,J)/ACON                                              NOZ   98
      ISH=ISHAPE(INDEX)                                                NOZ   99
      S2=S2ARR(ISH)                                                    NOZ  100
      XC=PARAM(2,INDEX)                                                NOZ  101
      YC=PARAM(1,INDEX)                                                NOZ  102
      STHETA=SIN(THETA)                                                NOZ  103
      CTHETA=COS(THETA)                                                NOZ  104
      TTHETA=STHETA/CTHETA                                             NOZ  105
      X1=XC-S1*S2*R*STHETA                                             NOZ  106
      Y1=S2*R*CTHETA+YC                                                NOZ  107
      PARAM(1,J)=Y1-S1*X1*TTHETA                                       NOZ  108
      PARAM(2,J)=S1*TTHETA                                             NOZ  109
      PARAM(3,J)=0.                                                    NOZ  110
```

```
      ATP(JP)=X1                                                        NOZ 111
      GO TO 130                                                         NOZ 112
90    IF (ISHAPE(J).NE.2.AND.ISHAPE(J).NE.3) GO TO 120                  NOZ 113
      IF (ISHAPE(INDEX).NE.1) GO TO 120                                 NOZ 114
C     CONDITION ICOND=3                                                 NOZ 115
      R=2.54*PAR(1,J)                                                   NOZ 116
      X1=2.54*PAR(2,J)                                                  NOZ 117
      ISH=ISHAPE(J)                                                     NOZ 118
      S2=S2ARR(ISH)                                                     NOZ 119
      THETA=PAR(1,INDEX)/ACON                                           NOZ 120
      Y1=PARAM(1,INDEX)+PARAM(2,INDEX)*X1                               NOZ 121
      PARAM(1,J)=Y1-S2*R*COS(THETA)                                     NOZ 122
      PARAM(2,J)=X1+S1*S2*R*SIN(THETA)                                  NOZ 123
      PARAM(3,J)=R                                                      NOZ 124
      ATP(JP)=X1                                                        NOZ 125
      GO TO 130                                                         NOZ 126
100   IF (ISHAPE(J).NE.2.AND.ISHAPE(J).NE.3) GO TO 120                  NOZ 127
      IF (ISHAPE(INDEX).NE.1) GO TO 120                                 NOZ 128
      INDX2=J-ISGN                                                      NOZ 129
      IF (ISHAPE(INDX2).NE.1) GO TO 120                                 NOZ 130
C     CONDITION ICOND=4                                                 NOZ 131
      R=2.54*PAR(1,J)                                                   NOZ 132
      X1=2.54*PAR(2,J)                                                  NOZ 133
      THETA1=PAR(1,INDEX)/ACON                                          NOZ 134
      THETA2=PAR(1,INDX2)/ACON                                          NOZ 135
      PHI=0.5*(PI-ABS(THETA2-THETA1))                                   NOZ 136
      IF (THETA2.GT.THETA1.AND.ISHAPE(J).EQ.3) GO TO 120                NOZ 137
      IF (THETA2.LT.THETA1.AND.ISHAPE(J).EQ.2) GO TO 120                NOZ 138
      Y1=PARAM(1,INDEX)+PARAM(2,INDEX)*X1                               NOZ 139
      XL=R/SIN(PHI)                                                     NOZ 140
      ATP(JP)=X1-S1*XL*COS(PHI)*COS(THETA1)                             NOZ 141
      PARAM(3,J)=R                                                      NOZ 142
      IF (ISHAPE(J).EQ.3) GO TO 110                                     NOZ 143
      PARAM(2,J)=X1-S1*XL*COS(PHI-THETA1)                               NOZ 144
      PARAM(1,J)=Y1+XL*SIN(PHI-THETA1)                                  NOZ 145
      GO TO 130                                                         NOZ 146
110   PARAM(2,J)=X1+S1*XL*COS(PHI-THETA2)                               NOZ 147
      PARAM(1,J)=Y1-XL*SIN(PHI-THETA2)                                  NOZ 148
      GO TO 130                                                         NOZ 149
120   WRITE (6,350) J                                                   NOZ 150
      CALL EXIT                                                         NOZ 151
130   CONTINUE                                                          NOZ 152
140   CONTINUE                                                          NOZ 153
C                                                                       NOZ 154
C     OUTPUT                                                            NOZ 155
