# NASA Contractor Report 159380

(NASA-CR-159380) A STUDY OF OPTIMUM COWL SHAPES AND FLOW PORT LOCATIONS FOR MINIMUM DRAG WITH EFFECTIVE ENGINE COOLING, VOLUME 2 (North Carolina State Univ.) 203 p HC A10/MF A01 CSCL 01A G3/02
N83-16289
Unclas
02548

## A STUDY OF OPTIMUM COWL SHAPES AND FLOW PORT LOCATIONS FOR MINIMUM DRAG WITH EFFECTIVE ENGINE COOLING - VOLUME II

Stan R. Fox and Frederick O. Smetana

NORTH CAROLINA STATE UNIVERSITY
Raleigh, North Carolina 27650

NASA Grant NSG-1584
November 1980

RELEASE DATE:

PUBLICLY RELEASE ON JANUARY 31, 1983

National Aeronautics and
Space Administration

**Langley Research Center**
Hampton, Virginia 23665

NASA Contractor Report 159380

# A STUDY OF OPTIMUM COWL SHAPES AND FLOW PORT LOCATIONS FOR MINIMUM DRAG WITH EFFECTIVE ENGINE COOLING - VOLUME II


Stan R. Fox and Frederick O. Smetana

NORTH CAROLINA STATE UNIVERSITY
Raleigh, North Carolina 27650



NASA Grant NSG-1584
November 1980

NASA
National Aeronautics and
Space Administration

**Langley Research Center**
Hampton, Virginia 23665

# ABSTRACT

The successful prediction of the performance of a new or modified aircraft depends heavily on an accurate estimation of its lift and drag. This report consists of the listings, user's instructions, sample inputs, and sample outputs of two computer programs which are especially useful in obtaining an approximate solution of the viscous flow over an arbitrary non-lifting three-dimensional body. The first program performs a potential flow solution by a well-known panel method and readjusts this initial solution to account for the effects of the boundary-layer displacement thickness, a nonuniform but unidirectional onset flow field, and the presence of air intakes and exhausts. The second program is effectually a geometry package which allows the user to change or refine the shape of a body to satisfy particular needs without a significant amount of human intervention.

The report represents in part an effort to reduce the cruise drag of light aircraft through an analytical study of the contributions to the drag arising from the engine cowl shape and the forward fuselage area and also that resulting from the cooling air mass flowing through intake and exhaust sites on the nacelle. The programs may be effectively used to determine the appropriate body modifications or flow port locations to reduce the cruise drag as well as to provide sufficient air flow for cooling the engine.

# TABLE OF CONTENTS

# LIST OF FIGURES

# INTRODUCTION

For the past contract year a research program has been conducted for the purpose of reducing the cruise drag of light aircraft for better performance and fuel economy through studying the effects arising from modifications to body shape and the surrounding flow field. The procedures have been described [1] and coded into two FORTRAN computer programs.

The purpose of this report is to present these programs with their complete user's instructions. Sample inputs and outputs are also given to provide references for proper program executions at other computing installations.

The first program - FLOWBODY - performs a potential flow solution by the Hess low-speed panel method [1], [2], [3] and readjusts this initial solution to account for the effects of the boundary-layer displacement thickness, a nonuniform but unidirectional onset flow field, and the presence of air intakes and exhausts. The logic of the program can conveniently be described by the following steps:

(1) The surface of the isolated fuselage is represented by a sufficiently large number of quadrilaterals or four-sided panels.

(2) All four corners of the panel are moved into the same plane through a procedure which determines the direction of the normal.

(3) A nonuniform onset flow field may be superimposed onto the uniform onset flow field.

(4) A source of undetermined strength is placed on each panel, and the prescribed normal boundary condition is required to be satisfied.

(5) The resulting system of equations are solved for the source strengths from which the velocity and pressures over the body surface are calculated.

(6) The system of equations may be resolved for the source strengths to account for the presence of air intakes and exhausts.

(7) Two-dimensional, momentum-integral-type boundary layer computations are performed along the streamlines to find the local values of displacement thickness and wall shear.

(8) The wall shear is integrated over the surface to find the skin friction drag.

(9) The body shape is modified by attaching a wake-body toward the trailing edge and by accounting for the displacement thickness effects.

(10) A new set of source strengths and surface pressures corresponding to the wake-body shape is calculated.

(11) The surface pressures are integrated to find the lift and pressure drage.

(12) The total drag is determined from the sum of the skin friction drag and the pressure drag.

The second program - GRIDPLOT - is a geometry package which may be used to correct body misrepresentations, to change the body geometry, to refine the network or grid of the panels or quadrilaterals that form the surface of the body, and to plot various orthographic, perspective, and stereoscopic views of the original and the modified body. The workhorse of this program is a cubic-spline curve-fitting method coupled to a coordinate-system rotation-translation technique that is very effective in modeling body shapes with regions of high curvature or changes in slope.

The U. S. customary units used in this report reflect those most commonly used in this country by engineers and scientists in the General Aviation field. The reader may choose to use S.I. units in lieu of U. S. customary units in both FLOWBODY and GRIDPLOT programs with only one restriction in the FLOWBODY program. If locations for air intakes and exhausts are to be specified, U. S. customary units must be used throughout FLOWBODY since the constants in the derivations for internal mass flow obtain from U. S. customary units. Otherwise, S.I. units are completely permissible.

# USER'S INSTRUCTIONS - FLOWBODY PROGRAM

The program is written in FORTRAN IV and is designed to execute in single precision on an IBM 370/165 computer with an average execution time of 4 minutes 40 seconds for typically large data set. An average execution requires approximately 426,000 bytes of core storage. The program accepts multiple data sets.

Given a data set describing the half-body* under consideration, the program may be instructed to calculate an approximate solution of the three-dimensional viscous flow over an arbitrary body and to estimate the body lift and drag coefficients with or without

(a) a simulated propeller slipstream, and

(b) mass flow through the body.

The orientation of the body with respect to the body reference axes for the programs is shown in Figure 1.

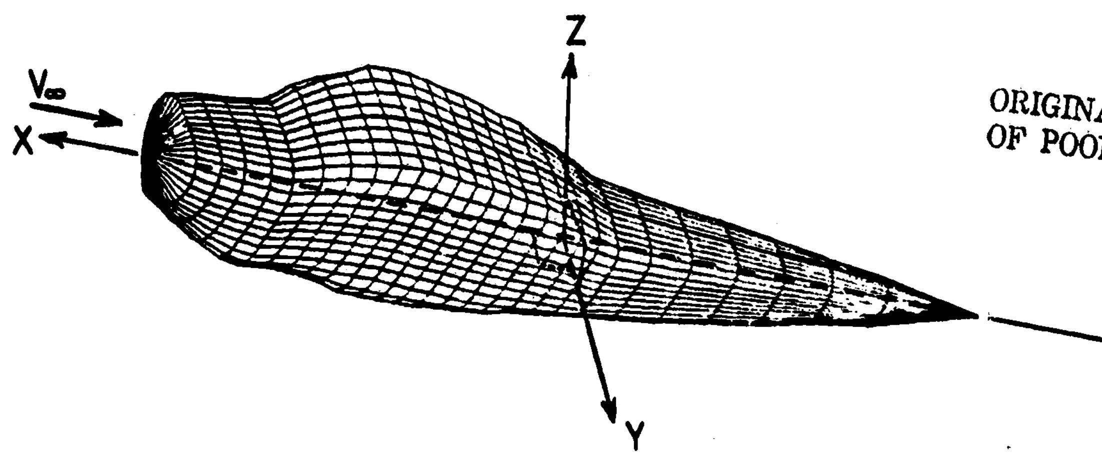





Figure 1: Orientation of the body with respect to the body reference axes

---

*Since the body is considered to be symmetrical about the X-Z plane, only half of the body is needed to describe the entire body.

The program requires the specification of the following input in the indicated order:

CARD 1:

The read unit number IDS:

IDS is a right-adjusted integer number occupying columns 1-5 and specifying that the data is to be read from cards, magnetic tape, disk, etc. The user must supply the suitable job control cards for the specific reads. The IDS parameter controls only the reading of CARD 2, CARD 7, and the Body Description cards.

CARD 2:

The title array TITLE:

The 80 characters of the array TITLE are used for identifying output. The reading of TITLE is controlled by the read unit number IDS.

** The Flow Control Variables **

CARD 3:

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-20 | VINF | Reference free-stream velocity (ft/sec) |
| 21-40 | VO | Kinematic viscosity of the fluid in which the body is moving ($ft^2$/sec) |
| 41-60 | ROE | Density of the fluid in which the body is moving (slug/$ft^3$) |
| 61-80 | REFA | Reference area upon which the aerodynamic coefficients will be based ($ft^2$) |

Parameters VINF, VO, ROE, and REFA are single-precision floating-point numbers in E20 fields.

CARD 4:

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-20 | HVF | Heat of combustion of the fuel being used to develop engine power (BTU/$lb_m$) |
| 21-40 | SFC | Specific fuel consumption of the engine ($lb_m$/ft-$lb_f$) |
| 41-60 | DEP | Developed engine power or power into airstream (ft-$lb_f$/sec) |
| 61-80 | CPHA | Specific heat at constant pressure for air (BTU/$lb_m$ °R) |

Parameters HVF, SFC, DEP, and CPHA are single-precision floating-point numbers in E20 fields.

CARD 5:

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-20 | TINF | Reference free-stream temperature (°R) |
| 21-40 | EOA | Effective orifice area-representative of that area seen by the cooling fluid passing through the body and about the body ($ft^2$) |
| 41-45 | IWRITE | Control variable which denotes the amount of output the user desires. IWRITE = 0 yields the normal maximum output ever desired by the user. IWRITE = 1 deletes information given for each input point. IWRITE = 2 deletes streamline and boundary layer information as well as input point information. IWRITE < 0 generates an enormous amount of output dealing with the streamline calculations, and therefore this option should be used with caution. |
| 46-50 | IPCH1 | Control variable which denotes the punching of the input cards in a form compatible to the NCSU PLOT program of Reference 3. IPCH1 = 0 yields no punched cards, while IPCH1 = 1 produces punched cards. |

CARD 5 (continued):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 51-55 | IPCH2 | Control variable which denotes the punching of cards of the body after the addition of the wake-body in a form compatible to the NCSU PLOT program of Reference 3. IPCH2 = 0 yields no punched cards, while IPCH2 = 1 produces punched cards. |
| 56-60 | IMATCH | Control variable which denotes the matching of the simulated slipstream's power to the specified power into the airstream. With IMATCH = 0, the matching is performed. With IMATCH = 1, no matching occurs and the program utilizes the user-supplied values. Normally, IMATCH = 0 should be specified for an overall program compatibility. |

Parameters TINF and EOA are single-precision floating-point numbers in E20 fields, while parameters IWRITE, IPCH1, IPCH2, and IMATCH are right-adjusted integer numbers in I5 fields.

** Conversion and Test Parameters **

CARD 6:

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-10 | CF | Conversion factor to change the units of the body coordinates points to units of feet. If CF = 0.0, the program automatically sets CF = 1.0. If CF = 1.0, the program assumes that the data units are compatible. |
| 11-15 | ITEST | Control parameter which allows only sufficient information of the propeller-slipstream (or series of vortex rings) calculation to be performed for the plotting of the location and diameter of the vortex rings and the induced velocity of such an arrangement of rings. (See Figure 2). ITEST = 0 produces no plots, while ITEST = 1 does. |

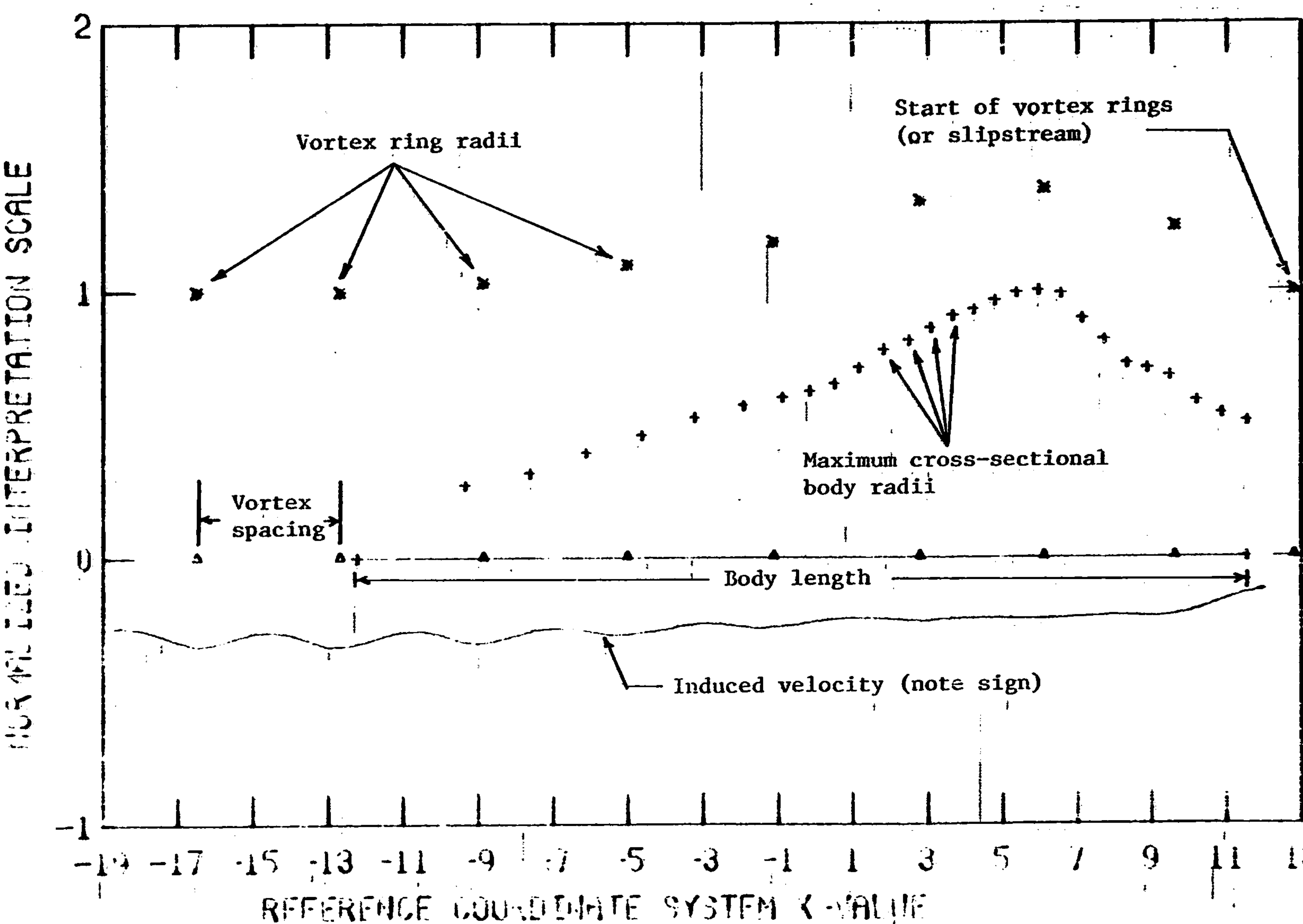



Figure 2: Typical plot of the location and radii of the vortex rings as well as the induced velocity of such an arrangement of rings

CARD 6 (continued):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 16-20 | NODE | Control parameter--equal in value to the node or panel number in question--that generates additional output in that portion of the streamline calculation near the specified node or panel. NODE = 0 produces no additional output, while NODE = "panel number" does. |

Parameter CF is a single-precision floating-point number in a F10 field.

Parameters ITEST and NODE are right-adjusted integer numbers in I5 fields.

CARD 7:

The number NQE of quadrilaterals or panels of the input half-body data:

Occupying columns 1-4 in an I4 field, NQE is a right-adjusted integer number determined by the product

(maximum MI - 1) * (maximum NI - 1)

where MI and NI are defined later. NQE should be restricted in value to approximately 600 because of the array dimensions in the program. The resulting of NQE is controlled by the read unit number IDS.

CARD 8: ** Ring Vortex Systems--Propeller Slipstream **

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-5 | NSRV | Number of independent ring-vortex systems. Normally NSRV = 1 or 0. |
| 6-10 | ISPACE | Generation parameter for ring vortices. If ISPACE = 0, no automatic generation and spacing of ring vortices are performed. If NSRV > 1, ISPACE must be equal to zero. If ISPACE > 0, NSRV is set equal to 1 and the program is permitted to generate a system of ring vortices. |
| 11-20 | XCRP(1) | The approximate vortex spacing along the x-axis before the x-position of maximum body diameter (ft). |

CARD 8 (continued):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 21-30 | XCRP(2) | The approximate vortex spacing along the x-axis aft of the x-position of maximum body diameter (ft). |
| 31-40 | AEXP(1) | Factor in the exponential of the equation $X_i = X_{i-1} - XCRP(1)*EXP(-AEXP(1)*(R-RMX)/RMX)$ where $X_i$, $X_{i-1}$ = x-locations; R = body radius at $X_i$; RMX = maximum body radius; for vortex spacing before the x-location of the maximum body diameter. |
| 41-50 | AEXP(2) | Factor in the exponential of the equation $X_i = X_{i-1} - XCRP(2)*EXP(-AEXP(2)*(R-RMX)/RMX)$ where $X_i$, $X_{i-1}$ = x-locations; R = body radius at $X_i$; RMX = maximum body radius; for vortex spacing aft of the x-location of the maximum body diameter. |
| 51-60 | XRM | Limiting downstream x-location of the axially-spaced vortices (ft). |

NSRV and ISPACE are right-adjusted integer numbers in I5 fields, while XCRP(1), XCRP(2), AEXP(1), AEXP(2) and XRM are single-precision floating-point numbers in F10 fields.

** Ring Vortex Parameters' Cards **

Two cards are necessary to specify the information about the location and orientation of the starting vortex (or vortices) of a generated system of vortices or an individual independent ring vortex. It should be noted that if NSRV = 0, no cards are specified under this section. If NSRV ≠ 0, the user must specify NSRV set(s) of these two cards in the order indicated in the section Circulation Variation Cards.

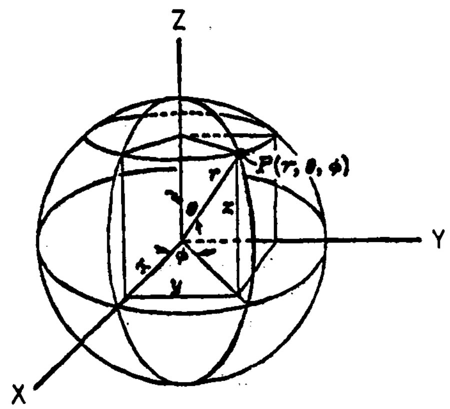
The first card contains

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-10 | XCV | x-coordinate of the center of the vortex ring (ft). |
| 11-20 | YCV | y-coordinate of the center of the vortex ring (ft). |
| 21-30 | ZCV | z-coordinate of the center of the vortex ring (ft). |
| 31-40 | RDM | Maximum vortex radius (ft). RDM is usually considered to be the boundary of the propeller slipstream. If NSRV = 1, ISPACE > 0, and if XCV, YCV, and ZCV correspond to the propeller's center of rotation, RDM is the propeller radius. |
| 41-50 | CA | Initial central angle, degrees (See Figure 3). |
| 51-60 | PHI | Rotation angle, degrees (See Figure 3). |
| 61-65 | NCA | Central angle increment number. NCA is that number dividing 360 degrees into NCA equal parts from which NCA segments are determined to approximate the perimeter of a circle. |
| 66-70 | NVPHI | Rotation tilt parameter.<br>NVPHI = 0: Plane of the vortex ring(s) is perpendicular to the x-axis of body (Set PHI = 90.0).<br>NVPHI = 1: Plane of the vortex ring(s) is not perpendicular to the x-axis of the body. Parameter allows PHI to change at the same rate as CA so that initial choices of PHI and CA determine the path taken by point P of Figure 3. |

first card (continued):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 71-75 | NRS | Number of radial stations or concentric vortex rings. |
| 76-80 | NPTS | Number of (circulation vs. radial distance) points ($\leq$ 50). |

The parameters XCV, YCV, ZCV, RDM, CA, and PHI are single-precision floating-point numbers in F10 fields. The parameters NCA, NVPHI, NRS, and NPTS are right-adjusted integer numbers in I5 fields.




CA = $\theta$

PHI = $\phi$

Figure 3: Definition of angles for specification of vortex ring(s) orientation (Point P described in Cartesian coordinates by $x = r \sin\theta \cos\phi$, $y = r \sin\theta \sin\phi$, and $z = r \cos\theta$)

The second card contains

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-10 | GFACT | The circulation-strength scale factor. If GFACT = 0.0, the program automatically sets GFACT = 1.0. Since an arbitrary normalized circulation variation may be specified that may not be compatible to the specified DEP, GFACT provides an easy means to adjust power into the air-stream by changing the magnitude of the circulation variation. If IMATCH = 0, GFACT is automatically adjusted to match |

second card (continued):

| | | |
|---|---|---|
| | | the simulated slipstream's power to the specified power into the airstream. GFACT is a single-precision floating-point number in an F10 field. |

** Circulation Variation Cards **

For each independent vortex system, a set of NPTS cards must be specified in this section. Each card of the set contains

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-10 | RAD | The radius at which the circulation (or strength) of the vortex ring is to be specified (ft). |
| 11-20 | GAM | The circulation (or strength) of the vortex ring at radius RAD. |

The parameters RAD and GAM are single-precision floating-point numbers in F10 fields.

Ordering of cards: For each of the NSRV vortex systems, the user must supply the two cards from the Ring-Vortex-Parameters' section first and, secondly, the NPTS cards of this section. For every independent ring-vortex system, the user must repeat this sequence.

** Body Description Cards **

Each card contains the information to specify one half-body point. Each card contains

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-12 | XI | x-coordinate |
| 13-24 | YI | y-coordinate |
| 25-36 | ZI | z-coordinate |

Body Description Cards (continued):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 37-40 | NI | N-station index (See Figure 4) |
| 41-44 | MI | M-station index (See Figure 4) |
| 45-48 | NS | Body number |

XI, YI and ZI are single-precision floating-point numbers in F12 fields, while NI, MI and NS are right-adjusted integer numbers in I4 fields. The maximum value of NI or MI must be restricted to less than or equal to 30 because of the array dimensions of the program. NS should be a constant for a given data set, which must be greater than zero but no equal to 1000. A blank card must be supplied at the end of these cards to signal the end of the body description cards. The reading of the body description cards is controlled by the read unit number IDS.

** Inlet and Exhaust Panel Cards **

Each card contains the information to specify one panel or quadrilateral either as an inlet (flow into the body) panel or as an exhaust (flow out of the body) panel. Each card contains

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-5 | MS1 | 1st reference M-station index [MS1 $\geq$ 1] |
| 6-10 | MS2 | 2nd reference M-station index [MS1 < MS2 $\leq$ max (MI)] |
| 11-15 | NS1 | 1st reference N-station index [NS1 $\geq$ 1] |
| 16-20 | NS2 | 2nd reference N-station index [NS1 < NS2 $\leq$ max (NI)] |

| Columns | FORTRAN Name | Description |
| --- | --- | --- |
| 21-25 | IPNS | Parameter which denotes the panel enclosed by the M- and N-station indexes as an inlet or exhaust panel. If IPNS = -1, the panel is designated as an inlet panel, while the panel is designated as an exhaust panel if INPS = +1. If IPNS = 0, the panel is bypassed as being impermeable to mass flow. |

The panels are designated by the order in which the data cards are encountered. It must be true--with the exception of a blank card--that

$$(MS2 - MS1) = 1$$

and

$$(NS2 - NS1) = 1$$

since only one panel specification is allowed per card. Since stagnation flows are not permitted in the program, at least one exhaust panel is required if at least one inlet panel is specified, and vice versa. MS1, MS2, NS1, NS2, and IPNS are right-adjusted integer numbers in I5 fields. A blank or zero card must be supplied to serve either of two purposes. If no inlet and exhaust panels are specified, the blank card terminates the attempt to read more cards. If inlet and exhaust panels are specified, the blank card signals the end of this information.

Specification of the cards above represent a complete set of data for a particular body. Additional data sets are programmed similarly starting again at CARD 1.