C                                                                       NOZ 156
C     PRINTED OUTPUT                                                    NOZ 157
      IF (IABORDL.EQ.1) GO TO 150                                       NOZ 158
      WRITE (6,360) (COMMNT(I),I=1,18)                                  NOZ 159
      GO TO 160                                                         NOZ 160
150   WRITE (6,370) (COMMNT(I),I=1,12)                                  NOZ 161
160   WRITE (6,380)                                                     NOZ 162
      WRITE (6,390)                                                     NOZ 163
      NS=NSECTS(1)+NSECTS(2)                                            NOZ 164
      DO 180 J=1,NS                                                     NOZ 165
```

```
      IF (J.EQ.NS) GO TO 170                                            NOZ 166
      WRITE (6,400) J,ATP(J),(PARAM(I,J),I=1,3)                         NOZ 167
      GO TO 180                                                         NOZ 168
170   WRITE (6,410) J,(PARAM(I,J),I=1,3)                                NOZ 169
180   CONTINUE                                                          NOZ 170
      IF (.NOT.CARDS) GO TO 240                                         NOZ 171
C     PUNCHED OUTPUT AND PRINTED CARD IMAGES                            NOZ 172
      RTHCM=1.27*DTH                                                    NOZ 173
      XZERO=2.54*XZERO1                                                 NOZ 174
      DO 190 I=1,2                                                      NOZ 175
190   FNSECT(I)=NSECTS(I)+0.1                                           NOZ 176
      DO 200 I=1,12                                                     NOZ 177
      IF (I.GT.NS) ISHAPE(I)=0                                          NOZ 178
200   FISHAP(I)=ISHAPE(I)+0.1                                           NOZ 179
      WRITE (6,420)                                                     NOZ 180
      IF (IMODL.EQ.1) GO TO 210                                         NOZ 181
      PUNCH 520, (COMMNT(I),I=1,16),NOZZLE                              NOZ 182
      WRITE (6,440) (COMMNT(I),I=1,16),NOZZLE                           NOZ 183
      GO TO 220                                                         NOZ 184
210   PUNCH 510, (COMMNT(I),I=1,11),NOZZLE                              NOZ 185
      WRITE (6,430) (COMMNT(I),I=1,11),NOZZLE                           NOZ 186
220   PUNCH 530, NOZZLE,RTHCM,XZERO,(FNSECT(I),I=1,2),NOZZLE            NOZ 187
      WRITE (6,450) NOZZLE,RTHCM,XZERO,(FNSECT(I),I=1,2),NOZZLE         NOZ 188
      PUNCH 540, (FISHAP(I),I=1,12),NOZZLE                              NOZ 189
      WRITE (6,460) (FISHAP(I),I=1,12),NOZZLE                           NOZ 190
      PUNCH 550, (ATP(I),I=1,6),NOZZLE                                  NOZ 191
      WRITE (6,470) (ATP(I),I=1,6),NOZZLE                               NOZ 192
      PUNCH 560, (ATP(I),I=7,11),NOZZLE                                 NOZ 193
      WRITE (6,480) (ATP(I),I=7,11),NOZZLE                              NOZ 194
      DO 230 K=1,6                                                      NOZ 195
      J1=2*K-1                                                          NOZ 196
      J2=J1+1                                                           NOZ 197
      KP=K+3                                                            NOZ 198
      WRITE (6,490) KP,((PARAM(I,J),I=1,3),J=J1,J2),NOZZLE              NOZ 199
230   PUNCH 570, KP,((PARAM(I,J),I=1,3),J=J1,J2),NOZZLE                 NOZ 200
      PUNCH 580, COMMNT(1),NOZZLE                                       NOZ 201
      WRITE (6,500) COMMNT(1),NOZZLE                                    NOZ 202
C     PRINTOUT OF PROFILE                                               NOZ 203
240   WRITE (6,590)                                                     NOZ 204
      WRITE (6,600)                                                     NOZ 205
      DX=0.03                                                           NOZ 206
      DO 310 I=1,201                                                    NOZ 207
      XINCH(I)=XSTART+(I-1)*DX                                          NOZ 208
      X=2.54*XINCH(I)                                                   NOZ 209
      NSM1=NS-1                                                         NOZ 210
      DO 250 J=1,NSM1                                                   NOZ 211
      K=J                                                               NOZ 212
      IF (ATP(J).GT.X) GO TO 260                                        NOZ 213
250   CONTINUE                                                          NOZ 214
      K=NS                                                              NOZ 215
260   ISJ=ISHAPE(K)                                                     NOZ 216
      GO TO (270,280,290), ISJ                                          NOZ 217
270   Y=PARAM(1,K)+PARAM(2,K)*X                                         NOZ 218
      GO TO 300                                                         NOZ 219
280   Y=PARAM(1,K)-SQRT(PARAM(3,K)**2-(X-PARAM(2,K))**2)                NOZ 220
```

```
      GO TO 300                                                          NOZ  221
200   Y=PARAM(1,K)+SQRT(PARAM(3,K)**2-(X-PARA(2,K))**2)                  NOZ  222
300   YINCH(I)=Y/2.54                                                    NOZ  223
      WRITE (6,610) XINCH(I),YINCH(I)                                    NOZ  224
310   CONTINUE                                                           NOZ  225
      IF (.NOT.PLOTS) GO TO 320                                          NOZ  226
C     PLOT OF PROFILE                                                    NOZ  227
      NC(1)=0                                                            NOZ  228
      NC(2)=1                                                            NOZ  229
      LINE(1)=1                                                          NOZ  230
      LINE(2)=2                                                          NOZ  231
      LL=0                                                               NOZ  232
      XLO=XINCH(1)                                                       NOZ  233
      XUP=XINCH(201)                                                     NOZ  234
      YLO=0.                                                             NOZ  235
      YUP=6.                                                             NOZ  236
      CALL GRAPH (1,0.0,201,XINCH,YINCH,YLO,XUP,YLO,YUP,16.,NC,LINE,ORDLNOZ  237
     1,ADSL,TITLE,LL,XVAL,PLABL,ICURVE)                                  NOZ  238
320   IF (.NOT.END JOB) GO TO 10                                         NOZ  239
      IF (PLOTS) CALL FRAME2                                             NOZ  240
      CALL EXIT                                                          NOZ  241
C                                                                        NOZ  242
330   FORMAT (18A4)                                                      NOZ  243
340   FORMAT (12A6)                                                      NOZ  244
350   FORMAT (30H0INCOMPATIBLE DATA FOR SECTION,I3)                      NOZ  245
360   FORMAT (1H1,18A4///)                                               NOZ  246
370   FORMAT (1H1,12A6///)                                               NOZ  247
380   FORMAT (26H NOZZLE PROFILE PARAMETERS)                             NOZ  248
390   FORMAT (1H0,8X,1HJ,17X,3HATP,10X,10HPARAM(1,J),10X,10HPARAM(2,J),1NOZ  249
     1 0X,10HPARAM(3,J))                                                 NOZ  250
400   FORMAT (I10,1P4E20.6)                                              NOZ  251
410   FORMAT (I10,20X,1P3E20.6)                                          NOZ  252
420   FORMAT ////20H DATA CARDS PRODUCED/)                               NOZ  253
430   FORMAT (1X,1HC,5X,11A6,1X,4HNOZZ,I3)                               NOZ  254
440   FORMAT (1X,1HC,5X,16A4,3X,4HNOZZ,I3)                               NOZ  255
450   FORMAT (7X,7HDATA ZP,I2,2H /,F10.5,1H,,F10.5,1H,,F6.1,1H,,F6.1,1H,NOZ  256
     1,20X,4HNOZZ,I3)                                                    NOZ  257
460   FORMAT (6X,1H1,6X,12(F4.1,1H,),1X,4HNOZZ,I3)                       NOZ  258
470   FORMAT (6X,1H2,6(F10.6,1H,),1X,4HNOZZ,I3)                          NOZ  259
480   FORMAT (6X,1H3,5(F10.6,1H,),12X,4HNOZZ,I3)                         NOZ  260
490   FORMAT (6X,I1,6(F10.6,1H,),1X,4HNOZZ,I3)                           NOZ  261
500   FORMAT (6X,1HA,6X,2H4H,A4,1H/,54X,4HNOZZ,I3)                       NOZ  262
510   FORMAT (1HC,5X,11A6,1X,4HNOZZ,I3)                                  NOZ  263
520   FORMAT (1HC,5X,16A4,3X,4HNOZZ,I3)                                  NOZ  264
530   FORMAT (6X,7HDATA ZP,I2,2H /,F10.5,1H,,F10.5,1H,,F6.1,1H,,F6.1,1H,NOZ  265
     1,20X,4HNOZZ,I3)                                                    NOZ  266
540   FORMAT (5X,1H1,6X,12(F4.1,1H,),1X,4HNOZZ,I3)                       NOZ  267
550   FORMAT (5X,1H2,6(F10.6,1H,),1X,4HNOZZ,I3)                          NOZ  268