ORIGINAL PAGE IS
OF POOR QUALITY

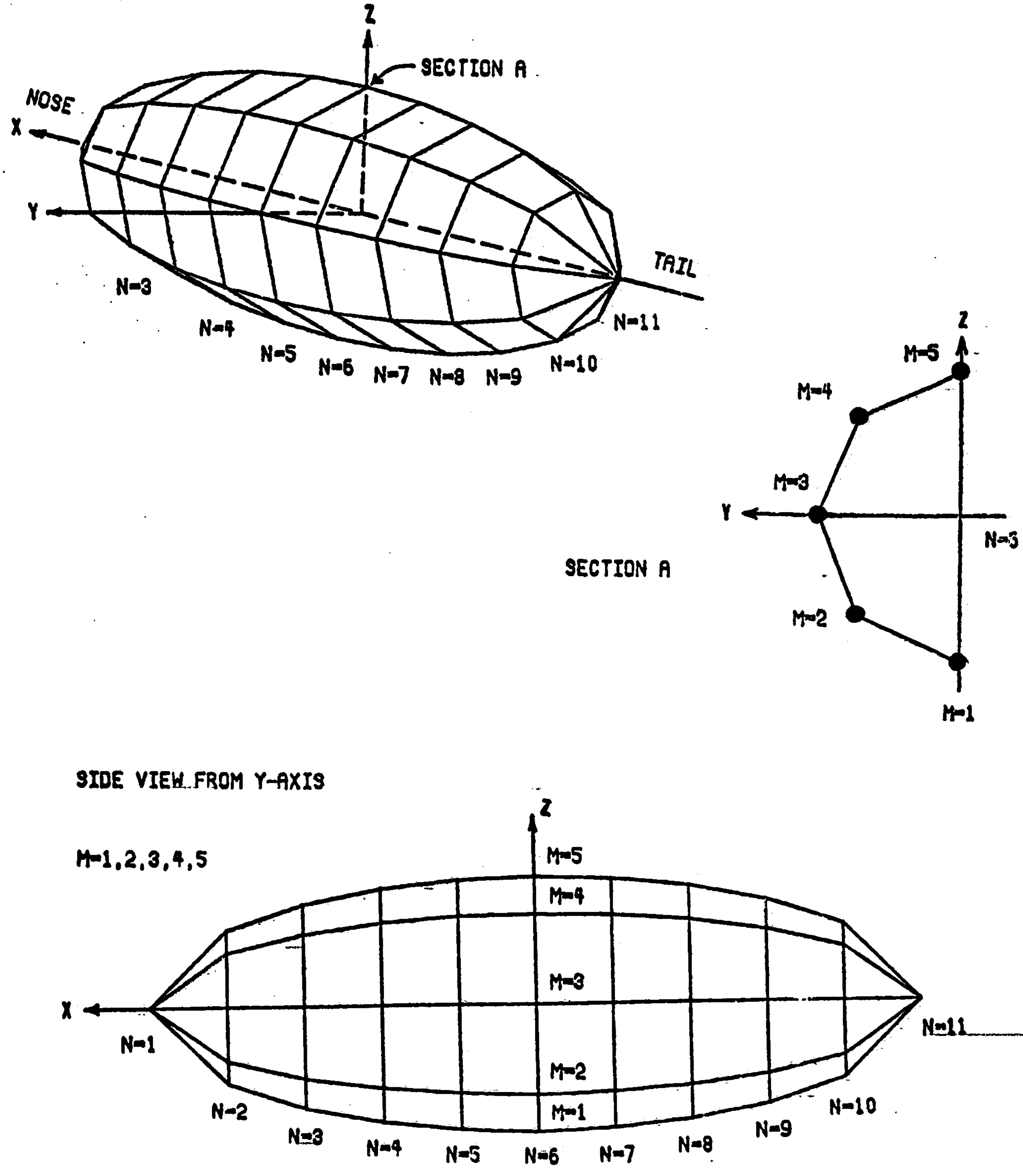



Figure 4: Schematic of indexing scheme

ORIGINAL PAGE IS OF POOR QUALITY

# PROGRAM LISTING - FLOWBODY

```
C
C***   THIS BASIC PROGRAM WAS OBTAINED FROM THE NAVAL SHIP RESEARCH AND
C***   DEVELOPMENT CENTER AND WAS THEN EXTENSIVELY MODIFIED FOR USE AT
C***   N.C. STATE UNIVERSITY BY F.O. SMETANA, ET AL AND BY STAN R. FOX.
C***
C***   REFERENCES: THE XYZ POTENTIAL FLOW PROGRAM
C***               CHARLES W. DAWSON & JANET S. DEAN
C***               REPORT 3892
C***               NAVAL SHIP RESEARCH & DEVELOPMENT CENTER
C***               BETHESDA, MARYLAND  20034
C***
C***               THE NCSU BODY PROGRAM
C***               F.O. SMETANA, ET AL
C***               NASA CR-2523
C***               N.C.STATE UNIVERSITY
C***               RALEIGH, NORTH CAROLINA  27650
C***
C***
C***   ANY QUESTIONS CONCERNING THE USE OF THIS PROGRAM SHOULD BE
C***   DIRECTED TO STAN R. FOX OR F.O. SMETANA
C***               2404 BROUGHTON BLDG
C***               DEPT ME ENGRG. NCSU
C***               RALEIGH, N.C.  27650       (919/737-2374)
C***
C
C
      DIMENSION INDEX(9,3),C(9,6),D(9),CZ(9),IP(9),XP(9),YP(9),ZP(9),X(6
     150),Y(650),Z(650),ID(31,31),B(192),FF(9,1),BHOLD(192),XNEW(150),YN
     1EW(150),ZNEW(150),XK(4),RK(4),IDD(650,4),IPT(4),XS(651),YS(651),ZS
     1(651),XCV(40),YCV(40),ZCV(40),FDM(40),CA(40),PHI(40),NCA(40),NVPHI
     1(40),NFS(40),NPTS(40),RAD(50),GAM(50),SVV(31,2),XCFP(2),AFXP(2)
      DIMENSION XORD(5,200),YORD(5,200),XDUM(5,200),YDUM(5,200),KIND(5)
      EQUIVALENCE (XNEW(1),DELS(1)),(YNEW(1),CFI(1)),(ZNEW(1),THT(1)),(C
     17(1),D(1)),(D(1),FF(1,1))
      COMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP(
     1650),AOP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW
      COMMON /S2V/CIRCV(40)
```

```
      CCMMON /S1K/SN(650),VNP(650),IPN(650)
      COMMON /RING/VTX(3,650)
      CCMMON /VCK/HVF,SFC,DEP,CPHA,TINF,AOUT,AIN,EOA
      COMMON /BL/VOV(75),SS(75),VINF,VO,ROE,DELS(150),CFI(150),THT(150),
     1STOTAL,KKK
      CCMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE
     15
      CCMMON /CANCEL/ICANCL,NODE
      INTEGER P,P1,P2,P3,P4,PC,P5,P6,P7,P8,P9
C
C***  SET CARRIAGE CCNTROL PARAMETERS FOR INSTALLATION
      JREAD=1
      JPUNCH=2
      JWRITE=3
      KFILE1=7
      KFILE2=8
      KFILE3=9
      KFILE4=10
      KFILE5=11
C***  SFT INITIAL PARAMETERS
      PI=3.1415927
      IPLOT=0
      LINE=0
      MXLINE=50
C***  READ DATA CARDS
    1 READ (JREAD,2, END=173) IDS
    2 FORMAT (I5)
      ICANCL=0
      READ (IDS,3) (TITLE(I),I=1,20)
    3 FCRMAT (20A4)
      WRITE (JWRITE,4)
    4 FORMAT (1H1,//,10X,105HNCSU FLOWBODY PROGRAM:  POTENTIAL FLOW + BO
     1UNDARY LAYER + NONUNIFORM SLIPSTREAM FLOW + INTERIOR MASS FLOW,//)
     1
      KFILE=KFILE1
      REWIND KFILE1
      REWIND KFILE2
      REWIND KFILE3
      REWIND KFILE4
      REWIND KFILE5
      KNEW=1
      ICK=3
```

```
      KWRITE=0
      JHOLD=0
      QHOLD=1.0E44
      J=0
      WRITE (JWRITE,5) (TITLE(I),I=1,20)
    5 FORMAT (1X,20A4)
      SA=.0
      IMATCH=0
      LMATCH=0
C***  DEFINE AND READ PERTINENT GEOMETRIC & FLOW CONTROL PARAMETERS
      READ (JREAD,6) VINF,VO,ROE,REFA,HVF,SFC,DEP,CPHA,TINF,EOA,IWRITE,I
     1PCH1,IPCH2,IMATCH
    6 FORMAT (4E20.13,/,4E20.13,/,2E20.13,4I5)
      READ (JREAD,7) CF,ITEST,NODE
    7 FORMAT (F10.0,2I5)
      IF (CF.EQ.0.0E0)CF=1.0E0
      READ (IDS,8) NQE
    8 FORMAT (I4)
      NSF=1
      MIX=150
      ISM=1
      EPS=0.0001
      ISP=0
      WRITE (JWRITE,9) NQE,NSF,MIX
    9 FORMAT (17HONO. OF QUADS.    =,I4/17H NO. OF SECTIONS=,I4/31H MAX. N
     1O. OF ITERATIONS X FLOW ,I3)
      WRITE (JWRITE,10) VINF,VO,ROE,REFA,HVF,SFC,DEP,CPHA,TINF,EOA,IWRIT
     1E,IPCH1,IPCH2,IMATCH,IDS,ITEST,NODE
   10 FORMAT (//,2X,7HVINF = ,E14.7,3X,5HVO = ,E14.7,3X,6HROE = ,E14.7,3
     1X,7HREFA = ,F14.7,3X,6HHVF = ,F14.7,//,2X,6HSFC = ,F14.7,3X,6HDEP
     1= ,E14.7,3X,7HCPHA = ,E14.7,3X,7HTINF = ,E14.7,3X,6HEOA = ,E14.7,/
     1/,2X,9HIWRITE = ,I2,3X,8HIPCH1 = ,I2,3X,8HIPCH2 = ,I2,3X,9HIMATCH
     1= ,I2,3X,6HIDS = ,I2,3X,8HITEST = ,I2,3X,7HNODE = ,I4)
C***  PRINT COORDINATE CONVERSION FACTOR
      WRITE (JWRITE,11) CF
   11 FORMAT (1X,/,2X,29HCOORDINATE CONVERSION FACTOR,,1X,F12.7,/)
      WRITE (JWRITE,12) ISM,EPS
   12 FORMAT (1X,/,2X,I1,18H PLANE OF SYMMETRY,/,2X,22HCONVERGENCE CRITE
     1RIA ,,F8.5)
      LINE=LINE+23
      NQES=NQE
C***  READ PERTINENT DATA FOR THE VORTEX FLOW(S)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      NPMAX=650
      DO 13 LL=1,NPMAX
      DC 13 I=1,3
   13 VTX(I,LL)=0.0
C***  READ NUMBER OF VORTEX SYSTEMS
      READ (JRFAD,14) NSRV,ISPACE,XCRP(1),XCRP(2),AEXP(1),AEXP(2),XRM
   14 FORMAT (2I5,5F10.0)
      IF (NSRV.LE.0)ISPACE=0
      IF (ISPACF.NE.0)NSRV=1
      ISEE=0
      IF (ISPACF.NE.0)ISEF=1
      INSRV=NSRV
      IF (NSRV.LE.0) GC TO 24
      WRITE (JWFITE,15) NSRV,ISPACE,XCRP(1),AEXP(1),XCRP(2),AEXP(2),XRM
   15 FCRMAT (1X,//,2X,I2,27H VORTEX SYSTEM(S) SPECIFIED,/,31X,7HISPACE=
     1,I2,/,31X,10H   XCRP(1)=,F10.5,3X,8HAFXP(1)=,F10.5,/,31X,10H   XCRP(
     12)=,F10.5,3X,8HAEXP(2)=,F10.5,5X,4HXRM=,F10.5,//)
      LINE=LINE+6
C***  RFAD COORDINATES FOR VORTEX CENTER, MAXIMUM VORTEX RADIUS,   THE
C***  INITIAL CENTRAL ANGLE, ROTATION ANGLE, CENTRAL-ANGLE-INCREMENT
C***  NUMBER, ROTATICN TILT PARAMETFR, NUMBFR OF RACIAL STATIONS FOR
C***  VORTEX SYSTEM, THE NUMBER OF POINTS OF CIRCULATION VS RADIAL
C***  DISTANCE, AND THE CIRCULATION STRENGTH SCALE FACTOR
      DO 23 LL=1,NSRV
      READ (JREAD,16) XCV(LL),YCV(LL),ZCV(LL),RDM(LL),CA(LL),PHI(LL),NCA
     1(LL),NVPHI(LL),NRS(LL),NPTS(LL),GFACT
   16 FORMAT (6F10.0,4I5,/,F10.0)
      IF (GFACT.EQ.0.0)GFACT=1.0
      IF (RDM(LL).GT.0.0) GO TO 17
      NRS(LL)=1
      NPTS(LL)=0
      ISEF=1
   17 IF ((LINE+7).LE.MXLINE) GO TO 19
      WRITE (JWRITF,18)
   18 FCRMAT (1H1)
      LINF=0
   19 WRITE (JWRITE,20) LL,XCV(LL),YCV(LL),ZCV(LL),RDM(LL),CA(LL),PHI(LL
     1),NCA(LL),NVPHI(LL),NRS(LL),NPTS(LL),GFACT
   20 FORMAT (1X,'FOR VORTFX SYSTFM:',I2,/,4X,'XCV=',F10.5,'   YCV=',F10.
     15,'   ZCV=',F10.5,/,4X,'PDM=',F10.5,'   CA=',F10.5,'   PHI=',F10.5,/,
     14X,'NCA=',I5,'   NVPHI=',I5,'   NRS=',I5,'   NPTS=',I5,/,4X,'GFACT=',
     1E16.9,//)
```

```
      LINE=LINE+7
      CA(LL)=CA(LL)*PI/180.0
      PHI(LL)=PHI(LL)*PI/180.0
      IF (NPTS(LL).EQ.0) GC TO 23
C***  READ SPLINE POINTS FOR CIRCULATICNS
      MNPTS=NPTS(LL)
      DO 22 I=1,MNPTS
      READ (JREAD,21) RAD(I),GAM(I)
   21 FCRMAT (2F10.0)
      GAM(I)=GAM(I)*GFACT
   22 CONTINUE
      CALL SPLINE(MNPTS,GAM,RAD,LL)
   23 CONTINUE
   24 K=0
      IF (NSRV.FQ.1)LMATCH=1
      XMIN=0.0
      XMAX=0.0
      MM=0
      P=1
      Q=1.0
      DO 25 I=1,31
      DC 25 J=1,31
   25 ID(I,J)=0
      J=0
C***  READ INPUT POINTS--AN ODD NUMBER MUST BE USED FOR BOTH NI AND MI
      RFAD (IDS,26) XI,YI,ZI,NI,MI,NS,NE,VN
   26 FCRMAT (3F12.9,4I4,F12.9)
      MMAX=MI
      MMIN=MI
      NMAX=NI
      NMIN=NI
      NSS=NS
      PC=1
      GO TO 31
   27 READ (IDS,26) XI,YI,ZI,NI,MI,NS,ME,VN
      GO TO 30
   28 MI=MI+1
      IF (MI.LE.MMAXOD) GO TO 29
      MI=1
      NI=NI+1
      IF (NI.FQ.6)NS=3
      IF (NI.EQ.6) GO TO 35
```

```
   29 XI=XNEW(PC+1)
      YI=YNEW(PC+1)
      ZI=ZNEW(PC+1)
   30 PC=PC+1
      IF (NS.NE.NSS) GO TO 35
   31 IF (KNEW.EQ.1.AND.NE.EQ.0) GO TO 32
      IF (KNEW.NE.1.AND.NE.EQ.0) GO TO 33
      IW=NI
      NI=MI
      MI=IW
C***  CONVERT BODY COORDINATES TO COMPATIBLE UNITS
   32 XI=XI*CF
      YI=YI*CF
      ZI=ZI*CF
C***  STORE INPUT POINTS IN POINT ARRAY
   33 X(PC)=XI
      Y(PC)=YI
      Z(PC)=ZI
      IF (KNEW.NE.1) GO TO 34
      XS(PC)=X(PC)
      YS(PC)=Y(PC)
      ZS(PC)=Z(PC)
   34 ID(MI,NI)=PC
      MMAX=MAX0(MMAX,MI)
      MMIN=MIN0(MMIN,MI)
      NMAX=MAX0(NMAX,NI)
      NMIN=MIN0(NMIN,NI)
      IF (XMIN.GE.XI)XMIN=XI
      IF (XMAX.LE.XI)XMAX=XI
      GO TO (27,28),KNEW
   35 IF ((LINE+3).LE.MXLINE) GO TO 36
      WRITE (JWRITE,18)
      LINE=0
   36 WRITE (JWRITE,37) NSS
   37 FORMAT (/,10H0 SECTION ,I4)
      IF (ISEF.EQ.0.OR.NSRV.EQ.0) GO TO 66
      PC=PC-1
C***  DEFINE CENTERLINE OF BODY
      X21=X(PC)-X(1)
      Y21=Y(PC)-Y(1)
      Z21=Z(PC)-Z(1)
      BCDC=SQ?(X21,Y21,Z21)
```

```
      XDCS=X21/BCDC
      YDCS=Y21/BCDC
      ZDCS=Z21/BCDC
      WRITE (JWRITE,33) XDCS,YDCS,ZDCS
   38 FORMAT (1X,//,10X,35HBODY-CENTERLINE DIRECTION COSINES( ,F7.4,3H ,
     1 ,F7.4,3H , ,F7.4,2H ))
      IF (ITEST.EQ.0) GO TO 39
      XORD(2,1)=X(1)
      XORD(2,2)=X(PC)
      YORD(2,1)=0.0
      YORD(2,2)=0.0
      KIND(2)=-1
      KIND(3)=3
C***  FOR EACH N-STATION, DETERMINE CROSS-SECTIONAL AREA AND MAX RADIUS
   39 LPT=1
      XMR=X(1)
      RXMR=0.0E0
      MMAXM1=MMAX-1
      LZP=0
      DO 41 JN=NMIN,NMAX
      LZP=LZP+1
      X3=X(LPT)
      XORD(3,JN)=X3
      Y3=Y(1)+(X3-X(1))*Y21/X21
      Z3=Z(1)+(X3-X(1))*Z21/X21
      AREAS=0.0E0
      DO 40 JM=MMIN,MMAXM1
      Y4=Y(LPT)
      Z4=Z(LPT)
      IF (JM.EQ.MMIN) RMX=SQ2F(0.0,0.0,Y3,Y4,Z3,Z4)
      Y5=Y(LPT+1)
      Z5=Z(LPT+1)
      AREA=0.5E0*ABS(Y3*Z4+Z3*Y5+Z5*Y4-(Z4*Y5+Z3*Y4+Y3*Z5))
      AREAS=AREAS+2.0E0*AREA
      RMX1=SQ2F(0.0,0.0,Y3,Y5,Z3,Z5)
      IF (RMX1.GT.RMX)RMX=RMX1
      IF (RMX.LT.1.0E-06)RMX=0.0E0
      IF (RMX.GT.RXMR)RXMR=RMX
      IF (RMX.EQ.RXMR)XMR=X3
   40 LPT=LPT+1
      LPT=LPT+1
      SVV(JN,1)=AREAS
```

```
      SVV(JN,2)=RMX
   41 CONTINUE
      IROUND=0
      IF (ITEST.EQ.0) GO TO 43
      DO 42 JN=1,LZP
   42 YORD(3,JN)=SVV(JN,2)/RXMR
   43 IF (ISPACE.EQ.0) GO TO 45
C***  COMPUTE LOCATION OF VORTICES
      CALL SPACF(NSRV,SVV,XCV,NMIN,NMAX,X-MMAX,XCRP,AEXP,XRM,ISPACE,RXMR
     1,XMR)
      INSRV=NSRV
      DO 44 JV=2,NSRV
      NRS(JV)=1
      NPTS(JV)=0
      CA(JV)=CA(1)
      PHI(JV)=PHI(1)
      NCA(JV)=NCA(1)
      RDM(JV)=0.0
   44 NVPHI(JV)=NVPHI(1)
   45 IF ((LINE+25).LE.MXLINE) GO TO 46
      WRITE (JWRITE,18)
      LINE=0
   46 IF (LMATCH.NE.1) GO TO 47
      IF (IMATCH.EQ.0) GO TO 49
   47 WRITE (JWRITE,48)
   48 FORMAT (1X,//,5X,24HRING VORTEX INFORMATION:,//,6X,6HNUMBER,9X,18H
     1CENTER COORDINATES,8X,6HRADIUS,3X,11HCIRCULATION,/,19X,1HX,9X,1HY,
     19X,1HZ,/)
      IROUND=1
   49 DO 60 JV=1,NSRV
      IF (JV.GT.1) GO TO 51
C***  COMPUTE A TOTAL CIRCULATION
      PDR=RDM(JV)
      NR=NRS(JV)
      GGDR=0.0
      DO 50 MV=1,NR
      RVX=F(NR,MV,PDR,DRX)/PDR
   50 GGDR=GGDR+G(JV,RVX)/RVX
      CIRCV(JV)=GGDR
   51 IF (RDM(JV).NE.0.0.AND.JV.NE.1.AND.IMATCH.NE.0) GO TO 60
C***  CENTER VORTEX ON BODY CENTERLINE
      X3=XCV(JV)
```

```
      IF (ITEST.EQ.0) GO TO 52
      XORD(1,JV)=XCV(JV)
      YORD(1,JV)=0.0
      KIND(1)=2
   52 YCV(JV)=Y(1)+(X3-X(1))*Y21/X21
      ZCV(JV)=Z(1)+(X3-X(1))*Z21/X21
      IF (JV.EQ.1) GO TO 56
C***  SCAN N-STATIONS FOR VORTEX LOCATION
      LPT=1
      NMAXM1=NMAX-1
      DO 53 JN=NMIN,NMAXM1
      LC1=JN
      LC2=JN+1
      IF (XCV(JV).LE.X(LPT).AND.XCV(JV).GE.X(LPT+MMAX)) GO TO 54
   53 LPT=LPT+MMAX
      RDM(JV)=RDM(1)
      GO TO 55
C***  VORTEX LOCATED, DETERMINE RADIUS
   54 RXTL=(X(LPT)-XCV(JV))/(X(LPT)-X(LPT+MMAX))
      RDM(JV)=SVV(LC1,2)+RXTL*(SVV(LC2,2)-SVV(LC1,2))
      RAV1=SQRT(SVV(LC1,1)/PI)
      RAV2=SQRT(SVV(LC2,1)/PI)
      RAV=RAV1+RXTL*(RAV2-RAV1)
      RAVD=SQRT(PDR**2+RAV**2)
      IF (RAVD.GT.RDM(JV))RDM(JV)=RAVD
   55 CIRCV(JV)=GGDR/(RDM(JV)/PDR)
   56 IF (ITEST.EQ.0) GO TO 57
      XORD(4,JV)=XCV(JV)
      YORD(4,JV)=RDM(JV)/PDR
      KIND(4)=6
   57 IF (LMATCH.NE.1) GO TO 58
      IF (IMATCH.EQ.0.AND.IROUND.EQ.0) GO TO 60
   58 WRITE (JWRITE,59) JV,XCV(JV),YCV(JV),ZCV(JV),RDM(JV),CIRCV(JV)
   59 FORMAT (8X,I2,5X,F9.5,1X,F9.5,1X,F9.5,1X,F9.5,2X,F11.7)
   60 CONTINUE
      IF (IROUND.EQ.1) GO TO 64
      IROUND=1
      CALL RGVRTX(XCV,YCV,ZCV,XCV(1),YCV(1),ZCV(1),RDM,NCA,CA,PHI,NVPHI,
     1NRS,NSRV,VX,VY,VZ,IM,NPTS)
      VI=SQ2(VX,VY,VZ)
      IF (IMATCH.NE.0.OR.LMATCH.NE.1) GO TO 64
      DELV=VINF*(1.0+VI)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PNET=ROE*PI*(RDM(1)*VINF)**2*DELV
      GFACT=DFP/PNET
      WRITE (JWRITE,61) VI
   61 FORMAT (1X,//,5X,29HREFERENCE INDUCED VELOCITY = ,E16.9)
      WRITE (JWRITE,62) GFACT
   62 FORMAT (1X,//,5X,80HMATCHING OF AIRSTREAM POWER TO DEVELOPED ENGIN
     1E POWER(DFP) REQUIRED GFACT TO BE ,E16.9,20H TIMES THE ORIGINAL.)
      DO 63 I=1,MNPTS
   63 GAM(I)=GAM(I)*GFACT
      CALL SPLINE(MNPTS,GAM,RAD,1)
      GO TO 47
   64 WRITE (JWRITE,65)
   65 FORMAT (1X,//)
      IF (ITEST.EQ.0) GO TO 66
C***  DETERMINE PARAMETERS FOR PLOT OF EXTENT OF VORTICES
      XKEEP=ABS(X21)
      XHOLD=ABS(XRM-X(1))
      IF (XHOLD.GT.XKEEP)XKEEP=XHOLD
      IKEEP1=IFIX(XRM-1.0)
      IKEEP2=IFIX(X(1)+2.0)
      IF (((2*IKEEP1)/2).NE.IKEEP1)IKEEP1=IKEEP1+1
      IF (((2*IKEEP2)/2).NE.IKEEP2)IKEEP2=IKEEP2+1
      LTEST=IABS(IKEEP1)+IABS(IKEEP2)
      XINC=2.0
      XKEEP1=FLOAT(IKEEP1)
      XKEEP2=FLOAT(IKEEP2)
      IHLD1=IFIX(X(1))
      IHLD2=IFIX(XKEEP)
      XHLD=FLOAT(IHLD1)+1.0
      LL1=1
      LL2=2*(IABS(IHLD1)+IABS(IHLD2))+1
      GO TO 111
   66 IF (IWRITE.GT.0) GO TO 68
C***  SET LINE COUNTER
      WRITE (JWRITE,18)
      LINE=0
      WRITE (JWRITE,67)
   67 FORMAT (4H0  M,7X,2HX1,12X,2HX2,12X,2HX3,12X,2HX4,12X,2HXC,12X,2HX
     1N,12X,1HA,13X,3HCZ4/4H   N,7X,2HY1,12X,2HY2,12X,2HY3,12X,2HY4,12X,
     12HYC,12X,2HYN,12X,2HFL,12X,3HCZ5/4H   P,7X,2HZ1,12X,2HZ2,12X,2HZ3,
     112X,2HZ4,12X,2HZC,12X,2HZN,12X,4HCZ1 ,10X,3HCZ6/)
      LINE=LINE+5
```

```
   68 IF (KNEW.EQ.2) GO TO 69
      STCTAL=XMAX-XMIN
      PEBODY=VINF*STOTAL/VO
   69 MMAXQD=MMAX
      NMAXQD=NMAX
      IF (KNEW.NE.1) GO TO 70
      NMX=NMAX+2
   70 IDD=(NMAXQD-3)*(MMAXQD-1)/12
      N1=NMIN
      MM2=MMAX-MMIN
      NN2=NMAX-NMIN
      IF (MOD(MM2,2).EQ.0.AND.MOD(NN2,2).EQ.0) GO TO 71
      ISP=1
   71 MM2=MM2/2
      NN2=NN2/2
      IF ((KNEW.EQ.1.AND.IPCH1.EQ.0).OR.(KNEW.EQ.2.AND.IPCH2.EQ.0)) GO T
     1O 72
C***  PUNCH CARDS FOR PLOTTING
      CALL PUNCH(NMIN,NMX,MMIN,MMAX,XS,YS,ZS,JPUNCH,JWRITE)
C***  DC LOOPS WHICH SWEEP POINT ARRAYS & SET UP PANEL GEOMETRY
   72 DO 110 NN=1,NN2
      M1=MMIN
      DO 109 MM=1,MM2
      NQ=1
      IT=ID(M1,N1)*ID(M1+1,N1)*ID(M1+2,N1)*ID(M1,N1+1)*ID(M1+1,N1+1)*ID(
     1M1+1,N1+2)*ID(M1,N1+2)*ID(M1+1,N1+2)*ID(M1+2,N1+2)
      IF (IT.EQ.0) GO TO 109
      M2=M1+1
      DO 108 M=M1,M2
      N2=N1+1
      DC 108 N=N1,N2
      GC TO (73,74,75,76),NQ
   73 P1=ID(M,N)
      P2=ID(M+1,N)
      P3=ID(M+1,N+1)
      P4=ID(M,N+1)
      P5=ID(M+2,N)
      P6=ID(M+2,N+1)
      P7=ID(M+1,N+2)
      P8=ID(M,N+2)
      P9=P1
      IF ((X(P1).NE.X(P2).OF.Y(P1).NE.Y(P2).OR.Z(P1).NE.Z(P2)).AND.(X(P1
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
     1).NE.X(P4).OR.Y(P1).NE.Y(P4).OR.Z(P1).NE.Z(P4))) GO TO 77          PF1  458
      P9=ID(M+2,N+2)                                                      PF1  459
      GO TO 77                                                            PF1  460
   74 P1=ID(M,N+1)                                                        PF1  461
      P2=ID(M,N)                                                          PF1  462
      P3=ID(M+1,N)                                                        PF1  463
      P4=ID(M+1,N+1)                                                      PF1  464
      P5=ID(M,N-1)                                                        PF1  465
      P6=ID(M+1,N-1)                                                      PF1  466
      P7=ID(M+2,N)                                                        PF1  467
      P8=ID(M+2,N+1)                                                      PF1  468
      P9=P1                                                               PF1  469
      IF ((X(P1).NE.X(P2).OR.Y(P1).NE.Y(P2).OR.Z(P1).NE.Z(P2)).AND.(X(P1  PF1  470
     1).NE.X(P4).OR.Y(P1).NE.Y(P4).OR.Z(P1).NE.Z(P4))) GO TO 77          PF1  471
      P9=ID(M+2,N-1)                                                      PF1  472
      GC TO 77                                                            PF1  473
   75 P1=ID(M+1,N)                                                        PF1  474
      P2=ID(M+1,N+1)                                                      PF1  475
      P3=ID(M,N+1)                                                        PF1  476
      P4=ID(M,N)                                                          PF1  477
      P5=ID(M+1,N+2)                                                      PF1  478
      P6=ID(M,N+2)                                                        PF1  479
      P7=ID(M-1,N+1)                                                      PF1  480
      P8=ID(M-1,N)                                                        PF1  481
      P9=P1                                                               PF1  482
      IF ((X(P1).NE.X(P2).OR.Y(P1).NE.Y(P2).OR.Z(P1).NE.Z(P2)).AND.(X(P1  PF1  483
     1).NE.X(P4).OR.Y(P1).NF.Y(P4).OR.Z(P1).NF.Z(P4))) GO TO 77          PF1  484
      P9=ID(M-1,N+2)                                                      PF1  485
      GC TO 77                                                            PF1  486
   76 P1=ID(M+1,N+1)                                                      PF1  487
      P2=ID(M,N+1)                                                        PF1  488
      P3=ID(M,N)                                                          PF1  489
      P4=ID(M+1,N)                                                        PF1  490
      P5=ID(M-1,N+1)                                                      PF1  491
      P6=ID(M-1,N)                                                        PF1  492
      P7=ID(M,N-1)                                                        PF1  493
      P8=ID(M+1,N-1)                                                      PF1  494
      P9=P1                                                               PF1  495
      IF ((X(P1).NE.X(P2).OR.Y(P1).NE.Y(P2).OR.Z(P1).NE.Z(P2)).AND.(X(P1  PF1  496
     1).NF.X(P4).OR.Y(P1).NE.Y(P4).OR.Z(P1).NF.Z(P4))) GO TO 77          PF1  497
      P9=ID(M-1,N-1)                                                      PF1  498
   77 IP(1)=P1                                                            PF1  499
```

```
      IP(2)=P2                                                          PF1 500
      IP(3)=P3                                                          PF1 501
      IP(4)=P4                                                          PF1 502
      IP(5)=P5                                                          PF1 503
      IP(6)=P6                                                          PF1 504
      IP(7)=P7                                                          PF1 505
      IP(8)=P8                                                          PF1 506
      IP(9)=P9                                                          PF1 507
C***  STORE CORNER POINTS TO PANEL NUMBER                               PF1 508
      DO 78 JX=1,4                                                      PF1 509
   78 IDD(P,JX)=IP(JX)                                                  PF1 510
C***  COMPUTE NORMAL VECTOR TO PANEL (IN TERMS OF REFERENCE COORDINATE   PF1 511
C***  SYSTEM)                                                           PF1 512
      X1=X(P3)-X(P1)                                                    PF1 513
      X2=X(P4)-X(P2)                                                    PF1 514
      Y1=Y(P3)-Y(P1)                                                    PF1 515
      Y2=Y(P4)-Y(P2)                                                    PF1 516
      Z1=Z(P3)-Z(P1)                                                    PF1 517
      Z2=Z(P4)-Z(P2)                                                    PF1 518
      XN=Y2*Z1-Y1*Z2                                                    PF1 519
      YN=X1*Z2-X2*Z1                                                    PF1 520
      ZN=X2*Y1-X1*Y2                                                    PF1 521
      R=SQ2(XN,YN,ZN)                                                   PF1 522
      XN=XN/R                                                           PF1 523
      YN=YN/R                                                           PF1 524
      ZN=ZN/R                                                           PF1 525
      AQ=.5*R                                                           PF1 526
C***  COMPUTE PANEL CENTROID (IN TERMS OF REFERENCE COORDINATE SYSTEM)   PF1 527
      X1=X(P3)-X(P2)                                                    PF1 528
      Y1=Y(P3)-Y(P2)                                                    PF1 529
      Z1=Z(P3)-Z(P2)                                                    PF1 530
      X5=Y1*Z2-Y2*Z1                                                    PF1 531
      Y5=Z1*X2-Z2*X1                                                    PF1 532
      Z5=X1*Y2-X2*Y1                                                    PF1 533
      A1=SQ2(X5,Y5,Z5)                                                  PF1 534
      A2=R-A1                                                           PF1 535
      IT=1                                                              PF1 536
      XC=(X(P2)+X(P4)+(A1*X(P3)+A2*X(P1))/R)/3.                         PF1 537
      YC=(Y(P2)+Y(P4)+(A1*Y(P3)+A2*Y(P1))/R)/3.                         PF1 538
      ZC=(Z(P2)+Z(P4)+(A1*Z(P3)+A2*Z(P1))/R)/3.                         PF1 539
   79 X4=YN*Z1-Y1*ZN                                                    PF1 540
      Y4=ZN*X1-Z1*XN                                                    PF1 541
```

```
      Z4=XN*Y1-X1*YN
      A=1./SQ2(X4,Y4,Z4)
      X4=X4*A
      Y4=Y4*A
      Z4=Z4*A
      X3=ZN*Y4-Z4*YN
      Y3=XN*Z4-X4*ZN
      Z3=YN*X4-Y4*XN
C***  CCMPUTE POINTS IN QUAD SYSTEM (I.E.. ELEMENT COORDINATE SYSTEM)
      DO 80 I=1,9
      L=IP(I)
      XP(I)=X3*(X(L)-XC)+Y3*(Y(L)-YC)+Z3*(Z(L)-ZC)
      YP(I)=X4*(X(L)-XC)+Y4*(Y(L)-YC)+Z4*(Z(L)-ZC)
   80 ZP(I)=XN*(X(L)-XC)+YN*(Y(L)-YC)+ZN*(Z(L)-ZC)
C***  COMPUTE MATRIX COEFFICIENTS TO FIND LOCAL BCDY SURFACE EQUATION
      DO 81 I=2,9
      C(I,1)=1.
      C(I,2)=XP(I)
      C(I,3)=YP(I)
      C(I,4)=XP(I)**2
      C(I,5)=YP(I)**2
      C(I,6)=YP(I)*XP(I)
   81 D(I)=ZP(I)
      DO 82 I=1,6
      C(1,I)=C(9,I)
      C(5,I)=C(5,I)+C(6,I)
   82 C(6,I)=C(7,I)+C(8,I)
      D(1)=D(9)
      D(5)=D(5)+D(6)
      D(6)=D(7)+D(8)
C***  SCLVE MATRIX EQ.  C*CZ=D  FOR CZ
      CALL MATINS(C,9,6,FF,6,1,DETERM,IDM,INDEX)
      IF (IDM.EQ.1) GO TO (84,85),IT
      WRITE (JWRITE,83)
   83 FORMAT (33H ERROR IN INPUT - SINGULAR MATRIX)
      LINE=LINE+1
      ISP=1
      GO TO 85
   84 IT=2
C***  CCMPUTE NEW NORMAL VECTORS
      XN=XN-CZ(2)*X3-CZ(3)*X4
      YN=YN-CZ(2)*Y3-CZ(3)*Y4
```

```
      ZN=ZN-CZ(2)*Z3-CZ(3)*Z4
      A=1./SQ2(XN,YN,ZN)
      XN=XN*A
      YN=YN*A
      ZN=ZN*A
      GO TO 79
C***  STORE DATA IN ARRAY TO BE WRITTEN ON TAPE
   85 B(J+1)=XP(1)
      B(J+2)=YP(1)
      B(J+3)=XP(2)
      B(J+4)=YP(2)
      B(J+5)=XP(3)
      B(J+6)=XP(4)
      B(J+7)=YP(4)
      B(J+8)=X3
      B(J+9)=Y3
      B(J+10)=Z3
      B(J+11)=X4
      B(J+12)=Y4
      B(J+13)=Z4
      XCP(K+1)=XC
      YCP(K+1)=YC
      ZCP(K+1)=ZC
      XNP(K+1)=XN
      YNP(K+1)=YN
      ZNP(K+1)=ZN
      AQP(K+1)=AQ
C***  CCMPUTE QUADRUPOLE MCMENTS
      XI1=XP(1)+XP(2)
      XI2=XP(1)+XP(4)
      XI3=XP(3)+XP(2)
      XI4=XP(3)+XP(4)
      XI5=XP(2)+XP(4)
      YI1=YP(1)+YP(2)
      YI2=YP(1)+YP(4)
      YI3=YP(3)+YP(2)
      YI4=YP(3)+YP(4)
      YI5=YP(2)+YP(4)
      R1=A1/24.
      R2=A2/24.
      R3=AQ/12.
      AXX=(XI1**2+XI2**2)*R1+(XI3**2+XI4**2)*R2+XI5**2*R3
```

```
      AXY=(XI1*YI1+XI2*YI2)*R1+(XI3*YI3+XI4*YI4)*R2+XI5*YI5*R3
      AYY=(YI1**2+YI2**2)*R1+(YI3**2+YI4**2)*R2+YI5**2*R3
C***  COMPUTE BODY SOLID ANGLE
      X1=XC*X3+YC*Y3+ZC*Z3
      Y1=XC*X4+YC*Y4+ZC*Z4
      Z1=XC*XN+YC*YN+ZC*ZN
      RD=1./SQ2(X1,Y1,Z1)
      RCU=RD**3
      RSV=RCU**2*RD
      SA=SA+Z1*(AQ*RCU-((AXX*(Y1**2+Z1**2-4.*X1**2)+AYY*(X1**2+Z1**2-4.*
     1Y1**2))*1.5-15.*X1*Y1*AXY)*RSV)
      B(J+14)=AXX
      B(J+15)=AXY
      B(J+16)=AYY
C***  BODY ERROR TESTS
      D1=SQ2((XP(3)-XP(1)),(YP(3)-YP(1)),0.)
      D2=SQ2((XP(4)-XP(2)),(YP(4)-YP(2)),0.)
      FL=.5*AMAX1(D1,D2)
      CZ23=ABS(CZ(2))+ABS(CZ(3))
      IF (ABS(CZ(2))+ABS(CZ(3)).GT.FL*.001) GO TO 86
      IF (ABS(CZ(1)).LT.FL*.3) GO TO 88
   86 IF (IWRITE.GT.0) GO TO 88
      WRITE (JWRITE,87) CZ23
   87 FORMAT (29H QUESTIONABLE POINT -POOR FIT,6E14.3)
      LINE=LINE+1
   88 IF (XP(4).LT.XP(1)) GO TO 89
      IF ((YP(4)-YP(3))*(YP(1)-YP(2)).GE.0.) GO TO 92
   89 IF (IWRITE.GT.0) GO TO 91
      WRITE (JWRITE,90) (XP(I),YP(I),I=1,4)
   90 FORMAT (30H ERROR IN INPUT - CROSSED QUAD,4(2F10.5,3X))
      LINE=LINE+1
   91 ISP=1
   92 CRCF=SQ2((XP(2)-XP(1)),(YP(2)-YP(1)),0.)+XP(3)-XP(2)+SQ2((XP(1)-XP
     1(4)),(YP(1)-YP(4)),0.)+SQ2((XP(4)-XP(3)),(YP(4)-YP(3)),0.)
      IF (36.*AQ.GT.CRCF**2) GO TO 94
      IF (IWRITE.GT.0) GO TO 104
      LINE=LINE+1
      WRITE (JWRITE,93)
   93 FORMAT (24H WARNING LONG THIN QUAD.)
   94 IF (IWRITE.GT.0) GO TO 104
      IF (Z1.GE.0.) GO TO 96
      WRITE (JWRITE,95)
```

```
   95 FORMAT (35H QUESTIONABLE POINT - INWARD NORMAL)
      LINE=LINE+1
C***  EDIT QUADRILATERAL INFORMATION
   96 GO TO (97,98,99,100),NQ
   97 WRITE (JWRITE,101) M,X(P1),X(P2),X(P3),X(P4),XC,XN,AQ,CZ(4),N,Y(P1
     1),Y(P2),Y(P3),Y(P4),YC,YN,FL,CZ(5),P,Z(P1),Z(P2),Z(P3),Z(P4),ZC,ZN
     1,CZ(1),CZ(6)
      GO TO 102
   98 WRITE (JWRITE,101) M,X(P2),X(P3),X(P4),X(P1),XC,XN,AQ,CZ(4),N,Y(P2
     1),Y(P3),Y(P4),Y(P1),YC,YN,FL,CZ(5),P,Z(P2),Z(P3),Z(P4),Z(P1),ZC,ZN
     1,CZ(1),CZ(6)
      GO TO 102
   99 WRITE (JWRITE,101) M,X(P4),X(P1),X(P2),X(P3),XC,XN,AQ,CZ(4),N,Y(P4
     1),Y(P1),Y(P2),Y(P3),YC,YN,FL,CZ(5),P,Z(P4),Z(P1),Z(P2),Z(P3),ZC,ZN
     1,CZ(1),CZ(6)
      GO TO 102
  100 WRITE (JWRITE,101) M,X(P3),X(P4),X(P1),X(P2),XC,XN,AQ,CZ(4),N,Y(P3
     1),Y(P4),Y(P1),Y(P2),YC,YN,FL,CZ(5),P,Z(P3),Z(P4),Z(P1),Z(P2),ZC,ZN
     1,CZ(1),CZ(6)
  101 FORMAT (1H ,I3,8E14.5/1X,I3,8E14.5/1X,I3,8E14.5/)
  102 LINE=LINE+4
      IF (LINE.LT.MXLINE) GO TO 104
      WRITE (JWRITE,103)
  103 FORMAT (4H1  M,7X,2HX1,12X,2HX2,12X,2HX3,12X,2HX4,12X,2HXC,12X,2HX
     1N,12X,1HA,13X,3HCZ4/4H   N,7X,2HY1,12X,2HY2,12X,2HY3,12X,2HY4,12X,
     12HYC,12X,2HYN,12X,2HFL,12X,3HCZ5/4H   P,7X,2HZ1,12X,2HZ2,12X,2HZ3,
     112X,2HZ4,12X,2HZC,12X,2HZN,12X,4HCZ1 ,10X,3HCZ6/)
      LINE=0
  104 J=J+16
      IF (KNEW.NE.1.AND.P.LE.NQES) GO TO 105
C***  INITIALIZE ARRAYS
      VNP(P)=0.0
      IFN(P)=0
  105 I=P
      P=P+1
      NQ=NQ+1
      K=K+1
C***  WRITE B ARRAY CONTAINING INFORMATION FOR 12 QUADRILATERALS
      IF (J.LT.192) GO TO 108
      IF (KNEW.EQ.1)KWRITE=KWRITE+1
      WRITE (KFILE)Q,(E(I),I=1,192)
      IF (KNEW.NE.1) GO TO 107
```

```
      IF (KWRITE.NE.IDO+1) GO TO 107
      QHCLD=Q
      DC 106 I=1,192
  106 BHOLD(I)=B(I)
  107 Q=P
      J=0
  108 CCNTINUE
  109 M1=M1+2
  110 N1=N1+2
      IF (KNEW.NE.1) GC TO 129
  111 IMESS=0
C***  READ AND TEST M AND N STATIONS TO SPECIFY NCNZERO NORMAL VELOCITY
C***  ON A QUADRILATERAL
  112 READ (JREAD,113, MS1,MS2,NS1,NS2,IPNS
  113 FCRMAT (5I5)
      IF (MS1.FQ.0.AND.IPNS.EQ.0) GO TO 129
      IF (IPNS.EQ.0) GO TO 112
      IF (((MS1.LF.0.OR.MS1.GT.MMAX).OR.(MS2.LE.0.OR.MS2.GT.MMAX)).OR.((
     1NS1.LE.0.OR.NS1.GT.NMAX).OR.(NS2.LE.0.OR.NS2.GT.NMAX)).OR.(MS2-MS1
     1).NE.1.OR.(NS2-NS1).NE.1) GO TO 116
      IF (IMESS.FQ.1) GO TO 115
      WRITE (JWRITE,114)
  114 FORMAT (1H1,///,5X,48HDFSIGNATION OF NORMAL VELOCITY ON QUADPILATE
     1RALS,/)
      LINE=6
  115 IMFSS=1
      GC TO 118
  116 WRITE (JWRITF,117) MS1,MS2,MMAX,NS1,NS2,NMAX
  117 FORMAT (1X,/,10X,36HAN ERROR IN DATA INPUT HAS OCCURRED:,/,15X,4HM
     1S1=,I3,3X,4HMS2=,I3,3X,5HMMAX=,I3,/,15X,4HNS1=,I3,3X,4HNS2=,I3,3X,
     15HNMAX=,I3,/,10X,12HCALLING EXIT,//)
      GC TO 172
  118 MC=0
      ICK=1
      LC=0
C***  LOCATE CORNER PCINTS OF NONZERC-NORMAL-VELOCITY QUADRILATERAL
      DC 120 NX=NMIN,NS2
      DC 120 MX=MMIN,MMAX
      LC=LC+1
      IF ((NX.EQ.NS1.OR.NX.FQ.NS2).AND.(MX.EQ.MS1.OR.MX.EQ.MS2)) GO TO 1
     119
      GO TO 120
```

```
  119 MC=MC+1
      IPT(MC)=LC
      IF (NX.EQ.NS2.AND.MX.EQ.MS2) GO TO 121
  120 CCNTINUE
C***  CORNER POINTS LOCATED. NOW FIND CORRESPONDING PANEL NUMBER
  121 DC 124 JX=1,NQE
      NPL=JX
      LC=0
      DC 123 MX=1,4
      DO 122 NX=1,4
      IF (IPT(MX).EQ.IDD(JX,NX))LC=LC+1
  122 CONTINUE
  123 CONTINUE
      IF (LC.EQ.4) GC TO 125
  124 CONTINUE
C***  PANEL NUMBER FOUND. STORE INDICATION OF NONZERO NORMAL VELOCITY
  125 IPN(NPL)=IFNS
      IF (IPNS.LT.0) GO TO 127
      WRITE (JWRITE,126) MS1,MS2,NS1,NS2,NPL
  126 FORMAT (2X,4(I5,1X),2X,6HPANEL ,I4,2X,75HHAS BEEN DESIGNATED TO HA
     1VE A NONZERO OUTWARD NORMAL COMPONENT OF VELCCITY.)
      LINE=LINE+1
      GO TO 112
  127 WRITE (JWRITE,128) MS1,MS2,NS1,NS2,NPL
  128 FORMAT (2X,4(I5,1X),2X,6HPANEL ,I4,2X,74HHAS BEEN DESIGNATED TO HA
     1VE A NCNZERO INWARD NORMAL COMPONENT OF VELCCITY.)
      LINE=LINE+1