560   FORMAT (5X,1H3,5(F10.6,1H,),12X,4HNOZZ,I3)                         NOZ  269
570   FORMAT (5X,I1,6(F10.6,1H,),1X,4HNOZZ,I3)                           NOZ  270
580   FORMAT (5X,1HA,6X,2H4H,A4,1H/,54X,4HNOZZ,I3)                       NOZ  271
590   FORMAT ////15H NOZZLE PROFILE)                                     NOZ  272
600   FORMAT (1H0,12X,7HX(INCH),13X,7HY(INCH))                           NOZ  273
610   FORMAT (1P2E20.3)                                                  NOZ  274
      END                                                                NOZ  275-
```

```
      SUBROUTINE FRAMEB (IWORDL,PLOTS)                                  FRI   1
      LOGICAL PLOTS                                                     FRI   2
C                                                                       FRI   3
C     FRAMEB FOR AVCO IBM 360/75                                        FRI   4
C     PRODUCES IDENTIFICATION FRAME FOR PLOTS AND CLOSES PLOT DATA SET  FRI   5
C                                                                       FRI   6
      DIMENSION XNAME(3),XMEMO(2)                                       FRI   7
      DATA XNAME /'  ','BADE','  '/, XMEMO /'NOZF','IT'/, BIN /'29'/    FRI   8
      INCOL=2                                                           FRI   9
      IF (PLOTS) CALL IDFRMV (XNAME,BIN,XMEMO)                          FRI  10
      RETURN                                                            FRI  11
      ENTRY FRAMB2                                                      FRI  12
      CALL PLTND                                                        FRI  13
      RETURN                                                            FRI  14
      END                                                               FRI  15-
```

```
      SUBROUTINE FRAMEB (IWORDL,PLOTS)
C
C     FRAMEB FOR UNIVAC 1108
C     DUMMY ROUTINE
C
      ENTRY FRAMB2
      IWORDL=1
      RETURN
      END
```

```
      FRU  1
      FRU  2
      FRU  3
      FRU  4
      FRU  5
      FRU  6
      FRU  7
      FRU  8
      FRU  9-
```

```
      SUBROUTINE GRAPH (NF,LX,LY,NP,X,Y,XL,   ,YL,YU,DC,NC,LINE,ORDL,ABSLGRA    1
     1,TITL,LL,PVAL,PLABL,ICURVE)                                        GRA    2
      DIMENSION X(120),Y(120),LINE(3),ORDL(12),ABSL(12),TITL(12),        GRA    3
     1     PLABL(3,2),NXY(2),IMAG(2),XYMAG(2),NC(2)                      GRA    4
C     NOTE - 48 CHARACTERS IN ABSCISSA, ORDINATE, AND TITLE LABELS       GRA    5
C     NOTE - 12 CHARACTERS IN EACH OF 2 LINES OF PARAMETER LABEL (PLABL)GRA    6
C     NOTE - CALL SETMIV (30,125,24,24) BEFORE FIRST CALL TO GRAPH,      GRA    7
C            IF PARAMETER TABLE IS TO BE PRINTED                         GRA    8
      IF (NF.EQ.0) GO TO 110                                             GRA    9
      JCURVE=0                                                           GRA   10
      NPL=0                                                              GRA   11
      CALL SMXYV (LX,LY)                                                 GRA   12
      IF (LX.EQ.0) GO TO 10                                              GRA   13
      DX=1.                                                              GRA   14
      NNN=1                                                              GRA   15
      GO TO 20                                                           GRA   16
10    CALL DXDYV (1,XL,XU,DX,NNN,III,NXX,DC,IERR)                        GRA   17
20    IF (LY.EQ.0) GO TO 30                                              GRA   18
      DY=1.                                                              GRA   19
      MMM=1                                                              GRA   20
      GO TO 40                                                           GRA   21
30    CALL DXDYV (2,YL,YU,DY,MMM,JJJ,NYY,DC,IERR)                        GRA   22
40    IF (LX.NE.0.AND.LY.NE.0) GO TO 50                                  GRA   23
      IF (IERR.EQ.0) GO TO 50                                            GRA   24
      WRITE (6,310) XL,XU,YL,YU                                          GRA   25
      RETURN                                                             GRA   26
50    XYMAG(1)=AMAX1(ABS(XL),ABS(XU))                                    GRA   27
      XYMAG(2)=AMAX1(ABS(YL),ABS(YU))                                    GRA   28
      DO 90 I=1,2                                                        GRA   29
      IMAG(I)=ALOG10(XYMAG(I))                                           GRA   30
      IF (IMAG(I).GT.5.OR.IMAG(I).LT.-3) GO TO 80                        GRA   31