      GO TO 112
C***  SET UP FOR NEXT SECTION
  129 IF (ITFST.NE.0) GC TO 150
      DC 130 M=MMIN,MMAX
      DO 130 N=NMIN,NMAX
  130 ID(M,N)=0
  131 NSS=NS
      PC=1
      MMAX=MI
      MMIN=MI
      NMIN=NI
      NMAX=NI
      NE=ME
      IF (NS.GT.0) GO TO 31
      JHOLD=((NMAXQD-3)*(MMAXQD-1)-12*IDD)*16
```

```
      IF (J.EQ.192) GO TO 133
      IF (KNEW.EQ.1)KWRITE=KWRITE+1
      WRITE (KFILF)Q,(B(I),I=1,192)
      IF (KNEW.NE.1) GO TO 133
      IF (KWRITE.NE.IDO+1) GO TO 133
      QHOLD=Q
      DO 132 I=1,192
  132 BHCLD(I)=B(I)
  133 NP=K
      NCUAD=NP
      ISMP=ISM+1
      GO TO (137,136,135,134),ISMP
  134 SA=SA+SA
  135 SA=SA+SA
  136 SA=SA+SA
  137 J=1
C***  CHECK SOLID ANGLE
      IF (ABS(SA-12.566).LT..05) GO TO 138
      IF (ABS(SA).LF..05) GO TO 140
  138 WRITE (JWRITE,139) SA
  139 FORMAT (40HOPROBABLE ERROR IN INPUT - SOLIC ANGLE =,2F12.3)
  140 REWIND KFILE
C***  CHECK NO. OF QUADRILATERALS
      IF (NP.FQ.NQF) GO TO 143
      WRITE (JWRITE,141) NP,NQE
  141 FCRMAT (1X,I4,27H QUADRILATERALS GIVEN, NOT ,I4)
      WRITE (JWRITE,142)
  142 FCRMAT (//,1X,11HEXIT CALLED,//)
      GO TO 172
  143 IF (ISP.LE.0) GO TO 144
      WRITE (JWRITE,142)
      GO TO 172
C***  CALL REMAINING SECTIONS OF THE POTENTIAL FLOW PROGRAM
  144 CALL PFP2(ISM,K)
      IF (ICANCL.NE.0) GO TO 172
      NSRV=INSRV
  145 IF (NSRV.LF.0) GO TO 158
      LL1=1
      LL2=NP
      IF (KNEW.NE.2) GO TO 146
      LL1=LL1S
      LL2=LL2S
```

```
146 IF (KNEW.EQ.1) WRITE (JWRITE,147)
147 FORMAT (1H1,//,1X,67HINDUCED VELOCITY BY RING VORTICES AT ORIGINAL
   1-BODY PANEL CENTROIDS:,/)
    IF (KNEW.EQ.2) WRITE (JWRITE,148)
148 FORMAT (1H1,//,1X,63HINDUCED VELOCITY BY RING VORTICES AT WAKE-BOD
   1Y PANEL CENTROIDS:,/)
    WRITE (JWRITE,149)
149 FORMAT (2X,5HPANEL,3X,10('-'),1X,23HPANEL CENTROID LOCATION,1X,11(
   1'-'),4X,12('-'),1X,19HVELOCITY COMPONENTS,1X,13('-'),/,1X,6HNUMBER
   1,9X,1HX,15X,1HY,15X,1HZ,17X,1HX,15X,1HY,15X,1HZ,/)
    LINE=0
    GO TO 153
150 WRITE (JWRITE,151)
151 FORMAT (1H1,//,1X,63HINDUCED VELOCITY BY RING VORTICES ALONG BODY-
   1CENTERLINE POINTS:,/)
    WRITE (JWRITE,152)
152 FORMAT (1X,5HPOINT,3X,15('-'),1X,14HPOINT LOCATION,1X,15('-'),4X,1
   12('-'),1X,19HVELOCITY COMPONENTS,1X,13('-'),/,1X,6HNUMBER,9X,1HX,1
   15X,1HY,15X,1HZ,17X,1HX,15X,1HY,15X,1HZ,/)
153 DO 157 LL=LL1,LL2
    IF (ITEST.EQ.0) GO TO 154
C***  FOR ITEST=1, SET AXIAL STATIONS FOR VORTEX INDUCED VELOCITIES
    YCP(LL)=0.0
    ZCP(LL)=0.0
    IF (LL.EQ.1)XCP(LL)=XHLD
    IF (LL.GT.1)XCP(LL)=XCP(LL-1)-0.5E0
    XORD(5,LL)=XCP(LL)
C***  COMPUTE INDUCED VELOCITY COMPONENTS AT PANEL CENTROIDS BY THE RING
C***  VORTICIES
154 CALL RGVRTX(XCV,YCV,ZCV,XCP(LL),YCP(LL),ZCP(LL),RDM,NCA,CA,PHI,NVP
   1HI,NRS,NSRV,VX,VY,VZ,IM,NPTS)
    VTX(1,LL)=VX
    VTX(2,LL)=VY
    VTX(3,LL)=VZ
    IF (ITEST.NE.0)YCRD(5,LL)=VTX(1,LL)
    IF ((LINE+1).LE.MXLINE) GO TO 155
    WRITE (JWRITE,13)
    IF (ITEST.EQ.0) WRITE (JWRITE,149)
    IF (ITEST.NE.0) WRITE (JWRITE,152)
    LINE=0
155 WRITE (JWRITE,156) LL,XCP(LL),YCP(LL),ZCP(LL),(VTX(IS,LL),IS=1,3)
156 FORMAT (2X,I4,3X,1H(,1PE14.7,2H, ,1PE14.7,2H, ,1PE14.7,4H)    ,1PE1
```

```
     14.7,2X,1PE14.7,2X,1PE14.7)
      LINE=LINE+1
  157 CONTINUE
      IF (ITEST.EQ.0) GO TO 158
C***  WARNING:  THE FOLLOWING PLOT CARDS MAY BE INSTALLATION-DEPENDENT.
      KIND(5)=-1
      IPLOT=1
      CALL PICSIZ(10.0,10.0)
      CALL GRAFF(7.5,XKEEP1,XKEEP2,XINC,0,'REFERENCE COORDINATE SYSTEM X
     1-VALUE_',XORD,XDUM,5.0,-1.0,2.0,1.0,0,'NORMALIZED INTERPRETATION S
     1CALE_',YORD,YDUM,1,5,5,200,NSRV,2,NMAX,NSRV,LL2,KIND,1,' _')
  158 IF (ITEST.NE.0) GO TO 172
      CALL PFP3(EPS,MIX)
      CALL PFP4(CLPA,CDPA,ICK)
C***  COMPUTE INTERIOR PRESSURE COEFFICIENT AND NORMAL VELOCITIES
      IF (ICK.EQ.1) CALL VINOUT(JWRITE,VINF,ROE,NQE,VI)
      IF (ICANCL.NE.0) GO TO 172
      NSRV=0
      ICK=ICK+1
      IF (ICK.EQ.2) GO TO 145
      IF (KNEW.NE.2) GO TO 160
C***  ON SECOND PASS THROUGH COMPUTE LIFT & DRAG COEFFICIENTS
      CL=CLPA/REFA
      CD=(CDPA+CDFA)/REFA
      WRITE (JWRITE,159) CL,CD,REFA,REBODY,STOTAL
  159 FORMAT (////,31X,23HTOTAL BODY COEFFICIENTS///,28X,29(1H*)/30X,16H
     1TOTAL BODY CL = ,F11.5,/30X,16HTOTAL BODY CD = ,F11.5,/29X,17HREFE
     1RENCE AREA = ,F11.5,/28X,18HREYNOLDS NUMBER = ,E11.4,/32X,14HBODY
     1LENGTH = ,F11.5,/28X,29(1H*),//)
      GO TO 172
  160 CALL PFP5(MMAXQD,NMAXQD,AREAAV,AVDELS,AVXCG,CDFA)
      IF (ICANCL.NE.0) GO TO 172
C***  REMAINING CARDS COMPUTE NECESSARY INFORMATION FOR WAKE-BODY
      KNEW=2
      NQFS=NQF
      MI=1
      NI=1
      NS=1000
      K=(NMAXQD-3)*(MMAXQD-1)
      MF=0
      NQE=NQE+2*(MMAXQD-1)
      LL1S=K
```

```
      LL2S=NQE
      REWIND KFILE1
      DC 161 I=1,IDO
      READ (KFILE1)Q,(B(IJK),IJK=1,192)
      WRITE (KFILE2)Q,(B(IJK),IJK=1,192)
  161 CONTINUE
      KFILE=KFILE2
      J=JHOLD
      IF (J.EQ.0) GO TO 163
      DO 162 I=1,J
  162 B(I)=BHOLD(I)
  163 Q=QHOLD
      JP=JHOLD/16
      QADD=QHOLD+.5
      P=QADD+JP
      ISTART=(NMAXQD-3)*MMAXQD
      IST=ISTART+MMAXQD
      XHCLD1=X(ISTART+1)
      XHOLD2=X(IST+1)
C***  Y REFERENCE LINE MUST BE 0.0
      ZAV=0.0
      DC 164 I=1,MMAXQD
      XNEW(I)=X(ISTART+I)
      YNEW(I)=Y(ISTART+I)
      ZNEW(I)=Z(ISTART+I)
  164 ZAV=ZAV+ZNEW(I)
      THETA=-90.0*3.14159/180.0
      ZAV=ZAV/MMAXQD
      RAV=0.0
      DC 165 I=1,MMAXQD
      RNFW=SQRT((YNEW(I))**2+(ZNEW(I)-ZAV)**2)
  165 PAV=RAV+RNEW
      RAV=RAV/MMAXQD
      FAC=(AVXCG-XHOLJ1)/(XHOLD2-XHOLD1)
      AVSLOP=C.0
      DO 166 I=1,MMAXJD
      YCG=YNEW(I)-(YNEW(I)-Y(IST+I))*FAC
      ZCG=ZNEW(I)-(ZNEW(I)-Z(IST+I))*FAC
      P1=SQRT((YNFW(I))**2+(ZNFW(I)-ZAV)**2)
      R2=SQRT((YCG)**2+(ZCG-ZAV)**2)+2.0*AVDELS
      SLOPF=(R2-R1)/(XHOLD1-AVXCG)
  166 AVSLOP=AVSLOP+SLOPE
```

```
      AVSLOP=AVSLOP/MMAXQD
      AREAT=4.0*AREAAV*2.0*MMAXQD
      XINF=XHOLD1-AREAT/(3.14159*RAV)
C***  CHECK TO MAKE SURE BODY SLOPE IS DECREASING
      IF (AVSLOP.GE.0.)AVSLOP=-ABS(RAV/(XINF-XHOLD1))
      WRITE (JWRITE,167) AVSLOP,AREAAV,AVXCG,XINF
  167 FORMAT (1H1/50X,24HBEGIN WAKE BODY GEOMETRY///,17H AVERAGE SLOPE =
     1 ,F8.5,3X,22H AVERAGE PANEL AREA = ,F8.5,3X,22H AVERAGE X-CENTROID
     1 = ,F10.5,3X,20H END OF BODY AT X = ,F10.5)
      DELTAZ=.01
      KK=1
      XX1=XHOLD1
      ZZ2=0.0
      RR1=RAV
  168 ZZ2=ZZ2+DELTAZ
      XX2=ZZ2*(XINF-XHOLD1)+XHOLD1
      RR2=RAV*EXP(AVSLOP*ZZ2)*(1.0-ZZ2)
      DTHETA=3.14159/(MMAXQD-1)
      AREA2=(RR2+RR1)*SIN(DTHETA/2.0)*SQRT((RR2-RR1)**2+(XX2-XX1)**2)
      IF (AREA2.LT.AREAAV) GO TO 168
      XK(KK)=XX2
      RK(KK)=RR2
      XX1=XX2
      RR1=RR2
      KK=KK+1
      IF (KK.GE.4) GO TO 169
      GO TO 168
  169 XK(4)=XINF
      RK(4)=0.0
      X1=XK(1)
      X2=XK(2)
      X3=XK(3)
      X4=XK(4)
      R1=RK(1)
      R2=RK(2)
      R3=RK(3)
      R4=RK(4)
      NSTRT1=MMAXQD
      NSTRT2=2*MMAXQD
      NSTRT3=3*MMAXQD
      NSTRT4=4*MMAXQD
      YFAC=(-Y(1))/(XINF-XHOLD1)
```

```
      ZFAC=(ZAV-Z(1))/(XINF-XHOLD1)
      DO 170 I=1,MMAXQD
      CCSTHT=COS(THETA)
      SINTHT=SIN(THETA)
      XNEW(NSTRT1+I)=X1
      XNEW(NSTRT2+I)=X2
      XNEW(NSTRT3+I)=X3
      XNEW(NSTRT4+I)=X4
      YNEW(NSTRT1+I)=R1*COSTHT-(XNEW(NSTRT1+I)-XHOLD1)*YFAC
      YNEW(NSTRT2+I)=R2*COSTHT-(XNEW(NSTRT2+I)-XHOLD1)*YFAC
      YNEW(NSTRT3+I)=R3*COSTHT-(XNEW(NSTRT3+I)-XHOLD1)*YFAC
      YNEW(NSTRT4+I)=R4*COSTHT-(XNEW(NSTRT4+I)-XHOLD1)*YFAC
      ZNEW(NSTRT1+I)=R1*SINTHT+ZAV-(XNEW(NSTRT1+I)-XHOLD1)*ZFAC
      ZNEW(NSTRT2+I)=R2*SINTHT+ZAV-(XNEW(NSTRT2+I)-XHOLD1)*ZFAC
      ZNEW(NSTRT3+I)=R3*SINTHT+ZAV-(XNEW(NSTRT3+I)-XHOLD1)*ZFAC
      ZNEW(NSTRT4+I)=R4*SINTHT+ZAV-(XNEW(NSTRT4+I)-XHOLD1)*ZFAC
  170 THETA=THETA+DTHETA
C***  STCRE COORDINATES FOR PUNCHING
      KPI=5*MMAXQD
      DO 171 I=1,KPI
      XS(ISTART+I)=XNEW(I)
      YS(ISTART+I)=YNEW(I)
  171 ZS(ISTART+I)=ZNEW(I)
      XI=XNEW(1)
      YI=YNEW(1)
      ZI=ZNEW(1)
      GO TO 131
C***  PREPARE FOR NEXT DATA SET
  172 IF (IDS.NE.JREAD) REWIND IDS
      GO TO 1
C***  WARNING:  THE FOLLOWING PLOT CARD MAY BE INSTALLATION-DEPENDENT.
  173 IF (IPLOT.NE.0) CALL PICSIZ(0.0,0.0)
      STOP
      END


      FUNCTION SQ2(X,Y,Z)
C
C***  FUNCTION SQ2 CALCULATES THE SQUARE ROOT OF R**2
C
      RS=X**2+Y**2+Z**2
```

```
      SQ2=SQRT(RS)
      RFTURN
      END

      SUBROUTINE MATINS(A,NR,N1,B,NC,M1,DETERM,ID,INDEX)
C
C***  MATRIX INVERSION(PIVOT METHOD) WITH SIMULTANEOUS EQUATION SOLUTION
C***  INPUT A(NR,NR),B(NR,NC),& INDEX(NR,3)--NR,NC ARE ARRAY DIMENSIONS.
C***  DEFINITIONS: N1-THE ORDER OF A. M1-THE NUMBER OF COLUMN VECTORS IN
C***  B(MAY BE 0). DETERM-CONTAINS DETERMINANT ON EXIT. INDEX-WORK ARRAY
C***  ID-(ID=2--SINGULAR MATRIX,ID=1--SUCESSFUL INVERSION). A-THE INPUT
C***  MATRIX WHICH IS REPLACED BY A INVERSE. B-COLUMN VECTORS WHICH IS
C***  REPLACED BY CORRESPONDING SOLUTION VECTORS.
C
      DIMENSION A(9,6),B(9,1),INDEX(9,3)
      FQUIVALFNCE (IROW,JROW),(ICOLUM,JCOLUM),(AMAX,T,SWAP)
C
C***  INITIALIZATION
      N=N1
      M=M1
      DETFRM=1.0
      DO 1 J=1,N
    1 INDEX(J,3)=0
      DO 20 I=1,N
C***  SEARCH FOR PIVOT ELEMENT
      AMAX=0.0
      DC 6 J=1,N
      IF (INDEX(J,3)-1)  2,6,2
    2 DC 5 K=1,N
      IF (INDFX(K,3)-1)  3,5,26
    3 IF (AMAX-ABS(A(J,K)))  4,5,5
    4 IRCW=J
      ICOLUM=K
      AMAX=ABS(A(J,K))
    5 CCNTINUF
    6 CCNTINUE
      INDFX(ICOLUM,3)=INDEX(ICOLUM,3)+1
      INDEX(I,1)=IROW
      INDFX(I,2)=ICOLUM
C***  INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGCNAL
```

```
      IF (IROW-ICOLUM)  7,11,7
    7 DETERM=-DETERM
      DO 8 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
    8 A(ICOLUM,L)=SWAP
      IF (M)  11,11,9
    9 DO 10 L=1,M
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLUM,L)
   10 B(ICOLUM,L)=SWAP
C***  DIVIDE PIVCT ROW BY PIVOT ELEMENT
   11 PIVOT=A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT
      A(ICOLUM,ICOLUM)=1.0
      DO 12 L=1,N
   12 A(ICOLUM,L)=A(ICCLUM,L)/PIVOT
      IF (M)  15,15,13
   13 DO 14 L=1,M
   14 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C***  REDUCE NON-PIVOT ROWS
   15 DO 20 L1=1,N
      IF (L1-ICOLUM)  16,20,16
   16 T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 17 L=1,N
   17 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      IF (M)  20,20,18
   18 DO 19 L=1,M
   19 B(L1,L)=B(L1,L)-B(ICCLUM,L)*T
   20 CONTINUE
C***  INTERCHANGE CCLUMNS
      DO 23 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2))  21,23,21
   21 JROW=INDEX(L,1)
      JCCLUM=INDEX(L,2)
      DO 22 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCCLUM)
      A(K,JCOLUM)=SWAP
   22 CONTINUE
```

```
23 CCNTINUE
   DO 24 K=1,N
   IF (INDEX(K,3)-1)  26,24,26
24 CONTINUF
   ID=1
25 RETURN
26 ID=2
   GC TO 25
   END

      SUBROUTINF PFP2(ISM,KM)
C
C***  SUBROUTINE PFP2 CALCULATES THE INDUCED VELOCITIES OF EACH QUAD-
C***  RILATERAL ON EVERY OTHER QUADRILATERAL
C
      DIMENSION B(193),V1(1000),C1(900),VX(8),VY(8),VZ(8)
      EQUIVALENCE (Y3,Y2)
      CCMMON TITLF(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP(
     1650),AQP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW
      CCMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE
     15
      COMMON /S1K/SN(650),VNP(650),IPN(650)
      COMMCN /CANCEL/ICANCL,NODE
C
      BLK=1.0
      IDW=0
C***  READ FIRST BLOCK OF B ARRAY
      READ (KFILE)(B(I),I=1,193)
      K=1
      J=1
      P=1
      JC=1
      JV=2
      KMM=5*((NP+4)/5)+1
C***  START LCOP OVER QUADRILATERALS
      IF (B(1)-P)  68,3,68
    3 J=2
C***  CHOOSE QUADRILATERAL AS REFERENCE POINT
    4 X1=B(J)
      Y1=B(J+1)
```

```
      X2=B(J+2)
      Y2=B(J+3)
      X3=B(J+4)
      X4=B(J+5)
      Y4=B(J+6)
      XN=XNP(K)
      YN=YNP(K)
      ZN=ZNP(K)
      XC=XCP(K)
      YC=YCP(K)
      ZC=ZCP(K)
      A=AQP(K)
      XX=B(J+7)
      YX=B(J+8)
      ZX=B(J+9)
      XY=B(J+10)
      YY=B(J+11)
      ZY=B(J+12)
C***  CCMPUTE LENGTH OF SIDES OF QUADRILATERAL
      D12=SQ2F(X1,X2,Y1,Y2,0.,0.)
      D23=SQ2F(X2,X3,Y2,Y3,.0,.0)
      D34=SQ2F(X3,X4,Y3,Y4,.0,.0)
      D41=SQ2F(X4,X1,Y4,Y1,.0,.0)
C***  CCMPUTE SLOPE OF SIDES OF QUADRILATERAL
      IF (X2-X3)  6,5,6
    5 CI23=1.
      GC TO 7
    6 CM23=(Y2-Y3)/(X2-X3)
      CI23=0.
    7 IF (X3-X4)  9,8,9
    8 CI34=1.
      GC TO 10
    9 CM34=(Y4-Y3)/(X4-X3)
      CI34=0.
   10 IF (X4-X1)  12,11,12
   11 CI41=1.
      GC TO 13
   12 CM41=(Y1-Y4)/(X1-X4)
      CI41=0.
   13 IF (X1-X2)  15,14,15
   14 CI12=1.
      GC TO 16
```

```
   15 CM12=(Y2-Y1)/(X2-X1)
      CI12=0.
C***  DEFINE QUACRAPOLE MOMENTS
   16 CIXX=B(J+13)
      CIXY=B(J+14)
      CIYY=B(J+15)
C***  INITIALIZE VARIABLES FOR SINE AND COSINE
      CY12=0.0
      CX12=0.0
      CY23=0.0
      CX23=0.0
      CY34=0.0
      CX34=0.0
      CY41=0.0
      CX41=0.0
C***  CCMPUTE SINE AND COSINE OF SLOPE ANGLE FOR EACH SIDE
      IF (D12) 17,18,17
   17 CY12=(Y2-Y1)/D12
      CX12=(X1-X2)/D12
   18 IF (D23) 19,20,19
   19 CY23=(Y3-Y2)/D23
      CX23=(X2-X3)/D23
   20 IF (D34) 21,22,21
   21 CY34=(Y4-Y3)/D34
      CX34=(X3-X4)/D34
   22 IF (D41) 23,24,23
   23 CY41=(Y1-Y4)/D41
      CX41=(X4-X1)/D41
C***  CCMPUTE MAXIMUM LENGTH OF QUADRILATERAL
   24 ST=ABS(X1-X3)
      ST2=SQ2F(X2,X4,Y2,Y4,.0,.0)
      ST=AMAX1(ST,ST2,D12,D23,D34,D41)
C***  START LOOP OVER CENTROID POINTS OF ALL QUADRILATERALS
      KC=1
      L=1
   25 I=KQ
      IS=1
      XCQ=XCP(I)
      YCQ=YCP(I)
      ZCQ=ZCP(I)
      XNQ=XNP(I)
      YNG=YNP(I)
```

```
      ZNQ=ZNP(I)                                                        PF2 119
C***  COMPUTE DISTANCE BETWEEN CENTROID OF REFERENCE QUADRILATERAL AND  PF2 120
C***  CENTROID OF ANOTHER QUADRILATERAL                                 PF2 121
   26 RPQ=SQ2F(XC,XCQ,YC,YCQ,ZC,ZCQ)                                    PF2 122
      IF (RPQ-ST*4)  27,27,39                                           PF2 123
C***  TRANSFORM POINT FROM REFFERENCE COORDINATE SYSTEM TO ELEMENT(QUAD) PF2 124
C***  COORDINATE SYSTEM                                                 PF2 125
   27 X=(XCQ-XC)*XX+(YCQ-YC)*YX+(ZCQ-ZC)*ZX                             PF2 126
      Y=(XCQ-XC)*XY+(YCQ-YC)*YY+(ZCQ-ZC)*ZY                             PF2 127
      Z=(XCQ-XC)*XN+(YCQ-YC)*YN+(ZCQ-ZC)*ZN                             PF2 128
      IF (RPQ-ST*2.0)  28,28,37                                         PF2 129
C***  COMPUTE VELOCITY COEFFICIENTS BY EXACT METHOD (QUAD SYSTEM)        PF2 130
   28 R1=SQ2F(X,X1,Y,Y1,Z,0.)                                           PF2 131
      R2=SQ2F(X,X2,Y,Y2,Z,0.)                                           PF2 132
      R3=SQ2F(X,X3,Y,Y3,Z,0.)                                           PF2 133
      R4=SQ2F(X,X4,Y,Y4,Z,0.)                                           PF2 134
      IF ((R1+R2).LE.D12) GO TO 75                                      PF2 135
      IF ((R2+R3).LE.D23) GO TO 75                                      PF2 136
      IF ((R3+R4).LE.D34) GO TO 75                                      PF2 137
      IF ((R4+R1).LE.D41) GO TO 75                                      PF2 138
      CLA1=ALOG((R1+R2-D12)/(R1+R2+D12))                                PF2 139
      CLA2=ALOG((R2+R3-D23)/(R2+R3+D23))                                PF2 140
      CLA3=ALOG((R3+R4-D34)/(R3+R4+D34))                                PF2 141
      CLA4=ALOG((R4+R1-D41)/(R4+R1+D41))                                PF2 142
      TVX=CY12*CLA1+CY23*CLA2+CY34*CLA3+CY41*CLA4                       PF2 143
      TVY=CX12*CLA1+CX23*CLA2+CX34*CLA3+CX41*CLA4                       PF2 144
      TVZ=0.                                                            PF2 145
      IF (ABS(Z/ST)-.010)  38,29,29                                     PF2 146
   29 ZSQ=Z**2                                                          PF2 147
      E1=ZSQ+(X-X1)**2                                                  PF2 148
      E2=ZSQ+(X-X2)**2                                                  PF2 149
      F3=ZSQ+(X-X3)**2                                                  PF2 150
      F4=ZSQ+(X-X4)**2                                                  PF2 151
      H1=(Y-Y1)*(X-X1)                                                  PF2 152
      H2=(Y-Y2)*(X-X2)                                                  PF2 153
      H3=(Y-Y3)*(X-X3)                                                  PF2 154
      H4=(Y-Y4)*(X-X4)                                                  PF2 155
      IF (CI12)  30,30,31                                               PF2 156
   30 WS1=(CM12*F1-H1)/(Z*R1)                                           PF2 157
      WS2=(CM12*E2-H2)/(Z*R2)                                           PF2 158
      TVZ=ATAN(WS1)-ATAN(WS2)                                           PF2 159
   31 IF (CI23)  32,32,33                                               PF2 160
```

```
   32 TVZ=TVZ+ATAN((CM23*E2-H2)/(Z*R2))-ATAN((CM23*E3-H3)/(Z*R3))       PF2 161
   33 IF (CI34)  34,34,35                                                PF2 162
   34 TVZ=TVZ+ATAN((CM34*E3-H3)/(Z*R3))-ATAN((CM34*E4-H4)/(Z*R4))       PF2 163
   35 IF (CI41)  36,36,38                                                PF2 164
   36 TVZ=TVZ+ATAN((CM41*F4-H4)/(Z*R4))-ATAN((CM41*E1-H1)/(Z*R1))       PF2 165
      GO TO 38                                                           PF2 166
C***  COMPUTE VELOCITY COEFFICIENTS BY QUADRAPOLE METHOD (QUAD SYSTEM)   PF2 167
   37 RPQ3=RPQ**3                                                        PF2 168
      RPQ7=(RPQ3**2)*RPQ                                                 PF2 169
      WS1=X/RPQ3                                                         PF2 170
      XSQ=X**2                                                           PF2 171
      YSQ=Y**2                                                           PF2 172
      ZSQ=Z**2                                                           PF2 173
      PS=YSQ+ZSQ-4.*XSQ                                                  PF2 174
      QS=XSQ+ZSQ-4.*YSQ                                                  PF2 175
      WS2=X*(9.*PS+30.*XSQ)/RPQ7                                         PF2 176
      WS3=3.*Y*PS/RPQ7                                                   PF2 177
      WS4=3.*X*QS/RPQ7                                                   PF2 178
      TVX=A*WS1-CIXY*WS3-CIXX*WS2-CIYY*WS4                               PF2 179
      WS1=Y/RPQ3                                                         PF2 180
      WS2=Y*(9.*QS+30.*YSQ)/RPQ7                                         PF2 181
      TVY=A*WS1-CIXX*WS3-CIXY*WS4-CIYY*WS2                               PF2 182
      TVZ=Z*(A/RPQ3-3.*(CIXX*PS-5.*CIXY*X*Y+CIYY*QS)/RPQ7)               PF2 183
C***  TRANSFORM FROM ELEMENT TO REFERENCE COORDINATE SYSTEM              PF2 184
   38 VX(IS)=TVX*XX+TVY*XY+TVZ*XN                                        PF2 185
      VY(IS)=TVX*YX+TVY*YY+TVZ*YN                                        PF2 186
      VZ(IS)=TVX*ZX+TVY*ZY+TVZ*ZN                                        PF2 187
      GO TO 40                                                           PF2 188
C***  COMPUTE VELOCITY COEFFICIENTS BY MONOPOLE METHOD (DIRECTLY IN THE  PF2 189
C***  REFERENCE COORDINATE SYSTEM)                                       PF2 190
   39 ARPQ3=A/(RPQ**3)                                                   PF2 191
      VX(IS)=(XCO-XC)*ARPQ3                                              PF2 192
      VY(IS)=(YCO-YC)*ARPQ3                                              PF2 193
      VZ(IS)=(ZCO-ZC)*ARPQ3                                              PF2 194
C***  REFLECT CENTROID POINT IN PLANE OF SYMMETRY                        PF2 195
   40 GO TO (41,43,45,46,48,49,50,51),IS                                 PF2 196
C***  DO LOOPS SET UP TO FORCE USE OF INDEX REGISTERS                    PF2 197
   41 J1=JV                                                              PF2 198
      J2=JC                                                              PF2 199
      V1(J1)=VX(1)                                                       PF2 200
      V1(J1+1)=VY(1)                                                     PF2 201
      V1(J1+2)=VZ(1)                                                     PF2 202
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF (ISM)  54,54,42
   42 IS=2
C***  XZ SYMMETRY
      YCQ=-YCQ
      GO TO 26
   43 IF (ISM-1)  53,53,44
C***  XY SYMMETRY
   44 IS=3
      ZCQ=-ZCQ
      GO TO 26
   45 IS=4
      YCQ=-YCQ
      GO TO 26
   46 IF (ISM-2)  52,52,47
C***  YZ SYMMETRY
   47 IS=5
      XCQ=-XCQ
      GO TO 26
   48 IS=6
      YCQ=-YCQ
      GC TO 26
   49 IS=7
      ZCQ=-ZCQ
      GO TO 26
   50 IS=8
      YCQ=-YCQ
      GC TO 26
C***  ADD CONTRIBUTIONS OF ALL REFLECTIONS TO OBTAIN NET INDUCED
C***  VELOCITY CCMPONENTS IN REFERENCE COORDINATE SYSTEM
   51 V1(J1)=V1(J1)+VX(8)+VX(7)+VX(6)+VX(5)
      V1(J1+1)=V1(J1+1)-VY(8)+VY(7)+VY(6)-VY(5)
      V1(J1+2)=V1(J1+2)-VZ(8)-VZ(7)+VZ(6)+VZ(5)
   52 V1(J1)=V1(J1)+VX(4)+VX(3)
      V1(J1+1)=V1(J1+1)+VY(4)-VY(3)
      V1(J1+2)=V1(J1+2)-VZ(4)-VZ(3)
   53 V1(J1)=V1(J1)+VX(2)
      V1(J1+1)=V1(J1+1)-VY(2)
      V1(J1+2)=V1(J1+2)+VZ(2)
C***  CALCULATE THE NORMAL VELOCITY INDUCED AT THE CONTROL POINT OF THE
C***  I-TH ELEMENT BY A UNIT SOURCE DENSITY ON THE J-TH ELEMENT (STORED
C***  BY COLUMNS)
   54 C1(J2)=XNQ*V1(J1)+YNQ*V1(J1+1)+ZNQ*V1(J1+2)
```

```
      JV=JV+3
   55 JC=JC+1
C***  WRITE COEFFICIENTS ON TAPE
      IF (JV-1001)  61,56,56
   56 JV=2
      V1(1)=BLK
      IF (BLK-636.0)  57,58,59
   57 WRITE (KFILE3)BLK,V1
      GO TO 60
   58 REWIND KFILE3
   59 WRITF (KFILF5)BLK,V1
   60 BLK=BLK+1.
   61 IF (JC-901)  63,62,62
   62 IDW=IDW+1
      WRITE (KFILE4)IDW,C1
      JC=1
   63 KQ=KQ+1
      L=L+1
C***  END OF LOOP OVER CENTROIDS
      IF (KQ-KM)  25,25,64
   64 C1(JC)=0
      IF (KQ-KMM)  55,65,65
   65 P=P+1
      K=K+1
      J=J+16
      IF (K-KM)  66,66,70
C***  FND OF LOOP OVER QUADRILATERALS
C***  READ NEXT BLOCK OF B ARRAY IF NEEDED
   66 IF (J-193)  4,67,67
   67 READ (KFILE)(B(I),I=1,193)
      J=2
      IF (B(1)-P)  68,4,68
   68 WRITE (JWRITE,69) B(1),P
   69 FORMAT (28H0 FOINTS OUT OF ORDFR  B(1)=,1F4.0,4H  P=,1F4.0)
      ICANCL=1
      RETURN
   70 IF (BLK-636.0)  71,72,73
C***  WRITE REMAINING COEFFICIENTS ON TAPE
   71 WRITE (KFILF3)BLK,V1
      REWIND KFILE3
      GC TO 74
   72 RFWIND KFILE3
```

```
   73 WRITE (KFILE5)BLK,V1
      REWIND KFILE5
   74 WRITE (KFILE4)IDW,C1
      REWIND KFILE4
      REWIND KFILE
      GO TO 77
   75 WRITE (JWRITE,76) L,P
   76 FORMAT (3H L=,I5,20X,3H P=,F5.1)
   77 RETURN
      END


      FUNCTION SQ2F(X1,X2,Y1,Y2,Z1,Z2)
C
C***  FUNCTION SQ2F CALCULATES THE DISTANCE BETWEEN TWO POINTS
C
      X=X1-X2
      Y=Y1-Y2
      Z=Z1-Z2
      RS=Z**2+Y**2+X**2
      SQ2F=SQRT(RS)
      RETURN
      END


      SUBROUTINE PFP3(EPS,MIX)
C
C***  SUBROUTINE PFP3 SOLVES FOR SOURCE DENSITY
C
      DIMENSION VIP(650),S(5,650),COEF(900),VTX(650)
      COMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP(
     1650),AQP(650),REFA,REFBDY,NQUAD,IWRITE,NP,KNEW
      COMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE
     15
      COMMON /SIK/SN(650),VNP(650),IPN(650)
      COMMON /RING/VXX(3,650)
C
      WRITE (JWRITE,1)
    1 FORMAT(1H1,///,24H0SOURCE DENSITY SOLUTION)
      WRITE (JWRITE,2) (TITLE(I),I=1,20)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
    2 FORMAT (1H0,20A4)
      K1=1
      K2=NP
C***  SET CONDITIONS FOR ONSET FLOW OF -1.0+VXX(1,K) IN X DIRECTION.
C***  ZERO IN Y AND Z DIRECTIONS
      FX=-1.0
C***  COMPUTE (1) DOT PRODUCT OF PANEL NORMAL TO ONSET FLOW, VIP(K).
C***  (2) DOT PRODUCT OF PANEL NORMAL TO NONUNIFORM FLOW, VTX(K).
C***  (3) INITIAL SOURCE DENSITY.  AND SET PARTIAL SUM VECTOR TO ZERO
      DO 3 K=1,NP
      VIP(K)=XNP(K)*FX
      VTX(K)=XNP(K)*VXX(1,K)
      S(5,K)=-(VIP(K)+VTX(K)-VNP(K))*.11936
    3 SN(K)=0.
      SN(NP+1)=0.
      SN(NP+2)=0.
      SN(NP+3)=0.
      SN(NP+4)=0.
      WRITE (JWRITE,30) FX
      WRITE (JWRITE,4)
    4 FORMAT (/,12X,37HITERATIVE MATRIX SOLUTION INFORMATION,/,27H0ITERA
     1TION    SUM OF CHANGES,9X,1HA,10X,2HB1,10X,2HB2)
      IT=1
      IC=5
C***  START ITERATION,READ FIRST BLOCK OF COEFS..START LOOP OVER QUADS.
    5 READ (KFILE4)IDW,COEF
C***  (DO LOOP 40) CALCULATES THE SUM OF THE PRODUCTS OF THE INDUCED
C***  VELOCITIES AND THE SOURCE DENSITIES
      J=0
      DO 8 K=1,NP
C***  PICK UP SOURCE DENSITY & START LOOP OVER CENTROID POINTS
      SP=S(IC,K)
      DO 8 KP=1,NP,5
      IF (J-900)  7,6,6
    6 READ (KFILE4)IDW,COEF
      J=0
C***  COMPUTE PARTIAL SUMS FOR NEXT 5 POINTS
    7 SN(KP)=SN(KP)+COEF(J+1)*SP
      SN(KP+1)=SN(KP+1)+COEF(J+2)*SP
      SN(KP+2)=SN(KP+2)+COEF(J+3)*SP
      SN(KP+3)=SN(KP+3)+COEF(J+4)*SP
      SN(KP+4)=SN(KP+4)+COEF(J+5)*SP
```

```
      J=J+5
C***  END LOOP OVER CENTROID POINTS & END LOOP OVER QUADS.
    8 CCNTINUE
C***  CCMPUTE NEW SOURCE DENSITIES
      REWIND KFILE4
      PASS=1.0
      SUM=0.
      DC 9 K=K1,K2
      SN(K)=-(SN(K)+VIP(K)-VNP(K)+VTX(K))/6.28318530B
      TEST=ABS(SN(K)-S(IC,K))
      SUM=SUM+TEST
      IF (TEST.GT.EPS)PASS=-1.0
    9 CCNTINUE
      WRITE (JWRITE,29) IT,SUM
      IF (PASS.EQ.1.0) GO TC 27
      IF (IT.EG.MIX) GC TO 27
      IT=IT+1
      IC=IC-1
      IF (IC.EQ.0) GO TO 11
      DO 10 K=K1,K2
      S(IC,K)=SN(K)
   10 SN(K)=0.
      GC TO 5
   11 A=0.
      B1=0.
      B2=0.
      DA=0.
      D1=0.
      D2=0.
      DC 17 K=K1,K2
      DS9=2*S(1,K)-SN(K)-S(2,K)
      IF (DS9.GT.0.) GO TO 12
      A=A+S(2,K)-S(1,K)
      CA=DA-DS9
      GC TO 13
   12 A=A+S(1,K)-S(2,K)
      DA=DA+DS9
   13 DS1=S(4,K)-S(3,K)
      DS2=S(3,K)-S(2,K)
      DS3=DS1-DS2
      DSS=S(2,K)-S(1,K)
      DS5=DS2-DSS
```

```
      DS6=DS1-DSS
      DS4=DS2-S(1,K)+SN(K)
      DS7=DS3*DS4-DS5*DS6
      DS8=DS6*DS5-DS4*DS3
      IF (DS7.GT.0.) GO TO 14
      B1=B1-DS1*DS4+DS2*DS6
      D1=D1-DS7
      GC TO 15
   14 B1=B1+DS1*DS4-DS2*DS6
      D1=D1+DS7
   15 IF (DS8.GT.0.) GC TO 16
      B2=B2-DS1*DS5+DS2*DS3
      D2=D2-DS8
      GO TO 17
   16 B2=B2+DS1*DS5-DS2*DS3
      D2=D2+DS8
   17 CONTINUE
      A=A/DA
      B1=B1/D1
      B2=B2/D2
      IF (IT.EQ.6) GO TO 23
      AA=ABS(A-AS)
      IF (AA.GT..02) GO TO 20
      DC 18 K=K1,K2
      S(5,K)=A*(SN(K)-S(1,K))+S(1,K)
   18 SN(K)=0.
      WRITE (JWRITE,19)
   19 FCRMAT (29X,17HA EXTRAPCLATION   )
      GC TO 25
   20 BB1=50.*ABS(B1-BS1)
      BB2=50.*ABS(B2-BS2)
      BBB=ABS(B1)+ABS(B2)
      IF ((BB1.GT.BBB).OR.(BB2.GT.BBB)) GO TO 23
      DC 21 K=K1,K2
      S(5,K)=S(2,K)+B1*(S(1,K)-S(2,K))+B2*(SN(K)-S(2,K))
   21 SN(K)=0.
      WRITE (JWRITE,22)
   22 FCRMAT (29X,17H3 EXTRAPOLATICN   )
      GC TO 25
   23 DO 24 K=K1,K2
      S(5,K)=SN(K)
   24 SN(K)=0.
```

```
   25 IC=5
      WRITE (JWRITE,26) A,B1,B2
   26 FORMAT (29X,3E12.3)
      AS=A
      BS1=B1
      BS2=B2
      GC TO 5
   27 DC 28 K=K1,K2
   28 S(1,K)=SN(K)
   29 FCRMAT (4X,I3,E18.5)
   30 FORMAT (/,13H0 X VELOCITY=,F4.1,24H+VXX(1,K)    Y VELOCITY=,4H 0.0
     1,15H     Z VELOCITY=,4H 0.0)
      RETURN
      FND


      SUBROUTINE PFP4(CLPA,CDPA,ICK)
C
C***  SUBROUTINF PFP4 COMPUTES VELOCITIES AND PRESSURE COEFFICIENTS AT
C***  THE PANEL CENTROID POINTS
C
      DIMENSION CV1(1000)
      CCMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP(
     1650),AQP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW
      CCMMCN /RING/VTX(3,650)
      COMMON /S1K/S1(650),VNP(650),IPN(650)
      COMMON /VCK/HVF,SFC,DEP,CPHA,TINF,AOUT,AIN,EDA
      CCMMON /CFSX/CPS(650),CPIN,CPOUT
      COMMON /VXYZ/VX1(650),VY1(650),VZ1(650),VABS(650)
      CCMMON /HOLD/XNH(650),ZNH(650),AQPH(650)
      COMMCN /INOUT/JREAD,JWRITE,KFILE,KFILF1,KFILE2,KFILE3,KFILE4,KFILE
     15
C
      WRITE (JWRITE,1)
    1 FCRMAT (1H1,//,1X,75H COMPUTATION OF VELOCITIES AND PRESSURE COEFF
     1ICIENTS AT THE PANEL CENTROIDS)
C***  INITIALIZE PARAMETERS
      D=-.5/3.14159265
      J=1
      K1=1
      IF (KNEW.EQ.1)KHOLD=NP
```

```
      K2=NP
      BBR=1.0
      CD=0.0
      CL=0.0
      AREA=0.0
C***  READ FIRST BLOCK OF COEFFICIENTS
      ITAPE=KFILE3
      READ (KFILE3)BB,CV1
      IF (BB-BBR)  45,2,45
C***  CALCULATE VELOCITY CCMPONENTS DUE TO ENTIRE FLOW FIELD
    2 DO 3 I=K1,K2
      VX1(I)=-1.0+VTX(1,I)+VNP(I)*XNP(I)-S1(I)*XNP(I)/D
      VY1(I)=VTX(2,I)+VNP(I)*YNP(I)-S1(I)*YNP(I)/D
      VZ1(I)=VTX(3,I)+VNP(I)*ZNP(I)-S1(I)*ZNP(I)/D
    3 CCNTINUE
C***  SFT UP LOOP OVFR QUADS..PICK UP SOURCE,SET UP LOOP OVER CENTROIDS
C***  TO SUM THE PRODUCTS OF THE VELOCITY REFLECTIONS AND THE SOURCE
C***  DENSITIES
      JC=2
    4 S1J=S1(J)
      DO 10 JP=K1,K2
C***  COMPUTE PARTIAL SUM FOR 3 COMPONENTS OF 3 VELOCITIES
      VX1(JP)=VX1(JP)+S1J*CV1(JC)
      VY1(JP)=VY1(JP)+S1J*CV1(JC+1)
      VZ1(JP)=VZ1(JP)+S1J*CV1(JC+2)
      JC=JC+3
C***  READ MORE COEFFICIENTS IF NEEDED
      IF (JC-1000)  10,5,5
    5 JC=2
      IF (BBR-635.0)  6,7,8
    6 READ (KFILE3)BB,CV1
      GO TO 9
    7 REWIND KFILE3
    8 READ (KFILF5)BB,CV1
    9 BBR=BBR+1.
      IF (BBR-BB)  45,10,45
C***  END OF LOOP OVER CENTROIDS THEN END OF LOOP OVER QUADS.
   10 CCNTINUE
      J=J+1
      IF (J-NP)  4,4,11
   11 IF (BBR-635.0)  12,12,13
   12 RFWIND KFILE3
```

```
      GO TO 14
   13 REWIND KFILES
   14 IP=K1+49
C***  THE VELOCITIES AT THE CONTROL POINTS ON THE BODY SURFACE HAVE BEEN
C***  CALCULATFD
      IS=K1
      IPAGE=1
      IIN=0
      ICUT=0
      CPIN=0.0
      CPOUT=0.0
      ACUT=0.0
      AIN=0.0
   15 IF (IP-K2)  16,16,28
C***  COMPUTE PRESSURE COEFFICIENT, ABSOLUTE VALUE OF VELOCITY, CL, & CD
   16 IF (IPAGE.EQ.1) WRITE (JWRITE,17) (TITLE(I),I=1,20),IPAGE
   17 FORMAT (//,1H0,20A4,8H  PAGE =,I5)
      IF (IPAGE.NE.1) WRITE (JWRITE,18) (TITLE(I),I=1,20),IPAGE
   18 FORMAT (1H1,///////,1X,20A4,8H  PAGE =,I5)
      WRITE (JWRITE,19)
   19 FORMAT (8H0 X FLOW)
      WRITE (JWRITE,20)
   20 FORMAT (4H PT.,10X,2HXC,8X,2HYC,8X,2HZC,12X,2HVX,8X,2HVY,8X,2HVZ,9
     1X,5HABS.V,8X,2HCP,6X,6HSOURCE,5X,8HV NORMAL,5X,4HAREA)
      DO 27 I=IS,IP
      VSQ=VX1(I)**2+VY1(I)**2+VZ1(I)**2
      VM=SQRT(VSQ)
      VABS(I)=VM
      CP=1.-VSQ
      CPS(I)=CP
      VNR=VX1(I)*XNP(I)+VY1(I)*YNP(I)+VZ1(I)*ZNP(I)
      IF (KNEW.EQ.2.OR.ICK.GE.2) GO TO 23
C***  CHECK INDICATOR FOR A NONZERO NORMAL VFLOCITY SPECIFICATION
      IF (IPN(I).EQ.0) GO TO 23
      IF ((1+IPN(I)/IABS(IPN(I))).EQ.0) GO TO 21
      GO TO 22
   21 CPIN=CPIN+CP*AQP(I)
      AIN=AIN+AQP(I)
      IIN=IIN+1
      GO TO 23
   22 CPOUT=CPOUT+CP*AQP(I)
      ACUT=ADUT+AQP(I)
```

```
      ICUT=IOUT+1
   23 IF (I.GT.KHOLD) GO TO 25
      IF (KNEW.EQ.2) GO TO 24
      CL=CL+CP*AQP(I)*ZNP(I)
      CD=CD+CP*AQP(I)*XNP(I)
      AREA=AREA+AQP(I)
      GO TO 25
C***  COMPUTE CL, CD, AND AREA FOR WAKE-BODY PORTICN
   24 CL=CL+CP*AQPH(I)*ZNH(I)
      CD=CD+CP*AQPH(I)*XNH(I)
      AREA=AREA+AQPH(I)
C***  WRITE CENTROID POINTS,VFLOCITIES,CP'S,CL-PRESSURE,& CD-PRESSURE
   25 WRITE (JWRITE,26) I,XCP(I),YCP(I),ZCP(I),VX1(I),VY1(I),VZ1(I),VM,C
     1P,S1(I),VNR,AQP(I)
   26 FORMAT (1X,I3,4X,3F10.5,4X,3F10.5,1F13.5,2F11.5,E12.2,2X,E10.3)
   27 CCNTINUE
      IS=IS+50
      IP=IP+50
      IPAGE=IPAGE+1
      IF (K2-IS)  29,28,15
   28 IP=K2
      GC TO 16
C***  ADJUST AND REFERENCE CL AND CD DUE TO SYMMETRY AND REFERENCE AREA
   29 CLPA=2.0*CL
      CDPA=2.0*CD
      CLP=CLPA/REFA
      CDP=CDPA/REFA
      WRITE (JWRITE,30) CLP,CDP,REFA,REBODY
   30 FCRMAT (1H1,//////,18X,35HPRESSURE LIFT AND DRAG COEFFICIENTS,//,2
     11X,29(1H*),/,25X,14HPRESSURE CL = ,F11.5,/,25X,14HPRESSURE CD = ,F
     111.5,/,22X,17HREFERENCE AREA = ,F11.5,/,21X,18HREYNOLDS NUMBER = ,
     1E11.4,/,21X,29(1H*),//)
      IF (KNEW.EQ.2) GO TO 35
      WRITE (JWRITE,31)
   31 FORMAT (1H1,//,2X,2HPT,5X,3HXNH,6X,3HYNH,6X,3HZNH,6X,2HCP,5X,6HVX*
     1XNH,3X,6HVY*YNH,3X,6HVZ*ZNH,5X,4HAQPH,3X,13H2*CP*AQPH*ZNH,3X,13H2*
     1CP*AQPH*XNH,4X,2HPT,//)
      LINE=0
      DC 34 I=1,NP
      VXXN=VX1(I)*XNP(I)
      VYYN=VY1(I)*YNP(I)
      VZZN=VZ1(I)*ZNP(I)
```

```
      CAZ=2.0*CPS(I)*AQP(I)*ZNP(I)
      CAX=2.0*CPS(I)*AQP(I)*XNP(I)
      LINE=LINE+1
      IF (LINE.LE.50) GO TO 32
      LINE=0
      WRITE (JWRITE,31)
   32 WRITE (JWRITE,33) I,XNP(I),YNP(I),ZNP(I),CPS(I),VXXN,VYYN,VZZN,AQP
     1(I),CAZ,CAX,I
   33 FORMAT (1X,I4,7F9.5,F8.3,3X,F10.7,6X,F10.7,5X,I4)
   34 CCNTINUE
      GO TO 41
   35 WRITE (JWRITE,36)
   36 FORMAT (1H1,//,2X,2HPT,5X,3HXNP,6X,3HYNP,6X,3HZNP,6X,2HCP,5X,6HVX*
     1XNP,3X,6HVY*YNP,3X,6HVZ*ZNP,5X,4HAQP  ,3X,13H2*CP*AQPH*ZNH,3X,13H2*
     1CP*AQPH*XNH,4X,2HPT,//)
      LINE=0
      DO 40 I=1,NP
      VXXN=VX1(I)*XNP(I)
      VYYN=VY1(I)*YNP(I)
      VZZN=VZ1(2)*ZNP(I)
      LINE=LINE+1
      IF (LINE.LF.50) GO TO 37
      LINE=0
      WRITF (JWRITE,36)
   37 IF (I.GT.KHOLD) GO TO 38
      CAZ=2.0*CPS(I)*AQPH(I)*ZNH(I)
      CAX=2.0*CPS(I)*AQPH(I)*XNH(I)
      WRITE (JWPITE,33) I,XNP(I),YNP(I),ZNP(I),CPS(I),VXXN,VYYN,VZZN,AQP
     1(I),CAZ,CAX,I
      GC TO 40
   38 WRITE (JWRITE,39) I,XNP(I),YNP(I),ZNP(I),CPS(I),VXXN,VYYN,VZZN,AQP
     1(I),I
   39 FCRMAT (1X,I4,7F9.5,F8.3,34X,I4)
   40 CONTINUF
   41 IF (KNEW.EQ.2.OR.ICK.GE.2) GO TO 47
C***  TEST FOR ERRORS
      IF (IIN.EQ.0.OR.IOUT.EQ.0)ICK=2
      IF (ICK.EQ.1) GO TO 44
      WRITF (JWRITE,42)
   42 FCRMAT (1X,///,1X,131HAT LEAST ONE INLET(OUTLET) WAS SPECIFIED WIT
     1HOUT AT LEAST ONE OUTLET(INLET)...ASSUMING NO INLETS OR OUTLETS...
     1EXECUTION PROCEEDING..///)
```

```
      DO 43 I=1,NP                                                      PF4 194
   43 VNP(I)=0.0                                                        PF4 195
      GO TO 47                                                          PF4 196
C***  CCMPUTE AVERAGE CP'S                                              PF4 197
   44 CPIN=CPIN/AIN                                                     PF4 198
      CPOUT=CPOUT/AOUT                                                  PF4 199
      GO TO 47                                                          PF4 200
   45 WRITE (JWRITE,46) ITAPE,BBR,BB                                    PF4 201
   46 FORMAT (6HITAPE ,I2,16H OUT OF POSITION/I4,6F6.1)                 PF4 202
   47 RETURN                                                            PF4 203
      END                                                               PF4 204


      SUBROUTINE VINOUT(JWRITE,VINF,ROE,NQE,VI)                         VNT   1
C                                                                       VNT   2
C***  SUBROUTINE VINOUT CALCULATES THE INTERIOR PRESSURE COEFFICIENTS   VNT   3
C***  AND THE NORMAL VELOCITIES ON SPECIFIED PANELS                     VNT   4
C                                                                       VNT   5
      COMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP( VNT   6
     1650),AQP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW                    VNT   7
      COMMON /S1K/SN(650),VNP(650),IPN(650)                             VNT   8
      COMMON /VCK/HVF,SFC,DEP,CPHA,TINF,AOUT,AIN,EOA                    VNT   9
      COMMON /CPSX/CPS(650),CPIN,CPOUT                                  VNT  10
      COMMON /CANCEL/ICANCL,NODE                                        VNT  11
C                                                                       VNT  12
      IPASS=0                                                           VNT  13
      WRITE (JWRITE,1) CPIN,AIN,CPOUT,AOUT                              VNT  14
    1 FORMAT (1H1,//,5X,75H*** CALCULATION OF INTERIOR PRESSURE COEFFICI VNT  15
     1ENTS AND NORMAL VELOCITIES ***,//,9X,29HAVERAGE CP ON INLET PANELS VNT  16
     1 =,1PE16.9,3X,13HINLET AREA  =,1PE16.9,/,9X,29HAVERAGE CP ON EXHA VNT  17
     1UST PANELS=,1PE16.9,3X,13HEXHAUST AREA=,1PE16.9,/)                VNT  18
C***  CALCULATE PRESSURE COEFFICIENTS                                   VNT  19
    2 CPXX=(AOUT**2*CPOUT+(EOA*AIN)**2*CPIN/(AIN**2+EOA**2))*((AIN**2+EO VNT  20
     1A**2)/((AOUT*AIN)**2+(AOUT**2+AIN**2)*EOA**2))                    VNT  21
      CPX=(AIN**2*CPIN+EOA**2*CPXX)/(AIN**2+EOA**2)                     VNT  22
      IF (CPX.GT.CPXX) GO TO 3                                          VNT  23
      IF (ABS(CPX-CPXX).LT.1.0E-05)CPXX=CPX                             VNT  24
    3 WRITE (JWRITE,4) CPX,CPXX                                         VNT  25
    4 FORMAT (10X,5HCPX =,1PE16.9,/,10X,5HCPXX=,1PE16.9,/)              VNT  26
      IF (CPX.LE.CPIN.AND.CPX.GE.CPXX) GO TO 7                          VNT  27
      WRITE (JWRITE,5)                                                  VNT  28
```

```
    5 FORMAT (12X,32HCPX > CPIN OR CPX < CPXX.  STOP..///)
    6 ICANCL=1
      RETURN
    7 IF (CPXX.LE.CPX.AND.CPXX.GE.CPOUT) GO TO 9
      WRITE (JWRITE,8)
    8 FORMAT (12X,34HCPXX > CPX OR CPXX < CPOUT.  STOP..///)
      GO TO 6
C***  ADJUST EFFECTIVE ORIFICE AREA EFFECT
    9 IPASS=IPASS+1
      IF (IPASS.GT.1) GO TO 14
      PNET=DEP*(778.0*HVF*SFC-1.0)
      IF (PNET)  10,10,12
   10 WRITE (JWRITE,11) PNET
   11 FORMAT (12X,23HDEP*(778.0*HVF*SFC-1) =,1PE16.9,/)
      IF (PNET.LT.0.0) GO TO 6
      GO TO 14
   12 TR=PNET/(ROE*32.2*AIN*CPHA*778.0*TINF*VINF*SQRT(CPIN-CPX))+1.0
      TR=SQRT(1.0/TR)
      ECA=EOA*TR
      WRITE (JWRITE,13) TR,ECA
   13 FORMAT (9X,58HADJUSTMENT DUE TO HEATING OF FLOW: TEMPERATURE RATIO
     1 =,1PE16.9,/,44X,23HEFFECTIVE ORIFICE AREA=,1PE16.9,//)
      GO TO 2
C***  CALCULATE NORMAL VELOCITIES
   14 DO 24 I=1,NQE
      ITEST=3
      IF (IPN(I).EQ.0) GO TO 24
      IF ((1+IPN(I)/IABS(IPN(I))).EQ.0) GO TO 15
      GO TO 16
   15 X1=CPS(I)-CPX
      IF (X1.LT.0.0)ITEST=1
      VNP(I)=-SQRT(ABS(X1))
      GO TO 17
   16 X1=CPXX-CPS(I)
      IF (X1.LT.0.0)ITEST=2
      VNP(I)=SQRT(ABS(X1))
   17 GO TO (18,20,22),ITEST
   18 WRITE (JWRITE,19) I,VNP(I),I
   19 FORMAT (10X,24HNORMAL VELOCITY AT PANEL,I4,3H = ,E14.7,10X,11HNOTE
     1: ( CP(,I4,11H)-CPX )<0.0,/)
      GO TO 24
   20 WRITE (JWRITE,21) I,VNP(I),I
```

```
 21 FORMAT (10X,24HNORMAL VELOCITY AT PANEL,I4,3H = ,E14.7,10X,16HNOTE  VNT
   1: ( CPXX-CP(,I4,6H))<0.0,/)                                           VNT
    GC TO 24                                                              VNT
 22 WRITE (JWRITE,23) I,VNP(I)                                            VNT
 23 FORMAT (10X,24HNORMAL VELOCITY AT PANEL,I4,3H = ,E14.7,/)            VNT
 24 CONTINUE                                                              VNT
    RETURN                                                                VNT
    END                                                                   VNT


    SUBROUTINE PUNCH(NMIN,NMAX,MMIN,MMAX,X,Y,Z,JPUNCH,JWRITE)            PCH
C                                                                         PCH
C***    SUBROUTINE PUNCH CONVERTS AND PUNCHES THE (BODY+WAKEBODY) DATA    PCH
C***    BACK TO A COMPATIBLE PLOT PROGRAM DATA SET FOR THE FUSELAGE       PCH
C                                                                         PCH
    DIMENSION X(651),Y(651),Z(651),XX(31,31),YY(31,31),ZZ(31,31)         PCH
C                                                                         PCH
C***    SET INITIAL PARAMETERS                                            PCH
    NS=1                                                                  PCH
    L=C                                                                   PCH
    LMN=1                                                                 PCH
C***    CONVERT TO ORIGINAL AXIS SYSTEM                                   PCH
    ZMAX=-1000.0                                                          PCH
    ZMIN=1000.0                                                           PCH
    DO 1 N=NMIN,NMAX                                                      PCH
    DC 1 M=MMIN,MMAX                                                      PCH
    L=L+1                                                                 PCH
    IF (M.GT.MMIN.AND.M.LT.MMAX) GO TO 1                                  PCH
    IF (ZMAX.LT.Z(L))ZMAX=Z(L)                                            PCH
    IF (ZMIN.GT.Z(L))ZMIN=Z(L)                                            PCH
  1 CCNTINUE                                                              PCH
    ZORG=ABS(ZMIN)+ABS(ZMAX)                                              PCH
    L=0                                                                   PCH
    DO 3 N=NMIN,NMAX                                                      PCH
    XP=X(1)-X(LMN)                                                        PCH
    DC 2 M=MMIN,MMAX                                                      PCH
    L=L+1                                                                 PCH
    YP=Y(L)                                                               PCH
    ZP=Z(L)                                                               PCH
    XX(N,M)=XP                                                            PCH
    YY(N,M)=YP                                                            PCH
```

```
      ZZ(N,M)=ZP+ZDRG                                                    PCH  32
    2 CONTINUE                                                           PCH  33
      LMN=LMN+MMAX                                                       PCH  34
    3 CONTINUE                                                           PCH  35
C***  PUNCH CARDS FOR NEW FUSELAGE                                       PCH  36
      WRITE (JPUNCH,4) (XX(J,1),J=NMIN,NMAX)                             PCH  37
    4 FORMAT (10F7.4)                                                    PCH  38
      DO 5 I=NMIN,NMAX                                                   PCH  39
      WRITE (JPUNCH,4) (YY(I,J),J=MMIN,MMAX)                             PCH  40
      WRITE (JPUNCH,4) (ZZ(I,J),J=MMIN,MMAX)                             PCH  41
    5 CONTINUE                                                           PCH  42
      RETURN                                                             PCH  43
      END                                                                PCH  44


      SUBROUTINE PFP5(MM,NN,AREAAV,AVDELS,AVXCG,CDFA)                    PF5   1
C                                                                        PF5   2
C***  SUBROUTINE PFP5 COMPUTES ON-BODY STREAMLINES                       PF5   3
C                                                                        PF5   4
      DIMENSION XC1(650),YC1(650),XC2(650),YC2(65),XC3(650),XC4(650),YC  PF5   5
     14(650),X3(650),Y3(650),Z3(650),X4(650),Y4(650),Z4(650),DMX(650),XL PF5   6
     1(75),YL(75),ZL(75),YC3(650),SF(5),XCR(5),YCR(5),STML(75),UABS(75), PF5   7
     1SKIN(650),DSTAR(650),NOTEST(5)                                     PF5   8
      EQUIVALENCE (YC3(1),YC2(1)),(XL(1),XNH(1)),(YL(1),AQPH(1)),(ZL(1), PF5   9
     1ZNH(1))                                                            PF5  10
      COMMON /HOLD/XNH(650),ZNH(650),AQPH(650)                           PF5  11
      COMMON TITLF(20),XCP(650),YCP(650),ZCP(650),XNP(65),YNP(650),ZNP(  PF5  12
     1650),AQP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW                     PF5  13
      COMMON /VXYZ/VX1(650),VY1(650),VZ1(650),VABS(65)                   PF5  14
      COMMON /BL/VOV(75),SS(75),VINF,VO,ROE,DELS(150),CFI(150),THT(150), PF5  15
     1STOTAL,KKK                                                         PF5  16
      COMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFIL5 PF5  17
      COMMON /RING/VTX(3,650)                                            PF5  18
      COMMON /CANCEL/ICANCL,NODE                                         PF5  19
      NAMELIST /NAM1/NP,NQ,DMX/NAM2/II,NSTART,NEND,NDEL/NAM3/LL,MID,DIRT PF5  20
     1,JI/NAM4/AF,UX,UY,UZ,CP,NQ,NCP,LNQ,JSTOP,J,JL,XLT,YLT,XL,YL,ZL/NAM PF5  21
     15/NQ,IQT,NR,NU,LXQ,UYQ,UZQ,UXP,UYR,UZR,UXU,UYU,UZU,UQ,VQ,CSR,UT,VT/ PF5  22
     1,XXR,XYR,UR,YXR,YYR,VR,UU,CSU,VU,XD,YD,ZD,XT,YTT,ZT,YT,XR,YR/NAM6/ PF5  23
     1XD,YD,ZD,XU,YT,ZT,YU,DEN,U1,U2,V1,V2,USL,VSL,UXP,UYP,UZP,VSQD/NAM7 PF5  24
     1/DEN,CXY,CYY,CXX,CC,NQ,XCR,YCR,SF,TEST/NAM8/N,XM,YM,SFM,AC,BC/NAM9 PF5  25
     1/XNTP,YNTP,TESTP/NAM10/TEST,XNT,YNT/NAM11/UX,UY,UZ,CP,UABS/NAM12/J PF5  26
```

```
     1L,J,JI,USL,VSL,UX,UY,UZ,VSQD,CORD,STML,CP,AF,LNQ/NAM13/KMOD,NTEST,
     1NQ,NQTEST,M/NAM14/M,I,TEST,DS1,DS2,DS3,DS4,XLT,YLT,ZLT,ZSQ/NAM15/T
     1EST,RC1,RC2,RC3,RC4/NAM16/TEST/NAM17/TEST/NAM18/TEST/NAM19/DIRT,JI
     1,JMIN/NAM20/JMAX,SSS,STML/NAM21/KEY,JJ,KK,JMIN/NAM22/XL,YL,ZL,UABS
     1/NAM23/KEY,KK,JJ,JMAX/NAM24/JMN,JMX,AF,L/NAM25/J,L,STML,UABS,AF,MI
     1D/NAM26/L,K,STML,UABS/NAM27/L,JMAX,JMIN,STML,XL,YL,ZL,UABS
      COMMON /S1K/SN(650),VNP(650),IPN(650)
C
      WRITE (JWRITE,1)
    1 FORMAT (1H1,///,35HOCALCULATION OF ON-BODY STREAMLINES)
      NB=(NP+11)/12
      NLIN=NQUAD
      DO 2 I=1,NB
      IFN=I*12
      IS=IFN-11
C***  READ PANEL GECMETRY FROM TAPE
      READ (KFILE)Q,(XC1(J),YC1(J),XC2(J),YC2(J),XC3(J),XC4(J),YC4(J),X3
     1(J),Y3(J),Z3(J),X4(J),Y4(J),Z4(J),(SKIP,K=1,3),J=IS,IFN)
      NQ=Q
      IF (NQ.NE.IS) GO TO 68
    2 CCNTINUE
      NOD1=NP+1
      NOD2=NOD1
      IWRTS=IWRITE
      IF (NODE.GT.NP)NODE=0
      IF (NODE.LE.0) GO TO 3
      NOD1=NODE-2
      NOD2=NODE+2
      IF (NOD1.LE.0)NOD1=1
      IF (NOD2.GT.NP)NOD2=NP
C***  FOR EACH QUAD, COMPUTE SQUARE OF THE RADIUS OF A CIRCLE ABOUT THE
C***  CENTROID WHICH ENCLOSFS THE QUAD WITH 1% TO SPARE
    3 DO 4 I=1,NP
      D1=(XC1(I)**2+YC1(I)**2)*1.01
      D2=(XC2(I)**2+YC2(I)**2)*1.01
      D3=(XC3(I)**2+YC3(I)**2)*1.01
      D4=(XC4(I)**2+YC4(I)**2)*1.01
      IF (I.GT.75) GO TO 4
      XL(I)=0.0
      YL(I)=0.0
      ZL(I)=0.0
      STML(I)=0.0
```

```
      UABS(I)=0.0
      VCV(I)=0.0
      SS(I)=0.0
    4 DMX(I)=AMAX1(D1,D2,D3,D4)
      IF (NODE.NE.0) WRITE (JWRITE,NAM1)
      MID=70
      MAXJ=75
      MINJ=1
      VXI=-1.0
      IF (NLIN.EQ.0) GO TO 70
      WRITF (JWRITE,5) (TITLF(I),I=1,20)
    5 FORMAT (///,1H0,20A4)
      IF (IWRITE.GT.1) GO TO 7
      WRITE (JWRITE,6) NLIN
    6 FORMAT (1X,47HFOR V INFINITY = -1.0+VXX(1,K),0.0,0.0, COMPUTE,I4,5
     10H STRFAMLINFS STARTING AT EACH PANEL CENTROID POINT)
C***  CALCULATE STREAMLINES FOR EACH PANEL CENTRCID OMITTING TRIANGULAR
C***  PANELS AT ENDS. LINES FOUND FROM NOSE TO 3 PTS PAST PANE_ CENTROID
    7 IUPPFR=NLIN+1-2*(MM-1)
      ILCWEF=2*(MM-1)
      NLINM1=NLIN-1
      DO 61 II=1,3
      IF (II-2) 8,9,10
    8 NSTART=2
      NEND=2*(MM-1)
      NDEL=2
      IWRTS=IWRITE
      GO TO 11
    9 NSTART=NEND+1
      NEND=(NN-3)*(MM-1)
      NDEL=1
      GO TO 11
   10 NSTART=NEND+1
      NEND=(NN-1)*(MM-1)-1
      NDEL=2
   11 IF (NODE.NE.0) WRITE (JWRITE,NAM2)
      DO 55 LL=NSTART,NEND,NDEL
      IWRITE=IWRTS
      IF (LL.LT.NOD1) GO TO 12
      IF (LL.GT.NOD2) GO TO 12
      IWRITE=-1
   12 MID=70
```

```
      DIRT=-1.
      JI=-1
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM3)
   13 AF=1.
      UX=0.
      UY=0.
      UZ=0.
      CP=0.
      STML(MID)=0.
      NQ=LL
      NCP=LL
      LNQ=NQ
      XL(MID)=XCP(LL)
      YL(MID)=YCP(LL)
      ZL(MID)=ZCP(LL)
      JSTOP=MID+3
      J=MID
      JL=J
      XLT=(XL(J)-XCP(NQ))*X3(NQ)+(YL(J)-YCP(NQ))*Y3(NQ)+(ZL(J)-ZCP(NQ))*
     1Z3(NQ)
      YLT=(XL(J)-XCP(NQ))*X4(NQ)+(YL(J)-YCP(NQ))*Y4(NQ)+(ZL(J)-ZCP(NQ))*
     1Z4(NQ)
      XL(J)=XLT*X3(NQ)+YLT*X4(NQ)+XCP(NQ)
      YL(J)=XLT*Y3(NQ)+YLT*Y4(NQ)+YCP(NQ)
      ZL(J)=XLT*Z3(NQ)+YLT*Z4(NQ)+ZCP(NQ)
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM4)
   14 IQT=MOD(NQ,4)+1
      GO TO (18,15,16,17),IQT
   15 NR=NQ+1
      NU=NQ+2
      GO TO 19
   16 NR=NQ+2
      NU=NQ-1
      GO TO 19
   17 NR=NQ-2
      NU=NQ+1
      GO TO 19
   18 NR=NQ-1
      NU=NQ-2
   19 UXQ=-(VXI*VX1(NQ))
      UYQ=-(VXI*VY1(NQ))
      UZQ=-(VXI*VZ1(NQ))
```

```
      UXR=-(VXI*VX1(NR))
      UYR=-(VXI*VY1(NR))
      UZR=-(VXI*VZ1(NR))
      UXU=-(VXI*VX1(NU))
      UYU=-(VXI*VY1(NU))
      UZU=-(VXI*VZ1(NU))
C***  TRANSFORM VELOCITIES TO QUAD. SYSTEM
      UQ=UXQ*X3(NQ)+UYQ*Y3(NQ)+UZQ*Z3(NQ)
      VQ=UXQ*X4(NQ)+UYG*Y4(NQ)+UZQ*Z4(NQ)
      CSR=1./(XNP(NQ)*XNP(NR)+YNP(NQ)*YNP(NR)+ZNP(NQ)*ZNP(NR))
      UT=UXR*X3(NR)+UYR*Y3(NR)+UZR*Z3(NR)
      VT=(UXR*X4(NR)+UYR*Y4(NR)+UZR*Z4(NR))*CSR
      XXR=(X3(NR)*X3(NQ)+Y3(NR)*Y3(NQ)+Z3(NR)*Z3(NQ))
      XYR=(X4(NR)*X3(NQ)+Y4(NR)*Y3(NQ)+Z4(NR)*Z3(NQ))
      UR=UT*XXR+VT*XYR
      YXR=(X3(NR)*X4(NQ)+Y3(NR)*Y4(NQ)+Z3(NR)*Z4(NQ))
      YYR=(X4(NR)*X4(NQ)+Y4(NR)*Y4(NQ)+Z4(NR)*Z4(NQ))
      VR=UT*YXR+VT*YYR
      UU=UXU*X3(NQ)+UYU*Y3(NQ)+UZU*Z3(NQ)
      CSU=(XNP(NQ)*XNP(NU)+YNP(NQ)*YNP(NU)+ZNP(NQ)*ZNP(NU))
      VU=(UXU*X4(NQ)+UYU*Y4(NQ)+UZU*Z4(NQ))/CSU
C***  FIND RELATIVE COORDINATES OF NEIGHBORING QUADS.
      XD=XCP(NR)-X ?(NQ)
      YD=YCP(NR)-YC?(NQ)
      ZD=ZCP(NR)-ZCP(NQ)
      XT=XD*X3(NR)+YD*Y3(NR)+ZD*Z3(NR ,
      YTT=XD*X4(NR)+YD*Y4(NR)+ZD*Z4(NR)
      ZT=XD*XNP(NR)+YD*YNP(NR)+ZD*ZNP(NR)
      YT=(-4.*SQRT(YTT**2+ZT**2)+YTT*CSR+YTT)*CSR*.16666667
      XR=XT*XXR+YT*XYR
      YR=XT*YXR+YT*YYR
      IF (IWRITE.LT.0) WRITE (JWRITE.NAM5)
      XD=XCP(NU)-XCP(NQ)
      YD=YCP(NU)-YCP(NQ)
      ZD=ZCP(NU)-ZCP(NQ)
      XU=XD*X3(NQ)+YD*Y3(NQ)+ZD*Z3(NQ)
      YT=XD*X4(NQ)+YD*Y4(NQ)+ZD*Z4(NQ)
      ZT=XD*XNP(NQ)+YD*YNP(NQ)+ZD*ZNP(NQ)
      YU=(4.*SQRT(YT**2+ZT**2)+YT/CSU+YT)*.16666667
C***  FIND COEFFICIENTS OF VELOCITY FUNCTIONS
      DEN=1./(XR*YU-XU*YR)
      U1=((UR-UQ)*YU-(UU-UQ)*YR)*DEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      U2=-((UR-UQ)*XU-(UU-UQ)*XR)*DEN
      V1=((VR-VQ)*YU-(VU-VQ)*YR)*DEN
      V2=-((VR-VQ)*XU-(VU-VQ)*XR)*DEN
C***  FIND VELOCITY AT STREAMLINE POINT
      USL=UQ+U1*XLT+U2*YLT
      VSL=VQ+V1*XLT+V2*YLT
      UXP=USL*X3(NQ)+VSL*X4(NQ)
      UYP=USL*Y3(NQ)+VSL*Y4(NQ)
      UZP=USL*Z3(NQ)+VSL*Z4(NQ)
      VSQD=USL**2+VSL**2
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM6)
      DEN=VSQD*SQRT(VSQD)
C***  FIND LOCAL STREAM FUNCTION
      CXY=(U1*VQ**2-V2*UQ**2)/VSQD
      CYY=U2-UQ*VQ*(U1+V2)/VSQD
      CXX=U2-CYY-V1
      CC=XLT*VQ-YLT*UQ-CXY*XLT*YLT-CYY*YLT**2-CXX*XLT**2
C***  FIND STREAM FUNCTION AT CORNER POINTS
      XCR(1)=XC1(NQ)
      XCR(2)=XC2(NQ)
      XCR(3)=XC3(NQ)
      XCR(4)=XC4(NQ)
      XCR(5)=XCR(1)
      YCR(1)=YC1(NQ)
      YCR(2)=YC2(NQ)
      YCR(3)=YC3(NQ)
      YCR(4)=YC4(NQ)
      YCR(5)=YCR(1)
      DO 20 N=1,4
   20 SF(N)=CO-VQ*XCR(N)+UQ*YCR(N)+CXY*XCR(N)*YCR(N)+CYY*YCR(N)**2+CXX*X
     1CR(N)**2
      SF(5)=SF(1)
      TEST=0.
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM7)
      DO 23 N=1,4
      IF (SF(N)*SF(N+1).GE.0.) GO TO 23
      XM=(XCR(N)+XCR(N+1))*.5
      YM=(YCR(N)+YCR(N+1))*.5
C***  FIND INTERSECTION WITH SIDE OF QUAD.
      SFM=CO-VQ*XM+UQ*YM+CXY*XM*YM+CYY*YM**2+CXX*XM**2
      AC=2.*(SF(N)-2.*SFM+SF(N+1))
      BC=3.*SF(N)-4.*SFM+SF(N+1)
```

```
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM8)                              PF5 237
      IF (AC.EQ.0) GO TO 21                                             PF5 238
      SR=SQRT(BC**2-4.*AC*SF(N))                                        PF5 239
      TP=(BC+SR)/(2.*AC)                                                PF5 240
      IF (TP.LE.1..AND.TP.GE.0.) GO TO 22                               PF5 241
      TP=(BC-SR)/(2.*AC)                                                PF5 242
      IF (TP.LE.1.0.AND.TP.GE.0.0) GO TO 22                             PF5 243
   21 IF (BC.EQ.0) GO TO 23                                             PF5 244
      TP=SF(N)/BC                                                       PF5 245
      IF (TP.GT.1.0.OR.TP.LT.0.0) GO TO 23                              PF5 246
   22 XNTP=(1.-TP)*XCR(N)+TP*XCR(N+1)                                   PF5 247
      YNTP=(1.-TP)*YCR(N)+TP*YCR(N+1)                                   PF5 248
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM9)                              PF5 249
      TESTP=((XNTP-XLT)*UQ+(YNTP-YLT)*VQ)*DIRT                          PF5 250
      IF (TESTP.LE.TEST) GO TO 23                                       PF5 251
      TEST=TESTP                                                        PF5 252
      XNT=XNTP                                                          PF5 253
      YNT=YNTP                                                          PF5 254
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM10)                             PF5 255
   23 CONTINUE                                                          PF5 256
      IF (TEST.EQ.0.) GO TO 34                                          PF5 257
C***  AVERAGE LAST VELOCITY AND CURVATURE                               PF5 258
      UX=(UX+UXP)*AF                                                    PF5 259
      UY=(UY+UYP)*AF                                                    PF5 260
      UZ=(UZ+UZP)*AF                                                    PF5 261
      CP=1.-(UX**2+UY**2+UZ**2)                                         PF5 262
      UABS(J)=SQRT(1.-CP)                                               PF5 263
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM11)                             PF5 264
      IF (J.GE.JSTOP) GO TO 35                                          PF5 265
C***  COMPUTE VELOCITY AT NEXT POINT                                    PF5 266
      JL=J                                                              PF5 267
      J=J+JI                                                            PF5 268
      USL=UQ+XNT*U1+YNT*U2                                              PF5 269
      VSL=VQ+XNT*V1+YNT*V2                                              PF5 270
      UX=USL*X3(NQ)+VSL*X4(NQ)                                          PF5 271
      UY=USL*Y3(NQ)+VSL*Y4(NQ)                                          PF5 272
      UZ=USL*Z3(NQ)+VSL*Z4(NQ)                                          PF5 273
      VSQD=USL**2+VSL**2                                                PF5 274
      CORD=SQRT((XNT-XLT)**2+(YNT-YLT)**2)*DIRT                         PF5 275
      STML(J)=STML(JL)+CORD                                             PF5 276
      CP=1.-VSQD                                                        PF5 277
      UABS(J)=SQRT(1.-CP)                                               PF5 278
```

```
      AF=.5
      LNQ=NQ
      IF (IWRITE.LT.C) WRITE (JWRITE,NAM12)
      XL(J)=XNT*X3(NQ)+YNT*X4(NQ)+XCP(NQ)
      YL(J)=XNT*Y3(NQ)+YNT*Y4(NQ)+YCP(NQ)
      ZL(J)=XNT*Z3(NQ)+YNT*Z4(NQ)+ZCP(NQ)
      IF (J.LE.MINJ.OR.J.GE.MAXJ) GO TO 54
C***  PROCEDURE FOR FINDING NEXT QUAD. WAS MODIFIED SO THAT, DE?ENDING
C***  ON THE DIRECTION, THERE ARE ONLY 5 POSSIBLE QUADS. TO TEST
      KMCD=MOD(NQ,2)
      NTEST=5
      IF (DIRT)  26,24,29
   24 WRITE (JWRITE,25) DIRT
   25 FORMAT (1H1,1X,7HDIRT = ,F10.7,26HPROGRAM TERMINATED IN PFP5)
      ICANCL=1
      RETURN
   26 IF (KMOD)  27,28,27
C***  LAST QUAD. IS ODD & WE ARE TESTING AGAINST THE STREAM DIRECTION
   27 NQTEST(1)=NQ-2*(MM-1)+1
      NQTEST(2)=NQ+2
      NQTEST(3)=NQ-2
      NQTEST(4)=NQTEST(1)+2
      NQTEST(5)=NQTEST(1)-2
      IF (NQ.NE.NLINM1) GO TO 32
      NQTEST(2)=NQTEST(5)
      NTEST=3
      GO TO 32
C***  LAST QUAD. IS EVEN & WE ARE TESTING AGAINST THE STREAM DIRECTION
   28 NQTEST(1)=NQ-1
      NCTEST(2)=NQ+2
      NQTEST(3)=NQ-2
      NQTEST(4)=NQ+1
      NQTEST(5)=NQ-3
      IF (NQ.GT.ILOWER) GO TO 32
      NQTEST(1)=NQTEST(2)
      NQTEST(2)=NQTEST(3)
      NTEST=2
      IF (NQ.EQ.2)NTEST=1
      GO TO 32
   29 IF (KMOD)  30,31,30
C***  LAST QUAD IS ODD & WE ARE TESTING IN THE STREAM DIRECTION
   30 NQTEST(1)=NQ+1
```

```
      NQTEST(2)=NQ+2
      NQTEST(3)=NQ-2
      NQTEST(4)=NQ+3
      NQTEST(5)=NQ-1
      IF (NQ.LT.IUPPER) GO TO 32
      NTEST=2
      NQTEST(1)=NQTEST(3)
      IF (NQ.EQ.NLINM1)NTEST=1
      GO TO 32
C***  LAST QUAD IS EVEN & WE ARE TESTING IN THE STREAM DIRECTION
   31 NQTEST(1)=NQ+2*(MM-1)-1
      NQTEST(2)=NQ+2
      NQTEST(3)=NQ-2
      NQTEST(4)=NQTEST(1)+2
      NQTEST(5)=NQTEST(1)-2
      IF (NQ.NE.2) GO TO 32
      NTEST=4
      NQTEST(3)=NQTEST(5)
   32 M=1
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM13)
   33 I=NQTEST(M)
      NQ=I
      TEST=(XL(J)-XCP(I))**2+(YL(J)-YCP(I))**2+(ZL(J)-ZCP(I))**2-DMX(I)
      IF (TEST.GT.0.) GO TO 34
      DS1=(XC1(I)-XC2(I))**2+(YC1(I)-YC2(I))**2
      DS2=(XC2(I)-XC3(I))**2+(YC2(I)-YC3(I))**2
      DS3=(XC3(I)-XC4(I))**2+(YC3(I)-YC4(I))**2
      DS4=(XC4(I)-XC1(I))**2+(YC4(I)-YC1(I))**2
      XLT=(XL(J)-XCP(I))*X3(I)+(YL(J)-YCP(I))*Y3(I)+(ZL(J)-ZCP(I))*Z3(I)
      YLT=(XL(J)-XCP(I))*X4(I)+(YL(J)-YCP(I))*Y4(I)+(ZL(J)-ZCP(I))*Z4(I)
      ZLT=(XL(J)-XCP(I))*XNP(I)+(YL(J)-YCP(I))*YNP(I)+(ZL(J)-ZCP(I))*ZNP
     1(I)
      ZSQ=ZLT**2
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM14)
      TEST=ZSQ-.1*DMX(I)
      IF (TEST.GT.0.) GO TO 34
      RC1=SQRT(ZSQ+(XLT-XC1(I))**2+(YLT-YC1(I))**2)
      RC2=SQRT(ZSQ+(XLT-XC2(I))**2+(YLT-YC2(I))**2)
      RC3=SQRT(ZSQ+(XLT-XC3(I))**2+(YLT-YC3(I))**2)
      RC4=SQRT(ZSQ+(XLT-XC4(I))**2+(YLT-YC4(I))**2)
      TEST=((RC1+RC2)**2)-DS1*1.3
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM15)
```

```
      IF (TEST.LT.0.) GC TO 14
      TEST=((RC2+RC3)**2)-DS2*1.3
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM16)
      IF (TEST.LT.0.) GO TO 14
      TEST=((RC3+RC4)**2)-DS3*1.3
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM17)
      IF (TEST.LT.0.) GO TO 14
      TEST=((RC4+RC1)**2)-DS4*1.3
      IF (IWRITF.LT.0) WRITF (JWRITE,NAM18)
      IF (TEST.LT.0.) GO TO 14
   34 M=M+1
      IF (M.LE.NTEST) GO TO 33
   35 IF (DIRT.GT.0.) GC TO 36
C***  CHANGE THF DIRFCTION IN WHICH THE STREAMLINE IS TRACED
      DIRT=1.
      JI=1
      JMIN=J
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM19)
      GO TO 13
   36 JMAX=J
      SSS=STML(JMIN)
      DO 37 J=JMIN,JMAX
   37 STML(J)=STML(J)-SSS
      IF (IWRITE.LT.0) WRITE (JWRITF,NAM20)
      IF ((STML(JMIN+3)-STML(JMIN)).LT.8.*(STML(JMIN+1)-STML(JMIN))) GO
     1TO 39
      KFY=1
      JJ=JMIN
      KK=JMIN+1
      JMIN=JMIN+1
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM21)