      IF (IMAG(I)) 60,60,70                                              GRA   32
60    NXY(I)=IABS(IMAG(I))+4                                             GRA   33
      GO TO 90                                                           GRA   34
70    NXY(I)=IMAG(I)+1                                                   GRA   35
      GO TO 90                                                           GRA   36
80    NXY(I)=-3                                                          GRA   37
90    CONTINUE                                                           GRA   38
      III=-III                                                           GRA   39
      JJJ=-JJJ                                                           GRA   40
      IF (LL.LT.2) GO TO 100                                             GRA   41
      XL=XL-0.022*(XU-XL)                                                GRA   42
      DYL=0.022*(YU-YL)                                                  GRA   43
      YL=YL-DYL                                                          GRA   44
      YU=YU+DYL                                                          GRA   45
100   CALL GRIDIV (1,XL,XU,YL,YU,DX,DY,NNN,MMM,III,JJJ,NXY(1),NXY(2))    GRA   46
      ICURVE=0                                                           GRA   47
110   ICURVE=ICURVE+1                                                    GRA   48
      IF (ICURVE.EQ.1) NPL=NPL+1                                         GRA   49
      JCURVE=JCURVE+1                                                    GRA   50
      IX1=NXV(X(1))                                                      GRA   51
      IY1=NYV(Y(1))                                                      GRA   52
      NC2=NC(2)                                                          GRA   53
      NC1=NC(1)                                                          GRA   54
      IF (NC1) 120,140,130                                               GRA   55
```

```
      NSYMBL=ICURVE                                                    GRA  56
      GO TO 140                                                        GRA  57
  130 NSYMBL=NC1                                                       GRA  58
  140 DO 180 I=1,NP                                                    GRA  59
      IF (NC1.EQ.0) GO TO 150                                          GRA  60
      CALL POINTB (IX1,IY1,NSYMBL,NC2)                                 GRA  61
  150 IF (I.EQ.NP) GO TO 180                                           GRA  62
      I2=I+1                                                           GRA  63
      IF (LINE(1)) 170,170,160                                         GRA  64
  160 CALL LINEB (X(I),Y(I),X(I2),Y(I2),LINE)                          GRA  65
  170 IX1=NXV(X(I2))                                                   GRA  66
      IY1=NYV(Y(I2))                                                   GRA  67
  180 CONTINUE                                                         GRA  68
      IF (JCURVE.GT.1) GO TO 190                                       GRA  69
      CALL CHSIZV (3,3)                                                GRA  70
      CALL RITE2V (80,9,1023,90,2,48,1,ABSL,NLAST)                     GRA  71
      CALL RITE2V (9,80,1023,180,2,48,1,ORDL,NLAST)                    GRA  72
      CALL RITE2V (80,1012,1023,90,2,48,1,TITL,NLAST)                  GRA  73
  190 IF (LL.EQ.0) RETURN                                              GRA  74
      CURVE=ICURVE                                                     GRA  75
      IF (LL.EQ.1) GO TO 200                                           GRA  76
      NXV1=NXV(X(1))                                                   GRA  77
      NYV1=NYV(Y(1))                                                   GRA  78
      IDXV=NXV(X(2))-NXV1                                              GRA  79
      IDYV=NYV(Y(2))-NYV1                                              GRA  80
      HYP=IDXV**2+IDYV**2                                              GRA  81
      HYP=SQRT(HYP)                                                    GRA  82
      IX1=NXV1-16.*IDXV/HYP                                            GRA  83
      IY1=NYV1-16.*IDYV/HYP                                            GRA  84
      CALL LABLV (CURVE,IX1,IY1,2,2,2)                                 GRA  85
  200 IF (ICURVE.GT.1) GO TO 230                                       GRA  86
      IY=950-(NPL-1)*300                                               GRA  87
      IY2=IY+15                                                        GRA  88