   38 XL(KK)=(XL(JJ)+XL(KK))/2.0
      YL(KK)=(YL(JJ)+YL(KK))/2.0
      ZL(KK)=(ZL(JJ)+ZL(KK))/2.0
      UABS(KK)=(UABS(JJ)+UABS(KK))/2.0
      IF (IWRITF.LT.0) WRITE (JWRITE,NAM22)
      IF (KFY.FQ.1) GO TO 39
      GO TO 40
   39 IF ((STML(JMAX)-STML(JMAX-3)).LT.8.*(STML(JMAX)-STML(JMAX-1))) GO
     1TO 40
      KEY=2
      KK=JMAX-1
```

```
      JJ=JMAX                                                            PF5 405
      JMAX=JMAX-1                                                        PF5 406
      IF (IWRITE.LT.0) WRITF (JWRITE,NAM23)                              PF5 407
      GO TO 38                                                           PF5 408
   40 JMN=JMIN+1                                                         PF5 409
      JMX=JMAX-2                                                         PF5 410
      AF=1.                                                              PF5 411
      L=JMN                                                              PF5 412
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM24)                              PF5 413
      IF (IWRITE.GT.1) GO TO 42                                          PF5 414
      WRITF (JWPITF,41) NCP                                              PF5 415
   41 FORMAT (///,37HO LINE PASSING THROUGH QUADRILATERAL ,I4)           PF5 416
   42 DO 46 J=JMN,JMX                                                    PF5 417
      IF ((STML(J+2)-STML(L-1)).LT.8.*(STML(J+1)-STML(L))) GO TO 45      PF5 418
      IF (IWRITE.GT.1) GO TO 44                                          PF5 419
      WRITE (JWRITF,43) XL(L),YL(L),ZL(L),XL(J+1),YL(J+1),ZL(J+1)        PF5 420
   43 FORMAT (14H POINT DELETED,10X,3F12.5,10X,3F12.5)                   PF5 421
   44 STML(L)=(AF*STML(L)+STML(J+1))/(AF+1.)                             PF5 422
      XL(L)=(AF*XL(L)+XL(J+1))/(AF+1.)                                   PF5 423
      YL(L)=(AF*YL(L)+YL(J+1))/(AF+1.)                                   PF5 424
      ZL(L)=(AF*ZL(L)+ZL(J+1))/(AF+1.)                                   PF5 425
      UABS(L)=(AF*UABS(L)+UABS(J+1))/(AF+1.0)                            PF5 426
      AF=AF+1.                                                           PF5 427
      IF (IWRITF.LT.0) WRITE (JWRITE,NAM25)                              PF5 428
      IF (L.LE.MID)MID=MID-1                                             PF5 429
      GO TO 46                                                           PF5 430
   45 AF=1.                                                              PF5 431
      L=L+1                                                              PF5 432
      K=J+1                                                              PF5 433
      STML(L)=STML(K)                                                    PF5 434
      XL(L)=XL(K)                                                        PF5 435
      YL(L)=YL(K)                                                        PF5 436
      ZL(L)=ZL(K)                                                        PF5 437
      UABS(L)=UABS(K)                                                    PF5 438
      IF (IWRITF.LT.0) WRITE (JWRITE,NAM26)                              PF5 439
   46 CONTINUE                                                           PF5 440
      L=L+1                                                              PF5 441
      STML(L)=STML(JMAX)                                                 PF5 442
      XL(L)=XL(JMAX)                                                     PF5 443
      YL(L)=YL(JMAX)                                                     PF5 444
      ZL(L)=ZL(JMAX)                                                     PF5 445
      UABS(L)=UABS(JMAX)                                                 PF5 446
```

```
      IF (IWRITE.LT.0) WRITE (JWRITE,NAM27)
      JMAX=L
      IF (IWRITE.GT.1) GO TO 48
      WRITE (JWRITE,47)
   47 FORMAT (3H0 I,6X,1HX,9X,1HY,9X,1HZ,13X,2HCP,8X,2HSL,7X,4HUABS)
   48 K=0
      LOHO=0
      LOGO=0
      IF ((JMAX-JMIN).LE.4)LOGO=1
      DO 51 I=JMIN,JMAX
      K=K+1
      VOV(K)=UABS(I)
      SS(K)=STML(I)
      IF (LOGO.EQ.0) GC TO 49
      SSTEST=1000.0
      IF (K.GT.1)SSTEST=(SS(K)-SS(K-1))
      IF (SSTFST.LF.0.0)LOHO=LOHO+1
   49 IF (IWRITE.GT.1) GO TO 51
C***  WRITE POINTS,VELCCITIES,&CP'S FOR THE STREAMLINE POINTS CALCULATED
      CP=1.0-UABS(I)**2
      WRITE (JWRITE,50) K,XL(I),YL(I),ZL(I),CP,STML(I),UABS(I)
   50 FORMAT (1X,I3,3F10.5,4X,3F10.5)
   51 CONTINUE
      IF (LOHO.GT.0) GC TO 52
C***  CALL ROUTINE TO CALCULATE BOUNDARY LAYER OVER THE STREAMLINE
      CALL BLCONT(K,IWRITE)
      IF (ICANCL.NF.0) RFTURN
      K2=(MID-JMIN)*KKK+1
C***  HOLD THE DISPLACEMENT THICKNESS & THE WALL SHEAR VALUES AT THE
C***  CENTROID POINT FOR WHICH THE STREAMLINE WAS CALCULATED
      SKIN(LL)=CFI(K2)
      DSTAR(LL)=DELS(K2)
      HSK=SKIN(LL)
      HDS=DSTAR(LL)
      GC TO 56
   52 SKIN(LL)=HSK
      DSTAR(LL)=HDS
      WRITE (JWRITE,53) LL
   53 FORMAT (1X,/,2X,65HVALUES OF WALL SHEAR AND DISPLACEMENT THICKNESS
     1 AT QUADRILATERAL ,I3,40H WAS SET DUE TO STREAMLINE DIFFICULTIES.,
     1/)
      GO TO 56
```

```
   54 WRITE (JWRITE,41) NCP
      WRITE (JWRITE,55)
   55 FORMAT (35H PROBABLE ERROR - LINE IS VERY LCNG)
      GC TO 35
   56 CONTINUE
      IWRITE=IWRTS
      IF (II-2)  57,61,59
   57 JEND=NEND-1
C***  ESTIMATE DSTAR & SKIN FOR TRIANGLES AT THE NOSE AND TAIL OF BODY
      DO 58 J=1,JFND,2
      DSTAR(J)=DSTAR(J+1)/3.0
   58 SKIN(J)=SKIN(J+1)/3.0
      GO TO 61
   59 JSTART=NSTART+1
      DO 60 J=JSTART,NLIN,2
      DSTAR(J)=DSTAR(J-1)
   60 SKIN(J)=SKIN(J-1)
   61 CONTINUE
C***  END OF LOOP ON PANEL CENTROIDS EXCEPT TRIANGLES
      WRITE (JWRITE,62)
   62 FORMAT (///,1X,56HSUMMARY OF BOUNDARY LAYER INFORMATION FOR QUADRI
     1LATERALS//2X,5HNQUAD,6X,1HX,9X,1HY,9X,1HZ,11X,5HDSTAR,5X,4HSKIN/)
      CDF=0.0
      AREA=0.0
C***  WRITE POINTS,DSTAR,& SKIN FOR EACH CENTROID. COMPUTE FRICTION CD
      DC 64 J=1,NLIN
      IF (SKIN(J).GT.900.0)SKIN(J)=0.0
      WRITE (JWRITE,63) J,XCP(J),YCP(J),ZCP(J),DSTAR(J),SKIN(J)
   63 FCRMAT (4X,I3,3F10.5,4X,2F10.5)
      CDF=CDF-SKIN(J)*AQP(J)*VX1(J)/(0.5*VABS(J))
      AREA=AREA+AQP(J)
   64 CCNTINUE
      AREAAV=AREA/NLIN
      CDFA=2.0*CDF
      CDF=CDFA/REFA
      WRITE (JWRITE,65) CDF,REFA,REBODY,STOTAL
   65 FCRMAT (///,15X,25HFRICTION DRAG COEFFICIENT,//,13X,29(1H*),/17X,1
     14HFRICTION CD = ,F11.5,/14X,17HREFERENCE AREA = ,F11.5,/13X,18HREY
     1NCLDS NUMBER = ,E11.4/17X,14HBODY LENGTH = ,F11.5/,13X,29(1H*),///
     1)
      REWIND KFILE3
      REWIND KFILE4
```

```
      REWIND KFILE5                                                     PF5 531
      SUMA=0.0                                                          PF5 532
      AVXCG=0.0                                                         PF5 533
      AVAREA=0.0                                                        PF5 534
      AVDELS=0.0                                                        PF5 535
      NDC=NLIN-2*MM+4                                                   PF5 536
C***  CALCULATE AVERAGE DSTAR,AVERAGE PANEL AREA, AND AVERAGE X-ORDINATE PF5 537
C***  FOR THE THIRD STATION OF INPUT POINTS FROM THE END OF BODY.  THESE PF5 538
C***  QUANTITIES ARE USED TO CALCULATE THE WAKE-BODY COORDINATES.        PF5 539
      DO 66 I=NDO,NLIN,2                                                PF5 540
      AVXCG=AVXCG+XCP(I-1)*AQP(I-1)                                     PF5 541
      AVDELS=AVDELS+DSTAR(I-1)*AQP(I-1)                                 PF5 542
      AVAREA=AVAREA+AQP(I-1)                                            PF5 543
   66 SUMA=SUMA+AQP(I)                                                  PF5 544
      AVDELS=AVDELS/AVAREA                                              PF5 545
      AVXCG=AVXCG/AVAREA                                                PF5 546
      SUMA=SUMA/(MM-1)                                                  PF5 547
      AREAAV=SQRT(AREAAV*SUMA)                                          PF5 548
      DO 67 I=1,NLIN                                                    PF5 549
      XNH(I)=XNP(I)                                                     PF5 550
      ZNH(I)=ZNP(I)                                                     PF5 551
   67 AQPH(I)=AQP(I)                                                    PF5 552
      RETURN                                                            PF5 553
   68 WRITE (JWRITE,69) IS,NQ                                           PF5 554
   69 FORMAT (14H TAPE 04 ERROR,2I4)                                    PF5 555
      REWIND KFILE                                                      PF5 556
   70 RETURN                                                            PF5 557
      END                                                               PF5 558

      SUBROUTINF BLCONT(KNP,IWRITE)                                     BLC   1
C                                                                       BLC   2
C***  BOUNDARY LAYER CONTROL ROUTINE WHICH CALLS LAMINR & TURB2. THE    BLC   3
C***  LAMINAR AND TURBULENT ROUTINES WHICH USE MOMENTUM INTEGRAL METHOD BLC   4
C                                                                       BLC   5
      DIMENSION V(150),S(150),DUDS(150)                                 BLC   6
      COMMON /BOUND/TAW(150),HMEAN(150),NP                              BLC   7
      COMMON /SPLN/VCOFF(4,75)                                          BLC   8
      COMMON /BL/VOV(75),SS(75),VINF,VO,ROE,DELS(150),CFI(150),THT(150),BLC   9
     1STOTAL,KKK                                                        BLC  10
      COMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE BLC  11
```

```
      15
       COMMON /CANCEL/ICANCL,NODE
C
       STRAN=.05
       NP=KNP
C***   SPLINE IS CALLED TO CURVE FIT THE STREAMLINE VALUES OF ABS3LUTE
C***   VELOCITY VERSUS SURFACE DISTANCE FROM THE NCSE OF THE BODY
       CALL SPLINE(NP,VOV,SS,0)
       M=NP-1
       SEND=SS(NP)
       KKK=2
       K=1
       S(1)=SS(1)
       V(1)=VOV(1)
       DUDS(1)=VCOEF(3,1)
       DO 1 I=1,M
       H=(SS(I+1)-SS(I))/KKK
       DC 1 J=1,KKK
       K=K+1
       S(K)=S(K-1)+H
       V(K)=((VCOFF(1,I)*S(K)+VCOEF(2,I))*S(K)+VCOEF(3,I))*S(K)+VCOEF(4,I
      1)
       DUDS(K)=(3.0D0*VCOEF(1,I)*S(K)+2.0D0*VCOEF(2,I))*S(K)+VCOEF(3,I)
       NPP=KKK*(NP-1)+1
     1 CONTINUF
       DC 2 I=1,NPP
       S(I)=S(I)/STOTAL
     2 DUDS(I)=DUDS(I)*STOTAL
       CALL LAMINR(S,V,DUDS,STRAN,II,NPP,IWRITE)
       IF (ICANCL.NE.0) RETURN
       IF (II.FQ.NPP) GC TO 3
       CALL TURB2(S,V,DUDS,SEND,II,NPP)
     3 IF (IWRITE.GT.1) GO TO 5
       WRITE (JWRITF,4)
     4 FORMAT (/,6X,1HS,13X,1HV,10X,4HDUDS,9X,5HHMEAN,8X,6HDELTAS,7X,6HTH
      1FTAS,8X,3HTAW,9X,3HCFI/)
     5 DO 6 I=1,NPP
       S(I)=S(I)*STOTAL
       V(I)=V(I)*VINF
       DUDS(I)=DUDS(I)*VINF/STOTAL
     6 DELS(I)=DELS(I)*STOTAL
       IF (IWRITE.GT.1) RETURN
```

```
      DO 7 I=1,NPP,KKK
    7 WRITE (JWRITE,8) S(I),V(I),DUDS(I),HMEAN(I),DELS(I),THT(I),TAW(I),
     1CFI(I)
    8 FORMAT (1X,8(F10.5,3X))
      RETURN
      END


      SUBROUTINE LAMINR(X,UDEL,DUDX,XTRAN,II,NPP,IWRITE)
C
C***  CALCULATES LAMINAR BOUNDARY LAYER BY MOMENTUM INTEGRAL
C
      DIMENSION X(150),UDEL(150),DUDX(150)
      COMMON /BOUND/TAW(150),HMEAN(150),NP
      COMMON /SPLN/VCOEF(4,75)
      COMMON /BL/VOV(75),SS(75),VINF,VO,ROE,DELS(150),CFI(150),THT(150),
     1STOTAL,KKK
      COMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE
     15
      COMMON /CANCEL/ICANCL,NODE
C
      II=1
      H=X(II+1)-X(II)
      IF (UDEL(II).LT.1.0E-06) GO TO 1
C***  STARTING CONDITIONS ASSUMING FLAT PLATE
      Z=0.0
      THT(II)=0.0
      DELS(II)=0.0
      TAW(II)=5000.0
      CFI(II)=1000.0
      HMEAN(II)=2.554
      II=2
      GO TO 4
C***  STARTING CONDITIONS ASSUMING STAGNATION POINT
    1 DUDX0=VCOEF(3,1)*STOTAL
      D2UDXC=2.0*VCOEF(2,1)*STOTAL**2
      Z=0.0770/DUDX0
      THT(II)=SQRT(Z*VO/(VINF*STOTAL))
      DELS(II)=THT(II)*2.308
      CFI(II)=0.664*UDEL(II)/(THT(II)*VINF*STOTAL/VO)
      TAW(II)=CFI(II)*0.5*ROE*VINF*VINF
```

```
      HMEAN(II)=2.308
      II=2
      GO TO 3
    2 II=II+1
      H=X(II)-X(II-1)
      GO TO 4
    3 AM1=H*(-0.0652*D2UDX0/(DUDX0**2))
      GO TO 5
    4 AKD=Z*DUDX(II-1)
      CALL FFK(AKD,FD,F1D,F2D)
      IF (ICANCL.NE.0) RETURN
      AM1=H*FD/UDEL(II-1)
    5 UDELP=0.5*(UDEL(II-1)+UDEL(II))
      DUDXP=0.5*(DUDX(II-1)+DUDX(II))
      AKD=(Z+AM1/2.0)*DUDXP
      CALL FFK(AKD,FD,F1D,F2D)
      IF (ICANCL.NE.0) RETURN
      AM2=H*FD/UDELP
      AKD=(Z+AM2/2.0)*DUDXP
      CALL FFK(AKD,FD,F1D,F2D)
      IF (ICANCL.NE.0) RETURN
      AM3=H*FD/UDELP
      AKD=(Z+AM3)*DUDX(II)
      CALL FFK(AKD,FD,F1D,F2D)
      IF (ICANCL.NE.0) RETURN
      AM4=H*FD/UDEL(II)
      Z=Z+(AM1+2.0*AM2+2.0*AM3+AM4)/6.0
      AKD=Z*DUDX(II)
      CALL FFK(AKD,FD,F1D,F2D)
      IF (ICANCL.NE.0) RETURN
      THT(II)=SQRT(Z*VO/(VINF*STOTAL))
      DELS(II)=THT(II)*F1D
      CFI(II)=2.0*F2D*UDEL(II)/(THT(II)*VINF*STOTAL/VO)
      TAW(II)=CFI(II)*0.5*ROE*VINF*VINF
      HMEAN(II)=DELS(II)/THT(II)
      IF (TAW(II).LT.0.0) GO TO 8
      IF (X(II).GE.XTRAN) GO TO 6
      IF (II.EQ.NPP) GO TO 6
      GO TO 2
    6 IF (IWRITE.GT.1) RETURN
      WRITE (JWRITE,7) X(II),II
    7 FORMAT (/1X,18HTRANSITION AT X = ,F10.5,17H FOR STEP NUMBER ,I3)
```

```
      RETURN                                                            LAM
    8 IF (IWRITE.GT.1) RETURN                                           LAM
      IK=II/KKK                                                         LAM
      WRITE (JWRITE,9) TAW(II),IK                                       LAM
    9 FORMAT (/,27H NEGATIVE VALUE OF TAWST = ,E14.7,21H AT OR AFTER STA LAM
     1TION ,I3,19H ON THE STREAMLINE.,38H FOR MORE INFORMATION ON THE LO LAM
     1CATION,/,108H OF LAMINAR SEPARATION USE IWRITE = 0 PRINT OPTION. T LAM
     1HE TURBULENT ROUTINE IS CALLED AT THE SEPARATION POINT.)          LAM
      RETURN                                                            LAM
      END                                                               LAM


      SUBROUTINE FFK(AKD,FD,F1D,F2D)                                    FFK
C                                                                       FFK
C***  ROUTINE FOR NUMERICAL INTERPOLATION OF LAMINAR B.L. FUNCTIONS      FFK
C                                                                       FFK
      DIMENSION AK(55),F(55),F1(55),F2(55)                              FFK
      COMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE FFK
     15                                                                 FFK
      COMMON /CANCEL/ICANCL,NODE                                        FFK
      DATA AK/0.094815,0.094632,0.094083,0.093166,0.091882,0.090234,0.08 FFK
     18223,0.085855,0.083134,0.080068,0.076664,0.072930,0.068877,0.06451 FFK
     16,0.059857,0.054912,0.049697,0.044223,0.038506,0.032562,0.026405,0 FFK
     1.020054,0.013524,0.006833,0.000000,-0.006957,-0.014321,-0.021170,- FFK
     10.028387,-0.035651,-0.042943,-0.050244,-0.057532,-0.064789,-0.0719 FFK
     195,-0.079129,-0.086171,-0.093104,-0.099906,-0.106558,-0.113043,-0. FFK
     1119341,-0.125434,-0.131304,-0.136935,-0.142309,-0.147411,-0.152224 FFK
     1,-0.156735,-0.160927,-0.164789,-0.168307,-0.171470,-0.174267,-0.17 FFK
     16687/                                                             FFK
      DATA F/-0.094815,-0.093915,-0.091177,-0.086542,-0.079963,-0.071400 FFK
     1,-0.060818,-0.048195,-0.033512,-0.016759,0.002068,0.022964,0.04591 FFK
     19,0.070916,0.097932,0.126938,0.157897,0.190770,0.225538,0.262060,0 FFK
     1.300369,0.340371,0.381999,0.425181,0.469841,0.515896,0.563264,0.61 FFK
     11853,0.661571,0.712321,0.764004,0.816516,0.869752,0.923601,0.97795 FFK
     12,1.032691,1.087700,1.142862,1.198055,1.253157,1.308043,1.362589,1 FFK
     1.416665,1.470145,1.522902,1.574802,1.625719,1.675522,1.724079,1.77 FFK
     11259,1.816936,1.860975,1.903251,1.943633,1.981994/                FFK
      DATA F1/2.250000,2.250686,2.252684,2.255924,2.260349,2.265904,2.27 FFK
     12545,2.280231,2.288928,2.298609,2.309248,2.320828,2.333333,2.34675 FFK
     11,2.361073,2.376291,2.392406,2.409414,2.427320,2.446130,2.465849,2 FFK
     1.486485,2.508057,2.530574,2.554053,2.578518,2.603995,2.630480,2.65 FFK
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
18031,2.686666,2.716418,2.747321,2.779412,2.812734,2.847327,2.88324
12,2.920529,2.959246,2.999447,3.041230,3.084570,3.129634,3.176471,3
1.225166,3.275810,3.328505,3.383357,3.440479,3.499999,3.562056,3.62
16788,3.694360,3.764940,3.838720,3.915905/
 DATA F2/0.355556,0.355293,0.354515,0.353236,0.351470,0.349229,0.34
16528,0.343380,0.339800,0.335801,0.331397,0.326601,0.321428,0.31589
12,0.310005,0.30378[illegible],0.297237,0.290383,0.283234,0.275804,0.268107,0
1.260156,0.251966,0.243549,0.234921,0.226093,0.217082,0.207899,0.19
18559,0.189077,0.179464,0.169736,0.159906,0.149987,0.139995,0.12994
11,0.119841,0.109708,0.099555,0.089397,0.079247,0.069120,0.059028,0
1.048985,0.039007,0.029105,0.019294,0.0[illegible]9588,0.000000,-0.009455,-0.
1018765,-0.027914,-0.036890,-0.045678,-0.054266/
C
  IF (AKD.LE.AK(1)) GO TO 1
  FD=F(1)
  F1D=F1(1)
  F2D=F2(1)
  AKD=AK(1)
  RETURN
1 IF (AKD.GT.AK(55)) GO TO 2
  FD=F(55)
  F1D=F1(55)
  F2D=F2(55)
  AKD=AK(55)
  RETURN
2 DO 3 I=1,54
  IF (AKD.LE.AK(I).AND.AKD.GT.AK(I+1)) GO TO 5
3 CONTINUE
  WRITE (JWRITE,4) AKD
4 FORMAT (1X,6HAKD = ,E12.5,22H EXCEEDS ALLOWED RANGE)
  ICANCL=1
  RETURN
5 FD=F(I)+(F(I+1)-F(I))*(AKD-AK(I))/(AK(I+1)-AK(I))
  F1D=F1(I)+(F1(I+1)-F1(I))*(AKD-AK(I))/(AK(I+1)-AK(I))
  F2D=F2(I)+(F2(I+1)-F2(I))*(AKD-AK(I))/(AK(I+1)-AK(I))
  RETURN
  END

  SUBROUTINE TURB2(SUMS,UE,DUEDX,SEND,N1,NPP)
C
```

```
C***    ROUTINE FOR TURBULENT BOUNDARY LAYER USING MCMENTUM  INTEGRAL
C
        COMMON /BOUND/TAW(150),HMEAN(150),NP
        COMMON /BL/VOV(75),SS(75),VINF,VO,ROE,DELS(150),CFI(150),THT(150),
       1STOTAL,KKK
        DIMENSION HVMEAN(2),SUMS(150),UE(150),DUEDX(150)
C
        DC 1 L=N1,NPP
        DUEDX(L)=VINF*DUEDX(L)/STOTAL
        SUMS(L)=SUMS(L)*STOTAL
      1 UE(L)=VINF*UE(L)
        THT(N1)=THT(N1)*STOTAL
        DELS(N1)=1.45*THT(N1)
        C1=0.56
        C2=.1667
        C3=1.65
        C4=0.246
        C5=0.678
        C6=0.268
        HMEAN(N1)=1.45
        HVMEAN(1)=1.269*HMEAN(N1)/(HMEAN(N1)-.379)
        F1=3.+2.*C2
        E2=1.+C2
        E3=3.+3.*C2
        T1=0.02*C1/C3*E2
        IBEG=N1+1
        DO 7 I=IBEG,NPP
        HMEAN(I)=0.0
        HVMEAN(2)=0.0
        UETR=(UE(I-1)/UE(I))**E3
        UEINT=0.5*(SUMS(I)-SUMS(I-1))*(UE(I)**E1+UE(I-1)**E1)
        THT(I)=(THT(I-1)**F2*UETR+T1*VO**C2/(UE(I)**E3)*UEINT)**(1./E2)
        TERMA=(0.02*C1)/((UE(I)*THT(I))/VO)**C2*1.1
        T2=THT(I)/UE(I)*DUEDX(I)
        HH=HMEAN(I-1)
        HV=HVMEAN(1)
        DO 6 J=1,6
        TERMB=-HV*C4*10.0**(-C5*HH)*(UE(I)*THT(I)/VO)**(-C6)
        TERMB=TERMB/2.
        TERMC=(HH-1.1)*HV*T2
        DHVDX=(TERMA+TERMB+TERMC)/THT(I-1)
        HVITR=HVMEAN(1)+DHVDX*(SUMS(I)-SUMS(I-1))
```

```
      IF (HVITR-1.85)  3,2,2
    2 HVITR=1.85
    3 IF (HVITR-1.55)  4,4,5
    4 HVITR=1.55
    5 HITR=0.379*HVITR/(HVITR-1.269)
      HVMEAN(2)=HVMEAN(2)+HVITR
      HMEAN(I)=HMEAN(I)+HITR
      HV=HVITR
    6 HH=HITR
      HVMEAN(1)=HVMEAN(2)/6
      HMEAN(I)=HMEAN(I)/6
      DELS(I)=THT(I)*HMEAN(I)
      CFI(I)=C4*10.0**(-C5*HMEAN(I))*(UE(I)*THT(I)/VO)**(-C6)*(UE(I)/VIN
     1F)**2
      TAW(I)=CFI(I)*0.5*ROE*VINF*VINF
      IF (SEND-SUMS(I))  8,8,7
    7 CONTINUE
    8 DO 9 L=N1,NPP
      DELS(L)=DELS(L)/STOTAL
      THT(L)=THT(L)/STOTAL
      DUEDX(L)=DUEDX(L)*STOTAL/VINF
      SUMS(L)=SUMS(L)/STOTAL
    9 UE(L)=UE(L)/VINF
      RETURN
      END


      SUBROUTINE RGVRTX(XC,YC,ZC,X0,Y0,Z0,RD,NTHETA,THETA,PHI,NVPHI,NRS,
     1NSRV,VX,VY,VZ,IM,NPTS)
C
C***  SUBROUTINE RGVRTX CALCULATES THE INDUCED VELOCITIES DUE TO THE
C***  RING VORTICES
C
      DIMENSION XC(40),YC(40),ZC(40),RD(40),NTHETA(40),THETA(40),PHI(40)
     1,NVPHI(40),NRS(40),UC(360),VC(360),WC(360),TX(360),NPTS(40)
C
C***  DEFINE INITIAL PARAMETERS
      PI=3.1415927
      C=1.0/(4.0*PI)
      VXX=0.0
      VYY=0.0
```

```
      VZZ=0.0
C***  SET UP LOOP OVER SYSTEMS OF VORTICES
      DO 8 L=1,NSRV
      TINC=2.0*PI/FLOAT(NTHETA(L)-1)
      NR=NRS(L)
      ISEE=0
      IF (NPTS(L).EQ.0.AND.NRS(L).EQ.1)ISEE=1
      DO 7 M=1,NR
C***  DETERMINE RADIAL STATION AND CIRCULATION
      IF (ISEE.EQ.0) GO TO 1
      GAMMA=CIRC(L)
      R=RD(L)
      DR=R
      GO TO 2
    1 R=F(NR,M,RD(L),DR)
      XR=R/RD(L)
      GAMMA=G(L,XR)
    2 T=THETA(L)
      PHX=PHI(L)
      NTH=NTHETA(L)
C***  TEST FOR FLOW SIMPLIFICATION BY DOUBLET
      TEST=SQRT((X0-XC(L))**2+(Y0-YC(L))**2+(Z0-ZC(L))**2)
      FACTOR=1.5
      DISX=FACTOR*(2.0*RD(L))
      IF (TEST.LT.DISX) GO TO 3
      FACX=0.00932
C***  CALCULATE DOUBLET STRENGTH AND FLOW
      GAM=FACX*GAMMA*R**2
      CALL DBLET(NVPHI,R,T,PHX,GAM,L,X0,Y0,Z0,XC,YC,ZC,VX,VY,VZ)
      IN=1
      GO TO 6
    3 DO 5 I=1,NTH
C***  DEFINE PARAMETRIC EQNS FOR POINT ON VORTEX RING
      XI=XC(L)+R*SIN(T)*COS(PHX)
      ETA=YC(L)+R*SIN(T)*SIN(PHX)
      ZETA=ZC(L)+R*COS(T)
C***  DEFINE DERIVATIVES OF PARAMETRIC EQNS WRT T
      DXI=R*COS(T)*COS(PHX)
      DETA=R*COS(T)*SIN(PHX)
      DZETA=-R*SIN(T)
C***  INITIALIZE OR DEFINE PARAMETER
      TX(I)=T
```

```
      UC(I)=0.0
      VC(I)=0.0
      WC(I)=0.0
C***  COMPUTE DISTANCE FROM FILAMENT TO ARBITRARY POINT
      RHO=SQRT((XI-X0)**2+(ETA-Y0)**2+(ZETA-Z0)**2)
      IF (RHO.LE.1.0E-10) GO TO 4
C***  COMPUTE VELOCITY CONTRIBUTIONS AROUND THE RING VORTEX UPON AN
C***  ARBITRARY POINT
      DENOM=RHO**3*RD(L)/(C*DR)
      UC(I)=(DETA*(Z0-ZETA)-DZETA*(Y0-ETA))/DENOM*GAMMA
      VC(I)=(DZETA*(X0-XI)-DXI*(Z0-ZETA))/DENOM*GAMMA
      WC(I)=(DXI*(Y0-ETA)-DETA*(X0-XI))/DENOM*GAMMA
    4 IF (I.EQ.NTH) GO TO 5
      T=T+TINC
      IF (NVPHI(L).NE.0)PHX=PHX+TINC
    5 CONTINUE
C***  INTEGRATE TO FIND VX, VY, VZ
      IM=0
      CALL TRAP(NTH,TX,UC,VX)
      CALL TRAP(NTH,TX,VC,VY)
      CALL TRAP(NTH,TX,WC,VZ)
    6 VXX=VXX+VX
      VYY=VYY+VY
      VZZ=VZZ+VZ
    7 CONTINUE
    8 CONTINUE
C***  DEFINE TOTAL INDUCED VELOCITY COMPONENTS
      VX=VXX
      VY=VYY
      VZ=VZZ
C***  MODIFY, IF NECESSARY, FOR PRECISION PROBLEMS
      VMX=ABS(VX)
      IF (ABS(VY).GT.VMX)VMX=ABS(VY)
      IF (ABS(VZ).GT.VMX)VMX=ABS(VZ)
      FVMX=1.0E-4*VMX
      IF (ABS(VX).LT.FVMX)VX=0.0
      IF (ABS(VY).LT.FVMX)VY=0.0
      IF (ABS(VZ).LT.FVMX)VZ=0.0
      RETURN
      END
```

```
      FUNCTION F(N,M,R,RD)
C
C***  FUNCTION F CALCULATES THE RADIAL STATION OF THE VORTEX RING
C
      RD=R/FLOAT(N)
      RX=R+RD
      DO 1 I=1,M
    1 RX=RX-RD
      F=RX
      RETURN
      END


      FUNCTION G(L,R)
C
C***  FUNCTION G COMPUTES THE CIRCULATION AS A FUNCTION OF RADIAL
C***  DISTANCE FROM THE VORTEX'S CENTER
C
      CALL GAMMA(L,R,B)
      G=B
      RETURN
      END


      FUNCTION CIRC(L)
C
C***  FUNCTION CIRC SETS THE VORTEX STRENGTH
C
      COMMON /S2V/CIRCV(40)
C
      CIRC=CIRCV(L)
      RETURN
      END


      SUBROUTINE SPLINE(N,Y,X,IC)
C
C***  THIS SUBROUTINE PERFORMS A SPLINE FIT ON THE TABULATED
```

```
C***     DATA Y VS. X.  THE SPLINE FIT PROVIDES A CURVE FIT OF THE
C***     TABULATED DATA THAT HAS A CONTINUOUS FIRST DERIVATIVE.  THE
C***     FORM OF THE CURVE FIT IS
C***        Y = AA(1,I)*X**3 + AA(2,I)*X**2
C***                             + AA(3,I)*X + AA(4,I)
C***     FOR          X(I) <= X <= X(I+1).
C***        N      - NUMBER OF TABULATED DATA POINTS
C***        Y      - TABULATED FUNCTION VALUES
C***        X      - TABULATED ARGUMENT VALUES
C***        AA     - ARRAY OF COEFFICIENTS OF THE CUBIC POLYNOMIAL
C***                 SPLINE FIT.  DIMENSIONS OF AA ARE 4 BY N-1.
C***     REFERENCE FOR THIS METHOD IS
C***        THE THEORY OF SPLINES AND THEIR APPLICATIONS
C***        J.H. AHLBERG, ET AL. ACADEMIC PRESS, NEW YORK , 1967
C
      DIMENSION H(75),Q(75),U(75),X(N),Y(N),CF(40,4,50)
      COMMON /SPLN/AA(4,75)
C
      NM1=N-1
      DO 1 I=1,NM1
    1 H(I)=X(I+1)-X(I)
C***  IF IC=0, SET SECOND DERIVATIVE TO ZERO FOR LEFT HAND END CONDITION
      Q(1)=0.0
      U(1)=0.0
      IF (IC.EQ.0) GO TO 2
C***  MODIFIED LEFT HAND END CONDITION THAT ALLEVIATES THE NEED
C***  TO SPECIFY THE X-DERIVATIVE OF Y AT POINT 1
      Q(1)=-31.0/32.0
      H1=H(1)
      H2=H(2)
      H3=H(3)
      U(1)=Y(1)*(32.0*H1+42.0*H2+21.0*H3)/(H1+H2)/(H1+H2+H3)-Y(2)*(11.0*
     1H1+42.0*H2+21.0*H3)/(H2+H3)/H2+Y(3)*H1*(11.0*H1+21.0*(H2+H3))/(H1+
     1H2)/H2/H3-Y(4)*H1*(11.0*H1+21.0*H2)/(H2+H3)/(H1+H2+H3)/H3
      U(1)=3.0*U(1)/H1/16.0
C***  GENERATE INTERNAL U(I) BY ALGORITHMN GIVEN BY AHLBERG
    2 HM=H(1)
      YY=Y(2)
      YM=Y(1)
      DO 3 I=2,NM1
      HP=H(I)
      YP=Y(I+1)
```

```
      D=3.0*((YP-YY)/HH-(YY-YM)/HM)/(HH+HM)
      C=0.5*HH/(HH+HM)
      A=0.5-C
      P=A*Q(I-1)+1.0
      Q(I)=-C/P
      U(I)=(D-A*U(I-1))/P
      HM=HH
      YM=YY
    3 YY=YP
C***  MODIFIED RIGHT HAND END CONDITION THAT ALLEVIATES THE NEED
C***  TO SPECIFY THE X-DERIVATIVE OF Y AT POINT N
      A=31.0/32.0
      P=A*Q(N-1)+1.0
      H1=H(NM1)
      H2=H(NM1-1)
      H3=H(NM1-2)
      D=Y(N)*(32.0*H1+42.0*H2+21.0*H3)/(H1+H2)/(H1+H2+H3)-Y(NM1)*(11.0*H
     11+42.0*H2+21.0*H3)/(H2+H3)/H2+Y(NM1-1)*H1*(11.0*H1+21.0*(H2+H3))/(
     1H1+H2)/H2/H3-Y(NM1-2)*H1*(11.0*H1+21.0*H2)/(H2+H3)/(H1+H2+H3)/H3
      D=3.0*D/H1/16.0
      U(N)=(D-A*U(N-1))/P
C***  SOLVE FOR THE SPLINE COEFFICIENTS CORRESPONDING TO AHLBERG'S
C***  M(0) TO M(N) AND STORE THEM IN THE U(I).
      DO 4 J=1,NM1
      I=N-J
    4 U(I)=Q(I)*U(I+1)+U(I)
C***  FORM THE AA(J,I) COEFFICIENTS FOR THE CONVENTIONAL FORM OF
C***  A CUBIC POLYNOMIAL FROM THE U(I)
      UU=U(1)
      XX=X(1)
      YY=Y(1)
      DO 5 I=1,NM1
      UP=U(I+1)
      XP=X(I+1)
      YP=Y(I+1)
      HH=H(I)
      AA(1,I)=(UP-UU)/HH/6.0
      AA(2,I)=0.5*(XP*UU-XX*UP)/HH
      AA(3,I)=0.5*(UP*XX*XX-UU*XP*XP)/HH+(UU-UP)*HH/6.0+(YP-YY)/HH
      AA(4,I)=(UU*XP*XP*XP-UP*XX*XX*XX)/HH/6.0+(UP*XX-UU*XP)*HH/6.0+(YY*
     1XP-YP*XX)/HH
      XX=XP
```

```
      UU=UP
    5 YY=YP
      IF (IC.EQ.0) GO TC 7
      DO 6 I=1,NM1
      DO 6 J=1,4
    6 CF(IC,J,I)=AA(J,I)
    7 RETURN
C
      ENTRY GAMMA(IC,T,Z)
      IF (T.GT.X(1)) GO TO 8
      I=1
      GO TO 10
    8 DO 9 I=1,NM1
      IF (X(I).LE.T.AND.X(I+1).GT.T) GO TO 10
    9 CCNTINUE
      I=NM1
   10 Z=((CF(IC,1,I)*T+CF(IC,2,I))*T+CF(IC,3,I))*T+CF(IC,4,I)
      RETURN
      END


      SUBROUTINE TRAP(ND,X,DY,Y)
C
C***  SUBROUTINF TRAP PERFORMS TRAPEZOIDAL INTEGRATION
C
C
      DIMENSION X(ND),DY(ND)
C
C***  INITIALIZE PARAMETER
      S2=0.0
      IF (ND-1) 4,3,1
C***  INTEGRATE OVER INTERVAL AND SUM
    1 DO 2 I=2,ND
      S1=S2
      S2=S2+0.5*(X(I)-X(I-1))*(DY(I)+DY(I-1))
    2 CCNTINUE
    3 Y=S2
    4 RETURN
      END
```

```
      SUBROUTINE DBLET(N,R,T,PHX,GAM,L,X0,Y0,Z0,XC,YC,ZC,VX,VY,VZ)
C
C***  SUBROUTINE DBLET CALCULATES DOUBLET FLOW
C
      DIMENSION N(40),XC(40),YC(40),ZC(40)
C
      PI=3.1415927
C***  DETERMINE DOUBLET AXIS VECTOR
      X1=R*SIN(T)*COS(PHX)
      Y1=R*SIN(T)*SIN(PHX)
      Z1=R*COS(T)
      PI2=0.5*PI
      X2=R*SIN(T+PI2)*COS(PHX+N(L)*PI2)
      Y2=R*SIN(T+PI2)*SIN(PHX+N(L)*PI2)
      Z2=R*COS(T+PI2)
      X3=Y1*Z2-Z1*Y2
      Y3=Z1*X2-X1*Z2
      Z3=X1*Y2-Y1*X2
      RD=SQRT(X3**2+Y3**2+Z3**2)
      X3=X3/RD
      Y3=Y3/RD
      Z3=Z3/RD
C***  DETERMINE DOUBLET STRENGTH VECTOR COMPONENTS
      XMU=GAM*X3
      YMU=GAM*Y3
      ZMU=GAM*Z3
      PI4=-0.5*PI2
C***  DFTFRMINF DOUBLFT-TO-ARBITRARY PCINT VFCTOR
      X1=X0-XC(L)
      Y1=Y0-YC(L)
      Z1=Z0-ZC(L)
C***  DETERMINE VELOCITY COMPONENTS
      RD=X1**2+Y1**2+Z1**2
      F=PI4/RD**1.5
      VX=F*XMU*(1.0-3.0*X1**2/RD)
      VY=F*YMU*(1.0-3.0*Y1**2/RD)
      VZ=F*ZMU*(1.0-3.0*Z1**2/RD)
      RETURN
      END
```

```
      SUBROUTINE SPACE(NSRV,SVV,XCV,NMIN,NMAX,X,MMAX,XCRP,AEXP,XRM,ISPAC
     1E,RXMR,XMR)
C
C***  SUBROUTINE SPACE CALCULATES VORTEX LOCATION VIA EXPONENTIAL
C***  FUNCTION
C
      DIMENSION SVV(31,2),XCV(40),X(650),XCRP(2),AEXP(2)
C
      MAX=(NMAX-1)*(MMAX-1)
      NMAXM1=NMAX-1
      JV=1
C***  SET MAXIMUM RADIUS
      RMX=RXMR
    1 JV=JV+1
C***  GUESS A LOCATICN
      J=1
      IF (XCV(JV-1).LT.XMR)J=2
      XCV(JV)=XCV(JV-1)-XCRP(J)
      LPT=1
C***  SCAN N-STATIONS
      DO 2 JN=NMIN,NMAXM1
      LC1=JN
      LC2=JN+1
      IF (XCV(JV).LE.X(LPT).AND.XCV(JV).GE.X(LPT+MMAX)) GO TO 3
    2 LPT=LPT+MMAX
      IF (ISPACE.EQ.2) GO TO 4
      JV=JV-1
      GC TO 5
C***  COMPUTE EXPONENTIAL LOCATION
    3 RXTL=(X(LPT)-XCV(JV))/(X(LPT)-X(LPT+MMAX))
      R=SVV(LC1,2)+RXTL*(SVV(LC2,2)-SVV(LC1,2))
      XCV(JV)=XCV(JV-1)-XCRP(J)*EXP(-AEXP(J)*(R-RMX)/RMX)
      IF (XCV(JV).LE.X(MAX).AND.ISPACE.EQ.1.OR.JV.EQ.40) GO TO 5
      GO TO 1
    4 F=0.0
      XCV(JV)=XCV(JV-1)-XCRP(J)*EXP(-AEXP(J)*(R-RMX)/RMX)
      IF (XCV(JV).GT.XRM.AND.JV.LT.40) GO TO 1
    5 NSRV=JV
      RETURN
      END
```

# SAMPLE INPUT - FLOWBODY

```
  13
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 550 PANELS -- FUSELAGE ONLY
              160.00E0            0.00016E0            0.002378E0           155.0E0
            .18000E+05          .252525E-06           .110000E+06           .240E0
              520.0E0                0.50E0     2     0      0       0
1.0             0      0
560
   1      2  3.5        4.0       -0.3       -0.05       -18.0
12.8       0.00       -.4271      2.5        0.0         90.0        360     0    4    21
0.16
0.0        0.0                              CIRCULATION DISTRIBUTION
0.05       0.5
0.1        1.0
0.15       1.5
0.2        2.0
0.25       2.35
0.3        2.64
0.35       2.86
0.4        3.0
0.45       3.05
0.5        3.0
0.55       2.6
0.6        2.25
0.65       1.95
0.7        1.55
0.75       1.2
0.8        1.0
0.85       0.65
0.9        0.45
0.95       0.25
1.0        0.001
  12.2085000   0.0000000   -0.4271002    1    1    1
  12.2085000   0.0000000   -0.4271002    1    2    1
  12.2085000   0.0000000   -0.4271002    1    3    1
  12.2085000   0.0000000   -0.4271002    1    4    1
  12.2085000   0.0000000   -0.4271002    1    5    1
  12.2085000   0.0000000   -0.4271002    1    6    1
```