      CALL LINEBR (930,IY2,1023,IY2,LINE)                              GRA  89
      DO 220 I=1,2                                                     GRA  90
      DO 210 J=1,2                                                     GRA  91
  210 CALL PRINTV (12,PLABL(J,I),924,IY)                               GRA  92
  220 IY=IY-20                                                         GRA  93
  230 IY=IY-20                                                         GRA  94
      IF (LL.EQ.1) GO TO 240                                           GRA  95
      CALL LABLV (CURVE,900,IY,2,1,2)                                  GRA  96
  240 CALL POINTB (924,IY,NSYMBL,NC2)                                  GRA  97
      IF (PVAL.EQ.0.) GO TO 250                                        GRA  98
      IPVM=ALOG10(ABS(PVAL))                                           GRA  99
      GO TO 260                                                        GRA 100
  250 IPVM=0                                                           GRA 101
  260 IF (IPVM.GT.4.OR.IPVM.LT.-2) GO TO 290                           GRA 102
      IF (IPVM) 270,270,280                                            GRA 103
  270 NCHAR=IABS(IPVM)+5                                               GRA 104
      NDMAX=1                                                          GRA 105
      GO TO 300                                                        GRA 106
  280 NCHAR=5                                                          GRA 107
      NDMAX=IPVM+1                                                     GRA 108
      GO TO 300                                                        GRA 109
  290 NDMAX=1                                                          GRA 110
```

```
      NCHAR=-3                                                          GRA 111
300   CALL LABLV (PVAL,940.,IY,NCHAR,1,NDMAX)                           GRA 112
      RETURN                                                            GRA 113
C                                                                       GRA 114
C                                                                       GRA 115
310   FORMAT (51HOGRID GENERATION IMPOSSIBLE WITH THE PARAMETERS XL=,1PEGRA 116
     110.3,5H, XU=,E10.3,5H, YL=,E10.3,5H, YU=,E10.3)                   GRA 117
      END                                                               GRA 118-
```

```
      SUBROUTINE POINTB (IX,IY,NSYMBL,IDARK)                            POI   1
      DIMENSION CHAR(10),K(10)                                          POI   2
      DATA CHAR /'X','+','D','*','H','C','C','Y','Y','H'/               POI   3
      DATA K /6*90,270,90,270,0/                                        POI   4
      CALL CHSIZV (3,2)                                                 POI   5
      CALL RITE2V (IX,IY,1023,K(NSYMBL),IDARK,1,1,CHAR(NSYMBL),NLAST)   POI   6
      RETURN                                                            POI   7
      END                                                               POI   8-
```

```
      SUBROUTINE LINEB (X1,Y1,X2,Y2,LLINE)                              LIN    1
C     LINE=LLINE(1), IDARK=LLINE(2), IPAR=LLINE(3)                      LIN    2
C     X1     = ABSCISSA OF FIRST POINT                                  LIN    3
C     Y1     = ORDINATE OF FIRST POINT                                  LIN    4
C     X2     = ABSCISSA OF SECOND POINT                                 LIN    5
C     X2     = ORDINATE OF SECOND POINT                                 LIN    6
C     LINE   = 1    CONNECT POINTS WITH A CONTINUOUS LINE               LIN    7
C              = 2    CONNECT POINTS WITH A DASHED LINE                  LIN    8
C              = 3    CONNECT POINTS WITH A STRING OF CHARACTRON SYMBOLS LIN    9
C     IF LINE=1    IPAR IS IGNORED                                      LIN   10
C     IF LINE=2    IPAR = LENGTH OF DASHES IN MILLIMETERS               LIN   11
C     IF LINE=3    IPAR = INTEGER (NS) SELECTING PLOT SYMBOL FROM       LIN   12
C          POINTV TABLE                                                 LIN   13
C     IDARK = NUMBER OF TIMES LINE OR SYMBOLS ARE DRAWN                 LIN   14
      LOGICAL RASTER                                                    LIN   15
      DIMENSION LLINE(3)                                                LIN   16
      RASTER=.FALSE.                                                    LIN   17
      GO TO 10                                                          LIN   18
      ENTRY LINEBR(MX1,MY1,MX2,MY2,LLINE)                               LIN   19