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 7 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 8 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 9 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 10 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 11 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 12 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 13 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 14 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 15 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 16 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 17 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 18 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 19 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 20 | 1 |
| 12.2085000 | 0.0000000 | -0.4271002 | 1 | 21 | 1 |
| 11.5418000 | 0.0000000 | -1.6353990 | 2 | 1 | 1 |
| 11.5418000 | 0.1833000 | -1.5978990 | 2 | 2 | 1 |
| 11.5418000 | 0.3583000 | -1.5270990 | 2 | 3 | 1 |
| 11.5418000 | 0.5146000 | -1.4430990 | 2 | 4 | 1 |
| 11.5418000 | 0.6708000 | -1.3499990 | 2 | 5 | 1 |
| 11.5418000 | 0.8187000 | -1.2520990 | 2 | 6 | 1 |
| 11.5418000 | 0.9688000 | -1.1270990 | 2 | 7 | 1 |
| 11.5418000 | 1.1083000 | -0.9958000 | 2 | 8 | 1 |
| 11.5418000 | 1.2645990 | -0.8436995 | 2 | 9 | 1 |
| 11.5418000 | 1.4083000 | -0.6520995 | 2 | 10 | 1 |
| 11.5418000 | 1.5166990 | -0.4271002 | 2 | 11 | 1 |
| 11.5418000 | 1.5707990 | -0.1854000 | 2 | 12 | 1 |
| 11.5418000 | 1.5625000 | 0.0728998 | 2 | 13 | 1 |
| 11.5418000 | 1.4917000 | 0.3228999 | 2 | 14 | 1 |
| 11.5418000 | 1.3291990 | 0.5229006 | 2 | 15 | 1 |
| 11.5418000 | 1.1062000 | 0.6729002 | 2 | 16 | 1 |
| 11.5418000 | 0.8833000 | 0.7396602 | 2 | 17 | 1 |
| 11.5418000 | 0.6667000 | 0.7708998 | 2 | 18 | 1 |
| 11.5418000 | 0.4437000 | 0.7771006 | 2 | 19 | 1 |
| 11.5418000 | 0.2167000 | 0.7813005 | 2 | 20 | 1 |
| 11.5418000 | 0.0000000 | 0.7813005 | 2 | 21 | 1 |
| 10.3755000 | 0.0000000 | -1.9271000 | 3 | 1 | 1 |
| 10.3755000 | 0.2208000 | -1.9229000 | 3 | 2 | 1 |
| 10.3755000 | 0.4458000 | -1.8853990 | 3 | 3 | 1 |
| 10.3755000 | 0.6625000 | -1.8186990 | 3 | 4 | 1 |
| 10.3755000 | 0.8750000 | -1.7103990 | 3 | 5 | 1 |
| 10.3755000 | 1.0542000 | -1.5728990 | 3 | 6 | 1 |