      RASTER=.TRUE.                                                     LIN   20
10    LINE=LLINE(1)                                                     LIN   21
      IF (LINE.LE.0) RETURN                                             LIN   22
      IDARK=LLINE(2)                                                    LIN   23
      IPAR=LLINE(3)                                                     LIN   24
      IF (.NOT.RASTER) GO TO 20                                         LIN   25
      NX1=MX1                                                           LIN   26
      NY1=MY1                                                           LIN   27
      NX2=MX2                                                           LIN   28
      NY2=MY2                                                           LIN   29
      GO TO 30                                                          LIN   30
20    NX1=NXV(X1)                                                       LIN   31
      NX2=NXV(X2)                                                       LIN   32
      NY1=NYV(Y1)                                                       LIN   33
      NY2=NYV(Y2)                                                       LIN   34
30    IF (LINE.GT.1) GO TO 50                                           LIN   35
      DO 40 I=1,IDARK                                                   LIN   36
40    CALL LINEV (NX1,NY1,NX2,NY2)                                      LIN   37
      RETURN                                                            LIN   38
50    SQ=(NX2-NX1)**2+(NY2-NY1)**2                                      LIN   39
      DIST=SQRT(SQ)                                                     LIN   40
      VX1=NX1                                                           LIN   41
      VY1=NY1                                                           LIN   42
      COSA=(NX2-NX1)/DIST                                               LIN   43
      SINA=(NY2-NY1)/DIST                                               LIN   44
      IF (LINE.GT.2) GO TO 80                                           LIN   45
      DEL=10.8*IPAR                                                     LIN   46
      NP=DIST/DEL                                                       LIN   47
      HDEL=0.5*DEL                                                      LIN   48
      IDX=HDEL*COSA                                                     LIN   49
      IDY=HDEL*SINA                                                     LIN   50
      DO 70 I=1,NP                                                      LIN   51
      DO 60 J=1,IDARK                                                   LIN   52
60    CALL LINE2V (NX1,NY1,IDX,IDY)                                     LIN   53
      IF (I.EQ.NP) GO TO 70                                             LIN   54
      VX1=VX1+DEL*COSA                                                  LIN   55
```

```
      VY1=VY1+DEL*SINA                                                  LIN 56
      NX1=VX1                                                           LIN 57
      NY1=VY1                                                           LIN 58
   70 CONTINUE                                                          LIN 59
      RETURN                                                            LIN 60
   80 IF (IPAR.EQ.0) GO TO 90                                           LIN 61
      DEL=15.                                                           LIN 62
      GO TO 100                                                         LIN 63
   90 DEL=5.                                                            LIN 64
  100 NP=DIST/DEL                                                       LIN 65
      DO 120 I=1,NP                                                     LIN 66
      DO 110 J=1,IDARK                                                  LIN 67
      CALL POINTV (NX1,NY1,-IPAR,1.)                                    LIN 68
  110 IF (I.EQ.NP) GO TO 120                                            LIN 69
      VX1=VX1+DEL*COSA                                                  LIN 70
      VY1=VY1+DEL*SINA                                                  LIN 71
      NX1=VX1                                                           LIN 72
      NY1=VY1                                                           LIN 73
  120 CONTINUE                                                          LIN 74
      RETURN                                                            LIN 75
      END                                                               LIN 76-
```

REFERENCES

1. Bade, W. L.; and Yos, J. M.: The NATA Code - Theory and Analysis, Volume I. NASA CR-2547, 1975.

2. Bade, W. L.; and Yos, J. M.: The NATA Code - User's Manual, Volume II. NASA CR-141743, 1975.

3. Anon.: Programmers' Reference Manual, S-C 4020 High-Speed Microfilm Recorder. Doc. No. 9500056, Stromberg-Carlson Division of General Dynamics, Oct. 1964.

4. UNIVAC Data Processing Division: 1108 Multi-Processor System, EXEC II Programmers Reference Manual. Univac Division of Sperry Rand Corporation UP-4058, 1966, pp. 3-20 to 3-33.

5. Dewey, C. F., Jr.; and Gross, J. F.: Exact Similar Solutions of the Laminar Boundary Layer Equations. Rep. RM-5089-ARPA, RAND Corp., 1967.

6. Lordi, J. A.; Mates, R. E.; and Moselle, J. R.: Computer Program for the Numerical Solution of Nonequilibrium Expansions of Reacting Gas Mixtures. NASA CR-472, May 1966.

7. International Business Machines Corporation: System/360 Scientific Subroutine Package, Version III, Programmer's Manual, Program No. 360A-CM-03X, Fifth Ed., August 1970.

8. Carnahan, B.; Luther, H. A.; and Wilkes, J. O.: Applied Numerical Methods. John Wiley & Sons, 1969, pp. 282-291.

9. Devoto, R. S.: Transport Coefficients of Partially Ionized Argon. Phys. Fluids, vol. 10, no. 2, Feb. 1957, pp. 354-364.

10. Mason, E. A.; Vanderslice, J. T.; and Yos, J. M.: Transport Properties of High-Temperature Multicomponent Gas Mixtures. Phys. Fluids, vol. 2, no. 6, Nov.-Dec. 1959, pp. 688-694.

11. Monchick, L.: Collision Integrals for the Exponential Repulsive Potential. Phys. Fluids, vol. 2, no. 6, Nov.-Dec. 1959, pp. 695-700.

12. Monchick, L.; and Mason, E. A.: Transport Properties of Polar Gases. J. Chem. Phys., vol. 35, no. 5, Nov. 1961, pp. 1676-1697.

13. Kihara, T.; Taylor, M. H.; and Hirschfelder, J. O.: Transport Properties for Gases Assuming Inverse Power Intermolecular Potentials. Phys. Fluids, vol. 3, no. 5, Sept.-Oct. 1960, pp. 715-720.

14. Winovich, W.: On the Equilibrium Sonic-Flow Method for Evaluating Electric-Arc Air Heater Performance. NASA TN D-2132, Mar. 1964.

15. Leipman, H. W.; and Roshko, A.: Elements of Gas Dynamics. John Wiley & Sons, 1957.

16. Boger, R. C.; and Aiello, G. F.: Hypersonic Flow on Yawed Wedges with Leading-Edge Bluntness and Viscous Interaction. J. Spacecraft and Rockets, vol. 8, no. 7, July 1971, pp. 729-735.

17. Anon: Procedures Manual, Sec. 5.c.7.5.0-SD 4060 System. Computation and Analysis Division - MSC, NASA Manned Spacecraft Center, Apr. 1969.