C-2

| | | | | | |
|---|---|---|---|---|---|
| 10.8755000 | 1.2208000 | -1.3957990 | 3 | 7 | 1 |
| 10.8755000 | 1.3604000 | -1.2061990 | 3 | 8 | 1 |
| 10.8755000 | 1.4854000 | -0.9958000 | 3 | 9 | 1 |
| 10.8755000 | 1.5791990 | -0.7603998 | 3 | 10 | 1 |
| 10.8755000 | 1.6457990 | -0.5103998 | 3 | 11 | 1 |
| 10.8755000 | 1.6667000 | -0.2521000 | 3 | 12 | 1 |
| 10.8755000 | 1.6562990 | 0.0187998 | 3 | 13 | 1 |
| 10.8755000 | 1.6042000 | 0.2979002 | 3 | 14 | 1 |
| 10.8755000 | 1.4792000 | 0.5521002 | 3 | 15 | 1 |
| 10.8755000 | 1.2561993 | 0.7396002 | 3 | 16 | 1 |
| 10.8755000 | 1.0207990 | 0.8354006 | 3 | 17 | 1 |
| 10.8755000 | 0.7708000 | 0.8646002 | 3 | 18 | 1 |
| 10.8755000 | 0.5146000 | 0.8853998 | 3 | 19 | 1 |
| 10.8755000 | 0.2500000 | 0.8979006 | 3 | 20 | 1 |
| 10.8755000 | 0.0000000 | 0.9063005 | 3 | 21 | 1 |
| 10.2085000 | 0.0000000 | -2.1146000 | 4 | 1 | 1 |
| 10.2085000 | 0.2437000 | -2.1146000 | 4 | 2 | 1 |
| 10.2085000 | 0.5000000 | -2.0936990 | 4 | 3 | 1 |
| 10.2085000 | 0.7646000 | -2.0562000 | 4 | 4 | 1 |
| 10.2085000 | 1.0375000 | -1.9812000 | 4 | 5 | 1 |
| 10.2085000 | 1.2749990 | -1.8374990 | 4 | 6 | 1 |
| 10.2085000 | 1.4792000 | -1.6291990 | 4 | 7 | 1 |
| 10.2085000 | 1.6083000 | -1.3750000 | 4 | 8 | 1 |
| 10.2085000 | 1.6833000 | -1.1041000 | 4 | 9 | 1 |
| 10.2085000 | 1.7124990 | -0.8228998 | 4 | 10 | 1 |
| 10.2085000 | 1.7292000 | -0.5521002 | 4 | 11 | 1 |
| 10.2085000 | 1.7292000 | -0.2811995 | 4 | 12 | 1 |
| 10.2085000 | 1.7229000 | 0.0000000 | 4 | 13 | 1 |
| 10.2085000 | 1.6750000 | 0.2917004 | 4 | 14 | 1 |
| 10.2085000 | 1.5479000 | 0.5533000 | 4 | 15 | 1 |
| 10.2085000 | 1.3228990 | 0.7603998 | 4 | 16 | 1 |
| 10.2085000 | 1.0874990 | 0.8813000 | 4 | 17 | 1 |
| 10.2085000 | 0.8292000 | 0.9313002 | 4 | 18 | 1 |
| 10.2085000 | 0.5583000 | 0.9688005 | 4 | 19 | 1 |
| 10.2085000 | 0.2750000 | 0.9979000 | 4 | 20 | 1 |
| 10.2085000 | 0.0000000 | 1.0103990 | 4 | 21 | 1 |
| 9.4795000 | 0.0000000 | -2.2396000 | 5 | 1 | 1 |
| 9.4795000 | 0.2583000 | -2.2354000 | 5 | 2 | 1 |
| 9.4795000 | 0.5333000 | -2.2291000 | 5 | 3 | 1 |
| 9.4795000 | 0.8333000 | -2.2186990 | 5 | 4 | 1 |
| 9.4795000 | 1.1708000 | -2.1875000 | 5 | 5 | 1 |
| 9.4795000 | 1.4750000 | -2.0562000 | 5 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 9.4795030 | 1.6729000 | -1.7916000 | 5 | 7 | 1 |
| 9.4795030 | 1.7624990 | -1.4770990 | 5 | 8 | 1 |
| 9.4795030 | 1.7749990 | -1.1499990 | 5 | 9 | 1 |
| 9.4795030 | 1.7770990 | -0.8562002 | 5 | 10 | 1 |
| 9.4795030 | 1.7812990 | -0.5728998 | 5 | 11 | 1 |
| 9.4795030 | 1.7812990 | -0.2958002 | 5 | 12 | 1 |
| 9.4795030 | 1.7770990 | -0.0042000 | 5 | 13 | 1 |
| 9.4795030 | 1.7375000 | 0.2979002 | 5 | 14 | 1 |
| 9.4795030 | 1.6187000 | 0.5854006 | 5 | 15 | 1 |
| 9.4795030 | 1.3916990 | 0.8083000 | 5 | 16 | 1 |
| 9.4795030 | 1.1457990 | 0.9438000 | 5 | 17 | 1 |
| 9.4795030 | 0.8854000 | 1.0062990 | 5 | 18 | 1 |
| 9.4795030 | 0.5938000 | 1.0417000 | 5 | 19 | 1 |
| 9.4795030 | 0.2917000 | 1.0625000 | 5 | 20 | 1 |
| 9.4795030 | 0.0000000 | 1.0938000 | 5 | 21 | 1 |
| 8.8755030 | 0.0000000 | -2.2811990 | 6 | 1 | 1 |
| 8.8755030 | 0.2708000 | -2.2811990 | 6 | 2 | 1 |
| 8.8755030 | 0.5521000 | -2.2770990 | 6 | 3 | 1 |
| 8.8755030 | 0.8583000 | -2.2666990 | 6 | 4 | 1 |
| 8.3755030 | 1.2145990 | -2.2458000 | 6 | 5 | 1 |
| 8.3755030 | 1.5312990 | -2.1167000 | 6 | 6 | 1 |
| 8.3755030 | 1.7458000 | -1.8436990 | 6 | 7 | 1 |
| 8.8755030 | 1.8166990 | -1.5041990 | 6 | 8 | 1 |
| 8.3755030 | 1.8228990 | -1.1666000 | 6 | 9 | 1 |
| 8.8755030 | 1.8228990 | -0.8604002 | 6 | 10 | 1 |
| 8.8755030 | 1.8228990 | -0.5728998 | 6 | 11 | 1 |
| 8.9755030 | 1.8228990 | -0.2895994 | 6 | 12 | 1 |
| 8.8755030 | 1.8228990 | 0.0103998 | 6 | 13 | 1 |
| 8.8755030 | 1.8021000 | 0.3333998 | 6 | 14 | 1 |
| 8.8755030 | 1.6759000 | 0.6271000 | 6 | 15 | 1 |
| 8.8755030 | 1.4541990 | 0.8688002 | 6 | 16 | 1 |
| 8.3755030 | 1.1978990 | 1.0103990 | 6 | 17 | 1 |
| 8.3755030 | 0.9208000 | 1.0728990 | 6 | 18 | 1 |
| 8.8755030 | 0.6167000 | 1.1[illegible] | 6 | 19 | 1 |
| 8.3755030 | 0.3042000 | 1.1238000 | 6 | 20 | 1 |
| 8.3755030 | 0.0000000 | 1.1353990 | 6 | 21 | 1 |
| 8.3125030 | 0.0000000 | -2.3228990 | 7 | 1 | 1 |
| 8.3125030 | 0.3062000 | -2.3228990 | 7 | 2 | 1 |
| 8.3125030 | 0.6250000 | -2.2979000 | 7 | 3 | 1 |
| 8.3125030 | 0.9521000 | -2.2458000 | 7 | 4 | 1 |
| 8.3125030 | 1.2958000 | -2.1499990 | 7 | 5 | 1 |
| 8.3125030 | 1.5916992 | -1.9080990 | 7 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 8.3125030 | 1.7875000 | -1.6666000 | 7 | 7 | 1 |
| 8.3125030 | 1.8708000 | -1.3228990 | 7 | 8 | 1 |
| 8.3125030 | 1.8750000 | -0.9791002 | 7 | 9 | 1 |
| 8.3125030 | 1.8750000 | -0.6624994 | 7 | 10 | 1 |
| 8.3125030 | 1.8750000 | -0.3646002 | 7 | 11 | 1 |
| 8.3125030 | 1.8750000 | -0.0728998 | 7 | 12 | 1 |
| 8.3125030 | 1.8750000 | 0.2354002 | 7 | 13 | 1 |
| 8.3125030 | 1.8708000 | 0.5770998 | 7 | 14 | 1 |
| 8.3125030 | 1.8542000 | 0.9604006 | 7 | 15 | 1 |
| 8.3125030 | 1.6978990 | 1.3187990 | 7 | 16 | 1 |
| 8.3125030 | 1.3999990 | 1.4833000 | 7 | 17 | 1 |
| 8.3125030 | 1.0563000 | 1.5313000 | 7 | 18 | 1 |
| 8.3125030 | 0.7125000 | 1.5687990 | 7 | 19 | 1 |
| 8.3125030 | 0.3479000 | 1.5654000 | 7 | 20 | 1 |
| 8.3125030 | 0.0000000 | 1.5938000 | 7 | 21 | 1 |
| 7.7085030 | 0.0000000 | -2.3646000 | 8 | 1 | 1 |
| 7.7085030 | 0.3375000 | -2.3646000 | 8 | 2 | 1 |
| 7.7085030 | 0.6917000 | -2.3395990 | 8 | 3 | 1 |
| 7.7085030 | 1.0500000 | -2.2770990 | 8 | 4 | 1 |
| 7.7085030 | 1.3936990 | -2.1187000 | 8 | 5 | 1 |
| 7.7085030 | 1.6499990 | -1.8604000 | 8 | 6 | 1 |
| 7.7085030 | 1.8332990 | -1.5311990 | 8 | 7 | 1 |
| 7.7085030 | 1.9583000 | -1.1771000 | 8 | 8 | 1 |
| 7.7085030 | 1.9167000 | -0.8228998 | 8 | 9 | 1 |
| 7.7085030 | 1.9167000 | -0.5021002 | 8 | 10 | 1 |
| 7.7085030 | 1.9167000 | -0.1978998 | 8 | 11 | 1 |
| 7.7085030 | 1.9167000 | 0.1021004 | 8 | 12 | 1 |
| 7.7085030 | 1.9167000 | 0.4167004 | 8 | 13 | 1 |
| 7.7085030 | 1.9167000 | 0.7645998 | 8 | 14 | 1 |
| 7.7085030 | 1.9062990 | 1.1667000 | 8 | 15 | 1 |
| 7.7085030 | 1.8291990 | 1.6188000 | 8 | 16 | 1 |
| 7.7085030 | 1.5479000 | 1.8479000 | 8 | 17 | 1 |
| 7.7085030 | 1.1833000 | 1.9167000 | 8 | 18 | 1 |
| 7.7085030 | 0.7917000 | 1.9541990 | 8 | 19 | 1 |
| 7.7085030 | 0.3854000 | 1.9688000 | 8 | 20 | 1 |
| 7.7085030 | 0.0000000 | 1.9688000 | 8 | 21 | 1 |
| 7.1252040 | 0.0000000 | -2.4061990 | 9 | 1 | 1 |
| 7.1252040 | 0.3583000 | -2.3957990 | 9 | 2 | 1 |
| 7.1252040 | 0.7375000 | -2.3750000 | 9 | 3 | 1 |
| 7.1252040 | 1.1187000 | -2.3021000 | 9 | 4 | 1 |
| 7.1252040 | 1.4541990 | -2.3999990 | 9 | 5 | 1 |
| 7.1252040 | 1.6978990 | -1.8062000 | 9 | 6 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 7.1252040 | 1.8667000 | -1.4520990 | 9 | 7 | 1 |
| 7.1252040 | 1.9333000 | -1.0832990 | 9 | 8 | 1 |
| 7.1252040 | 1.9375000 | -0.7291002 | 9 | 9 | 1 |
| 7.1252040 | 1.9375000 | -0.4020996 | 9 | 10 | 1 |
| 7.1252040 | 1.9375000 | -0.0936995 | 9 | 11 | 1 |
| 7.1252040 | 1.9375000 | 0.2083998 | 9 | 12 | 1 |
| 7.1252040 | 1.9375000 | 0.5271006 | 9 | 13 | 1 |
| 7.1252040 | 1.9375000 | 0.8832998 | 9 | 14 | 1 |
| 7.1252040 | 1.9375000 | 1.2979000 | 9 | 15 | 1 |
| 7.1252040 | 1.9167000 | 1.8063000 | 9 | 16 | 1 |
| 7.1252040 | 1.6562990 | 2.0938000 | 9 | 17 | 1 |
| 7.1252040 | 1.2603990 | 2.1563000 | 9 | 18 | 1 |
| 7.1252040 | 0.8438000 | 2.1578990 | 9 | 19 | 1 |
| 7.1252040 | 0.4104000 | 2.2125000 | 9 | 20 | 1 |
| 7.1252040 | 0.0000000 | 2.2188000 | 9 | 21 | 1 |
| 6.5625030 | 0.0000000 | -2.4271000 | 10 | 1 | 1 |
| 6.5625030 | 0.3708000 | -2.4208000 | 10 | 2 | 1 |
| 6.5625030 | 0.7667000 | -2.4061990 | 10 | 3 | 1 |
| 6.5625030 | 1.1667000 | -2.3374990 | 10 | 4 | 1 |
| 6.5625030 | 1.5145990 | -2.1187000 | 10 | 5 | 1 |
| 6.5625030 | 1.7666990 | -1.8083000 | 10 | 6 | 1 |
| 6.5625030 | 1.9041990 | -1.4229000 | 10 | 7 | 1 |
| 6.5625030 | 1.9375000 | -1.0228990 | 10 | 8 | 1 |
| 6.5625030 | 1.9375000 | -0.6666002 | 10 | 9 | 1 |
| 6.5625030 | 1.9375000 | -0.3395996 | 10 | 10 | 1 |
| 6.5625030 | 1.9375000 | -0.0311995 | 10 | 11 | 1 |
| 6.5625030 | 1.9375000 | 0.2708993 | 10 | 12 | 1 |
| 6.5625030 | 1.9375000 | 0.5896006 | 10 | 13 | 1 |
| 6.5625030 | 1.9375000 | 0.9457998 | 10 | 14 | 1 |
| 6.5625030 | 1.9375000 | 1.3634000 | 10 | 15 | 1 |
| 6.5625030 | 1.9167000 | 1.8658000 | 10 | 16 | 1 |
| 6.5625030 | 1.8166990 | 2.3646000 | 10 | 17 | 1 |
| 6.5625030 | 1.3374990 | 2.3646000 | 10 | 18 | 1 |
| 6.5625030 | 0.8812000 | 2.3646000 | 10 | 19 | 1 |
| 6.5625030 | 0.4250000 | 2.3646000 | 10 | 20 | 1 |
| 6.5625030 | 0.0000000 | 2.3646000 | 10 | 21 | 1 |
| 5.9585030 | 0.0000000 | -2.4061990 | 11 | 1 | 1 |
| 5.9585030 | 0.3854000 | -2.3978990 | 11 | 2 | 1 |
| 5.9585030 | 0.7875000 | -2.3853990 | 11 | 3 | 1 |
| 5.9585030 | 1.2020990 | -2.3478990 | 11 | 4 | 1 |
| 5.9585030 | 1.6268000 | -2.2124990 | 11 | 5 | 1 |
| 5.9585030 | 1.8542000 | -1.8395990 | 11 | 6 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 5.9585030 | 1.9062990 | -1.3750000 | 11 | 7 | 1 |
| 5.9585030 | 1.9271000 | -0.9707994 | 11 | 8 | 1 |
| 5.9585030 | 1.9271000 | -0.6187000 | 11 | 9 | 1 |
| 5.9585030 | 1.9271000 | -0.3000002 | 11 | 10 | 1 |
| 5.9585030 | 1.9271000 | 0.0103998 | 11 | 11 | 1 |
| 5.9585030 | 1.9271000 | 0.3125000 | 11 | 12 | 1 |
| 5.9585030 | 1.9271000 | 0.6271000 | 11 | 13 | 1 |
| 5.9585030 | 1.9271000 | 0.9896002 | 11 | 14 | 1 |
| 5.9585030 | 1.9062990 | 1.3979000 | 11 | 15 | 1 |
| 5.9585030 | 1.8832990 | 1.8832990 | 11 | 16 | 1 |
| 5.9585030 | 1.8207990 | 2.4208000 | 11 | 17 | 1 |
| 5.9585030 | 1.3437990 | 2.4271000 | 11 | 18 | 1 |
| 5.9585030 | 0.8896000 | 2.4271000 | 11 | 19 | 1 |
| 5.9585030 | 0.4271000 | 2.4271000 | 11 | 20 | 1 |
| 5.9585030 | 0.0000000 | 2.4271000 | 11 | 21 | 1 |
| 5.3755030 | 0.0000000 | -2.4061990 | 12 | 1 | 1 |
| 5.3755030 | 0.3854000 | -2.3978990 | 12 | 2 | 1 |
| 5.3755030 | 0.7875000 | -2.3853990 | 12 | 3 | 1 |
| 5.3755030 | 1.1978990 | -2.3305990 | 12 | 4 | 1 |
| 5.3755030 | 1.6146000 | -2.2020990 | 12 | 5 | 1 |
| 5.3755030 | 1.8395990 | -1.8270990 | 12 | 6 | 1 |
| 5.3755030 | 1.9020990 | -1.3646000 | 12 | 7 | 1 |
| 5.3755030 | 1.9167000 | -0.9666996 | 12 | 8 | 1 |
| 5.3755030 | 1.9167000 | -0.6146002 | 12 | 9 | 1 |
| 5.3755030 | 1.9167000 | -0.2979002 | 12 | 10 | 1 |
| 5.3755030 | 1.9167000 | 0.0103998 | 12 | 11 | 1 |
| 5.3755030 | 1.9167000 | 0.3104000 | 12 | 12 | 1 |
| 5.3755030 | 1.9167000 | 0.6250000 | 12 | 13 | 1 |
| 5.3755030 | 1.9167000 | 0.9854002 | 12 | 14 | 1 |
| 5.3755030 | 1.8999990 | 1.3916990 | 12 | 15 | 1 |
| 5.3755030 | 1.8750000 | 1.8750000 | 12 | 16 | 1 |
| 5.3755030 | 1.8125000 | 2.4104000 | 12 | 17 | 1 |
| 5.3755030 | 1.3437990 | 2.4271000 | 12 | 18 | 1 |
| 5.3755030 | 0.8896000 | 2.4271000 | 12 | 19 | 1 |
| 5.3755030 | 0.4271000 | 2.4271000 | 12 | 20 | 1 |
| 5.3755030 | 0.0000000 | 2.4271000 | 12 | 21 | 1 |
| 4.7915030 | 0.0000000 | -2.4061990 | 13 | 1 | 1 |
| 4.7915030 | 0.3750000 | -2.3999990 | 13 | 2 | 1 |
| 4.7915030 | 0.7750000 | -2.3604000 | 13 | 3 | 1 |
| 4.7915030 | 1.1875000 | -2.3499990 | 13 | 4 | 1 |
| 4.7915030 | 1.6000000 | -2.2249990 | 13 | 5 | 1 |
| 4.7915030 | 1.8228990 | -1.8478990 | 13 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 4.7915030 | 1.8916990 | -1.3999990 | 13 | 7 | 1 |
| 4.7915030 | 1.9062990 | -1.0000000 | 13 | 8 | 1 |
| 4.7915030 | 1.9062990 | -0.6561995 | 13 | 9 | 1 |
| 4.7915030 | 1.9062990 | -0.3374996 | 13 | 10 | 1 |
| 4.7915030 | 1.9062990 | -0.0311995 | 13 | 11 | 1 |
| 4.7915030 | 1.9062990 | 0.2666998 | 13 | 12 | 1 |
| 4.7915030 | 1.9062990 | 0.5812998 | 13 | 13 | 1 |
| 4.7915030 | 1.9062990 | 0.9375000 | 13 | 14 | 1 |
| 4.7915030 | 1.8853990 | 1.3396000 | 13 | 15 | 1 |
| 4.7915030 | 1.8625000 | 1.8187990 | 13 | 16 | 1 |
| 4.7915030 | 1.7791990 | 2.3291990 | 13 | 17 | 1 |
| 4.7915030 | 1.3125000 | 2.3438000 | 13 | 18 | 1 |
| 4.7915030 | 0.8750000 | 2.3438000 | 13 | 19 | 1 |
| 4.7915030 | 0.4208000 | 2.3438000 | 13 | 20 | 1 |
| 4.7915030 | 0.0000000 | 2.3438000 | 13 | 21 | 1 |
| 4.2295030 | 0.0000000 | -2.3853990 | 14 | 1 | 1 |
| 4.2295030 | 0.3667000 | -2.3750000 | 14 | 2 | 1 |
| 4.2295030 | 0.7500000 | -2.3604000 | 14 | 3 | 1 |
| 4.2295030 | 1.1541990 | -2.3332990 | 14 | 4 | 1 |
| 4.2295030 | 1.5728990 | -2.2186990 | 14 | 5 | 1 |
| 4.2295030 | 1.7979000 | -1.8687000 | 14 | 6 | 1 |
| 4.2295030 | 1.8750000 | -1.4354000 | 14 | 7 | 1 |
| 4.2295030 | 1.8916990 | -1.0311990 | 14 | 8 | 1 |
| 4.2295030 | 1.8957990 | -0.6917000 | 14 | 9 | 1 |
| 4.2295030 | 1.8957990 | -0.3771000 | 14 | 10 | 1 |
| 4.2295030 | 1.8957990 | -0.0728998 | 14 | 11 | 1 |
| 4.2295030 | 1.8957990 | 0.2229004 | 14 | 12 | 1 |
| 4.2295030 | 1.8957990 | 0.5313005 | 14 | 13 | 1 |
| 4.2295030 | 1.8916990 | 0.8895993 | 14 | 14 | 1 |
| 4.2295030 | 1.8750000 | 1.2917000 | 14 | 15 | 1 |
| 4.2295030 | 1.8458000 | 1.7645990 | 14 | 16 | 1 |
| 4.2295030 | 1.7396000 | 2.2333000 | 14 | 17 | 1 |
| 4.2295030 | 1.2833000 | 2.2396000 | 14 | 18 | 1 |
| 4.2295030 | 0.8521000 | 2.2396000 | 14 | 19 | 1 |
| 4.2295030 | 0.4083000 | 2.2396000 | 14 | 20 | 1 |
| 4.2295030 | 0.0000000 | 2.2396000 | 14 | 21 | 1 |
| 3.6665030 | 0.0000000 | -2.3646000 | 15 | 1 | 1 |
| 3.6665030 | 0.3625000 | -2.3604000 | 15 | 2 | 1 |
| 3.6665030 | 0.7396000 | -2.3395990 | 15 | 3 | 1 |
| 3.6665030 | 1.1416990 | -2.3228990 | 15 | 4 | 1 |
| 3.6665030 | 1.5458000 | -2.2124990 | 15 | 5 | 1 |
| 3.6665030 | 1.7812990 | -1.8729000 | 15 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 3.6665030 | 1.8604000 | -1.4436990 | 15 | 7 | 1 |
| 3.6665030 | 1.8750000 | -1.0437000 | 15 | 8 | 1 |
| 3.6665030 | 1.8750000 | -0.7061996 | 15 | 9 | 1 |
| 3.6665030 | 1.8750000 | -0.3936996 | 15 | 10 | 1 |
| 3.6665030 | 1.8750000 | -0.0936995 | 15 | 11 | 1 |
| 3.6665030 | 1.8750000 | 0.1978998 | 15 | 12 | 1 |
| 3.6665030 | 1.8750000 | 0.5063000 | 15 | 13 | 1 |
| 3.6665030 | 1.8708000 | 0.8563004 | 15 | 14 | 1 |
| 3.6665030 | 1.8542000 | 1.2562990 | 15 | 15 | 1 |
| 3.6665030 | 1.8332990 | 1.7292000 | 15 | 16 | 1 |
| 3.6665030 | 1.7082990 | 2.1708000 | 15 | 17 | 1 |
| 3.6665030 | 1.2582990 | 2.1771000 | 15 | 18 | 1 |
| 3.6665030 | 0.8333000 | 2.1771000 | 15 | 19 | 1 |
| 3.6665030 | 0.4042000 | 2.1771000 | 15 | 20 | 1 |
| 3.6665030 | 0.0000000 | 2.1771000 | 15 | 21 | 1 |
| 3.0835030 | 0.0000000 | -2.3021000 | 16 | 1 | 1 |
| 3.0835030 | 0.3479000 | -2.3321000 | 16 | 2 | 1 |
| 3.0835030 | 0.7188000 | -2.2916000 | 16 | 3 | 1 |
| 3.0835030 | 1.0958000 | -2.2666990 | 16 | 4 | 1 |
| 3.0835030 | 1.4896000 | -2.1729000 | 16 | 5 | 1 |
| 3.0835030 | 1.7500000 | -1.8687000 | 16 | 6 | 1 |
| 3.0835030 | 1.8479000 | -1.4686990 | 16 | 7 | 1 |
| 3.0835030 | 1.8750000 | -1.0749990 | 16 | 8 | 1 |
| 3.0835030 | 1.8750000 | -0.7375002 | 16 | 9 | 1 |
| 3.0835030 | 1.8750000 | -0.4250002 | 16 | 10 | 1 |
| 3.0835030 | 1.8750000 | -0.1250000 | 16 | 11 | 1 |
| 3.0835030 | 1.8750000 | 0.1667004 | 16 | 12 | 1 |
| 3.0835030 | 1.8750000 | 0.4750004 | 16 | 13 | 1 |
| 3.0835030 | 1.8637000 | 0.8249998 | 16 | 14 | 1 |
| 3.0835030 | 1.8458000 | 1.2229000 | 16 | 15 | 1 |
| 3.0835030 | 1.8125000 | 1.6771000 | 16 | 16 | 1 |
| 3.0835030 | 1.6374990 | 2.0500000 | 16 | 17 | 1 |
| 3.0835030 | 1.2082990 | 2.0521000 | 16 | 18 | 1 |
| 3.0835030 | 0.8021000 | 2.0521000 | 16 | 19 | 1 |
| 3.0835030 | 0.3854000 | 2.0521000 | 16 | 20 | 1 |
| 3.0835030 | 0.0000000 | 2.0521000 | 16 | 21 | 1 |
| 2.5005030 | 0.0000000 | -2.2396000 | 17 | 1 | 1 |
| 2.5335030 | 0.3333330 | -2.2306000 | 17 | 2 | 1 |
| 2.5005030 | 0.6813000 | -2.2291000 | 17 | 3 | 1 |
| 2.5335030 | 1.0521000 | -2.2145990 | 17 | 4 | 1 |
| 2.5335030 | 1.4473990 | -2.1416990 | 17 | 5 | 1 |
| 2.5005030 | 1.7147990 | -1.8687000 | 17 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 2.5005030 | 1.8542000 | -1.4854000 | 17 | 7 | 1 |
| 2.5005030 | 1.8750000 | -1.1041000 | 17 | 8 | 1 |
| 2.5005030 | 1.8750000 | -0.7686996 | 17 | 9 | 1 |
| 2.5005030 | 1.8750000 | -0.4561996 | 17 | 10 | 1 |
| 2.5005030 | 1.8750000 | -0.1561995 | 17 | 11 | 1 |
| 2.5005030 | 1.8750000 | 0.1353998 | 17 | 12 | 1 |
| 2.5005030 | 1.8750000 | 0.4438000 | 17 | 13 | 1 |
| 2.5005030 | 1.8708000 | 0.7938004 | 17 | 14 | 1 |
| 2.5005030 | 1.8542000 | 1.1916990 | 17 | 15 | 1 |
| 2.5005030 | 1.7917000 | 1.6270990 | 17 | 16 | 1 |
| 2.5005030 | 1.5666990 | 1.9271000 | 17 | 17 | 1 |
| 2.5005030 | 1.1520990 | 1.9271000 | 17 | 18 | 1 |
| 2.5005030 | 0.7667000 | 1.9271000 | 17 | 19 | 1 |
| 2.5005030 | 0.3687000 | 1.9271000 | 17 | 20 | 1 |
| 2.5005030 | 0.0000000 | 1.9271000 | 17 | 21 | 1 |
| 1.8335030 | 0.0000000 | -2.1561990 | 18 | 1 | 1 |
| 1.8335030 | 0.3138000 | -2.1561990 | 18 | 2 | 1 |
| 1.8335030 | 0.6563000 | -2.1520990 | 18 | 3 | 1 |
| 1.8335030 | 1.0145990 | -2.1395990 | 18 | 4 | 1 |
| 1.8335030 | 1.4041990 | -2.0791990 | 18 | 5 | 1 |
| 1.8335030 | 1.6375000 | -1.8395990 | 18 | 6 | 1 |
| 1.8335030 | 1.8125000 | -1.4686990 | 18 | 7 | 1 |
| 1.8335030 | 1.8637000 | -1.1041000 | 18 | 8 | 1 |
| 1.8335030 | 1.8750000 | -0.7686996 | 18 | 9 | 1 |
| 1.8335030 | 1.8750000 | -0.4561996 | 18 | 10 | 1 |
| 1.8335030 | 1.8750000 | -0.1561995 | 18 | 11 | 1 |
| 1.8335030 | 1.8750000 | 0.1353998 | 18 | 12 | 1 |
| 1.8335030 | 1.8750000 | 0.4438000 | 18 | 13 | 1 |
| 1.8335030 | 1.8687000 | 0.7938004 | 18 | 14 | 1 |
| 1.8335030 | 1.8437990 | 1.1916990 | 18 | 15 | 1 |
| 1.8335030 | 1.7812990 | 1.6146000 | 18 | 16 | 1 |
| 1.8335030 | 1.5103990 | 1.8438000 | 18 | 17 | 1 |
| 1.8335030 | 1.1104000 | 1.8438000 | 18 | 18 | 1 |
| 1.8335030 | 0.7354000 | 1.8438000 | 18 | 19 | 1 |
| 1.8335030 | 0.3542000 | 1.8438000 | 18 | 20 | 1 |
| 1.8335030 | 0.0000000 | 1.8438000 | 18 | 21 | 1 |
| 1.1665030 | 0.0000000 | -2.0728990 | 19 | 1 | 1 |
| 1.1665030 | 0.3000000 | -2.0728990 | 19 | 2 | 1 |
| 1.1665030 | 0.6146000 | -2.0625000 | 19 | 3 | 1 |
| 1.1665030 | 0.9458000 | -2.0458000 | 19 | 4 | 1 |
| 1.1665030 | 1.2212990 | -1.9582990 | 19 | 5 | 1 |
| 1.1665030 | 1.5625000 | -1.7603990 | 19 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 1.1665030 | 1.7417000 | -1.4603990 | 19 | 7 | 1 |
| 1.1665030 | 1.7917000 | -1.1041000 | 19 | 8 | 1 |
| 1.1665030 | 1.7917000 | -0.7811995 | 19 | 9 | 1 |
| 1.1665030 | 1.7917000 | -0.4833002 | 19 | 10 | 1 |
| 1.1665030 | 1.7917000 | -0.1978998 | 19 | 11 | 1 |
| 1.1665030 | 1.7917000 | 0.0833998 | 19 | 12 | 1 |
| 1.1665030 | 1.7917000 | 0.3750000 | 19 | 13 | 1 |
| 1.1665030 | 1.7917000 | 0.7146006 | 19 | 14 | 1 |
| 1.1665030 | 1.7707990 | 1.0896000 | 19 | 15 | 1 |
| 1.1665030 | 1.6562990 | 1.4478990 | 19 | 16 | 1 |
| 1.1665030 | 1.3332990 | 1.5687990 | 19 | 17 | 1 |
| 1.1665030 | 1.0382990 | 1.6188000 | 19 | 18 | 1 |
| 1.1665030 | 0.6792000 | 1.6458990 | 19 | 19 | 1 |
| 1.1665030 | 0.3333000 | 1.6708000 | 19 | 20 | 1 |
| 1.1665030 | 0.3333000 | 1.6771000 | 19 | 21 | 1 |
| 0.5205040 | 0.0000000 | -1.9896000 | 20 | 1 | 1 |
| 0.5205040 | 0.2792000 | -1.9875000 | 20 | 2 | 1 |
| 0.5205040 | 0.5708000 | -1.9770990 | 20 | 3 | 1 |
| 0.5205040 | 0.8646000 | -1.9478990 | 20 | 4 | 1 |
| 0.5205040 | 1.1875000 | -1.8811990 | 20 | 5 | 1 |
| 0.5205040 | 1.4687990 | -1.7166990 | 20 | 6 | 1 |
| 0.5205040 | 1.6667000 | -1.4541990 | 20 | 7 | 1 |
| 0.5205040 | 1.7208000 | -1.1250000 | 20 | 8 | 1 |
| 0.5205040 | 1.7292000 | -0.8541002 | 20 | 9 | 1 |
| 0.5205040 | 1.7292000 | -0.5270990 | 20 | 10 | 1 |
| 0.5205040 | 1.7292000 | -0.2500000 | 20 | 11 | 1 |
| 0.5205040 | 1.7292000 | 0.0208998 | 20 | 12 | 1 |
| 0.5205040 | 1.7292000 | 0.2645998 | 20 | 13 | 1 |
| 0.5205040 | 1.7250000 | 0.6250000 | 20 | 14 | 1 |
| 0.5205040 | 1.6375000 | 0.9792004 | 20 | 15 | 1 |
| 0.5205040 | 1.5458000 | 1.2475000 | 20 | 16 | 1 |
| 0.5205040 | 1.2375000 | 1.3832990 | 20 | 17 | 1 |
| 0.5205040 | 0.9292000 | 1.4250000 | 20 | 18 | 1 |
| 0.5205040 | 0.6292000 | 1.4562990 | 20 | 19 | 1 |
| 0.5205040 | 0.3083000 | 1.4833000 | 20 | 20 | 1 |
| 0.5205040 | 0.0000000 | 1.4896000 | 20 | 21 | 1 |
| -0.1454964 | 0.0000000 | -1.9271000 | 21 | 1 | 1 |
| -0.1454964 | 0.2500000 | -1.9271000 | 21 | 2 | 1 |
| -0.1454964 | 0.5208000 | -1.9208000 | 21 | 3 | 1 |
| -0.1454964 | 0.7833000 | -1.8853990 | 21 | 4 | 1 |
| -0.1454964 | 1.0832990 | -1.8311990 | 21 | 5 | 1 |
| -0.1454964 | 1.3542000 | -1.6978990 | 21 | 6 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| -0.1454964 | 1.5521000 | -1.4645990 | 21 | 7 | 1 |
| -0.1454964 | 1.6250000 | -1.1687000 | 21 | 8 | 1 |
| -0.1454964 | 1.6457990 | -0.8792000 | 21 | 9 | 1 |
| -0.1454964 | 1.6457990 | -0.6083002 | 21 | 10 | 1 |
| -0.1454964 | 1.6457990 | -0.3436995 | 21 | 11 | 1 |
| -0.1454964 | 1.6457990 | -0.0853996 | 21 | 12 | 1 |
| -0.1454964 | 1.6457990 | 0.1813002 | 21 | 13 | 1 |
| -0.1454964 | 1.6374990 | 0.4875002 | 21 | 14 | 1 |
| -0.1454964 | 1.5937990 | 0.8167000 | 21 | 15 | 1 |
| -0.1454964 | 1.4125000 | 1.0604000 | 21 | 16 | 1 |
| -0.1454964 | 1.1146000 | 1.1291990 | 21 | 17 | 1 |
| -0.1454964 | 0.8333000 | 1.1604000 | 21 | 18 | 1 |
| -0.1454964 | 0.5708000 | 1.1916990 | 21 | 19 | 1 |
| -0.1454964 | 0.2792000 | 1.2167000 | 21 | 20 | 1 |
| -0.1454964 | 0.0000000 | 1.2396000 | 21 | 21 | 1 |
| -0.8744965 | 0.0000000 | -1.8436990 | 22 | 1 | 1 |
| -0.8744965 | 0.2208000 | -1.8436990 | 22 | 2 | 1 |
| -0.8744965 | 0.4583000 | -1.8395990 | 22 | 3 | 1 |
| -0.8744965 | 0.7000000 | -1.8291990 | 22 | 4 | 1 |
| -0.8744965 | 0.9792000 | -1.7895990 | 22 | 5 | 1 |
| -0.8744965 | 1.2396000 | -1.6812000 | 22 | 6 | 1 |
| -0.8744965 | 1.4167000 | -1.4728990 | 22 | 7 | 1 |
| -0.8744965 | 1.5032990 | -1.2124990 | 22 | 8 | 1 |
| -0.8744965 | 1.5375000 | -0.9478998 | 22 | 9 | 1 |
| -0.8744965 | 1.5500000 | -0.6978998 | 22 | 10 | 1 |
| -0.8744965 | 1.5521000 | -0.4478993 | 22 | 11 | 1 |
| -0.8744965 | 1.5500000 | -0.2061996 | 22 | 12 | 1 |
| -0.8744965 | 1.5458000 | 0.0458002 | 22 | 13 | 1 |
| -0.8744965 | 1.5312990 | 0.3312998 | 22 | 14 | 1 |
| -0.8744965 | 1.4833000 | 0.6313000 | 22 | 15 | 1 |
| -0.8744965 | 1.2624990 | 0.8083000 | 22 | 16 | 1 |
| -0.8744965 | 0.9958000 | 0.8708000 | 22 | 17 | 1 |
| -0.8744965 | 0.7500000 | 0.9000006 | 22 | 18 | 1 |
| -0.8744965 | 0.5104000 | 0.9271032 | 22 | 19 | 1 |
| -0.8744965 | 0.2458000 | 0.9433000 | 22 | 20 | 1 |
| -0.8744965 | 0.0000000 | 0.9475998 | 22 | 21 | 1 |
| -1.9164960 | 0.0000000 | -1.7186990 | 23 | 1 | 1 |
| -1.9164960 | 0.2083000 | -1.7136990 | 23 | 2 | 1 |
| -1.9164960 | 0.4271000 | -1.7166990 | 23 | 3 | 1 |
| -1.9164960 | 0.6533000 | -1.7124990 | 23 | 4 | 1 |
| -1.9164960 | 0.9271000 | -1.6895990 | 23 | 5 | 1 |
| -1.9164960 | 1.1729000 | -1.5874990 | 23 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| -1.9164960 | 1.3166990 | -1.3708000 | 23 | 7 | 1 |
| -1.9164960 | 1.3853990 | -1.1167000 | 23 | 8 | 1 |
| -1.9164960 | 1.4125000 | -0.8750000 | 23 | 9 | 1 |
| -1.9164960 | 1.4167000 | -0.6457996 | 23 | 10 | 1 |
| -1.9164960 | 1.4167000 | -0.4166002 | 23 | 11 | 1 |
| -1.9164960 | 1.4167000 | -0.1957998 | 23 | 12 | 1 |
| -1.9164960 | 1.4167000 | 0.0354004 | 23 | 13 | 1 |
| -1.9164960 | 1.4167000 | 0.3042002 | 23 | 14 | 1 |
| -1.9164960 | 1.3583000 | 0.5688000 | 23 | 15 | 1 |
| -1.9164960 | 1.1533000 | 0.7333002 | 23 | 16 | 1 |
| -1.9164960 | 0.9250000 | 0.8042002 | 23 | 17 | 1 |
| -1.9164960 | 0.6979000 | 0.8396006 | 23 | 18 | 1 |
| -1.9164960 | 0.4708000 | 0.8604002 | 23 | 19 | 1 |
| -1.9164960 | 0.2292000 | 0.8750000 | 23 | 20 | 1 |
| -1.9164960 | 0.0000000 | 0.8853998 | 23 | 21 | 1 |
| -3.2084960 | 0.0000000 | -1.5728990 | 24 | 1 | 1 |
| -3.2084960 | 0.1875000 | -1.5728990 | 24 | 2 | 1 |
| -3.2084960 | 0.3875000 | -1.5686990 | 24 | 3 | 1 |
| -3.2084960 | 0.5958000 | -1.5625000 | 24 | 4 | 1 |
| -3.2084960 | 0.8438000 | -1.5458000 | 24 | 5 | 1 |
| -3.2084960 | 1.0728990 | -1.4541990 | 24 | 6 | 1 |
| -3.2084960 | 1.1957990 | -1.2520990 | 24 | 7 | 1 |
| -3.2084960 | 1.2354000 | -1.0103990 | 24 | 8 | 1 |
| -3.2084960 | 1.2417000 | -0.7916002 | 24 | 9 | 1 |
| -3.2084960 | 1.2500000 | -0.5853996 | 24 | 10 | 1 |
| -3.2084960 | 1.2500000 | -0.3853998 | 24 | 11 | 1 |
| -3.2084960 | 1.2500000 | -0.1896000 | 24 | 12 | 1 |
| -3.2084960 | 1.2500000 | 0.0103998 | 24 | 13 | 1 |
| -3.2084960 | 1.2396000 | 0.2396002 | 24 | 14 | 1 |
| -3.2084960 | 1.1625000 | 0.4604006 | 24 | 15 | 1 |
| -3.2084960 | 1.0312990 | 0.6353998 | 24 | 16 | 1 |
| -3.2084960 | 0.8479000 | 0.7354002 | 24 | 17 | 1 |
| -3.2084960 | 0.6417000 | 0.7708998 | 24 | 18 | 1 |
| -3.2084960 | 0.4333000 | 0.7896004 | 24 | 19 | 1 |
| -3.2084960 | 0.2125000 | 0.7979002 | 24 | 20 | 1 |
| -3.2084960 | 0.0000000 | 0.8021002 | 24 | 21 | 1 |
| -4.6244830 | 0.0000000 | -1.4061990 | 25 | 1 | 1 |
| -4.6244830 | 0.1750000 | -1.4061990 | 25 | 2 | 1 |
| -4.6244830 | 0.3500000 | -1.3999990 | 25 | 3 | 1 |
| -4.6244830 | 0.5375000 | -1.3916990 | 25 | 4 | 1 |
| -4.6244830 | 0.7500000 | -1.3646000 | 25 | 5 | 1 |
| -4.6244830 | 0.9271000 | -1.2582990 | 25 | 6 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| -4.6244880 | 1.0249990 | -1.0791990 | 25 | 7 | 1 |
| -4.6244880 | 1.0625000 | -0.8708000 | 25 | 8 | 1 |
| -4.6244880 | 1.0728990 | -0.6854000 | 25 | 9 | 1 |
| -4.6244880 | 1.0832990 | -0.5042000 | 25 | 10 | 1 |
| -4.6244880 | 1.0832990 | -0.3332996 | 25 | 11 | 1 |
| -4.6244880 | 1.0832990 | -0.1666002 | 25 | 12 | 1 |
| -4.6244880 | 1.0832990 | 0.0103998 | 25 | 13 | 1 |
| -4.6244880 | 1.0749990 | 0.2083998 | 25 | 14 | 1 |
| -4.6244880 | 1.0249990 | 0.4104004 | 25 | 15 | 1 |
| -4.6244880 | 0.9208000 | 0.5791998 | 25 | 16 | 1 |
| -4.6244880 | 0.7708000 | 0.6875000 | 25 | 17 | 1 |
| -4.6244880 | 0.5833000 | 0.7208004 | 25 | 18 | 1 |
| -4.6244880 | 0.3553000 | 0.7375002 | 25 | 19 | 1 |
| -4.6244880 | 0.1917000 | 0.7396002 | 25 | 20 | 1 |
| -4.6244880 | 0.0000000 | 0.7396002 | 25 | 21 | 1 |
| -6.1044990 | 0.0000000 | -1.2395000 | 26 | 1 | 1 |
| -6.1044990 | 0.1530000 | -1.2333000 | 26 | 2 | 1 |
| -6.1044990 | 0.3042000 | -1.2228990 | 26 | 3 | 1 |
| -6.1044990 | 0.4667000 | -1.2020990 | 26 | 4 | 1 |
| -6.1044990 | 0.6333000 | -1.1624990 | 26 | 5 | 1 |
| -6.1044990 | 0.7313000 | -1.0707990 | 26 | 6 | 1 |
| -6.1044990 | 0.8354000 | -0.9103994 | 26 | 7 | 1 |
| -6.1044990 | 0.8750000 | -0.7354002 | 26 | 8 | 1 |
| -6.1044990 | 0.8854000 | -0.5791998 | 26 | 9 | 1 |
| -6.1044990 | 0.8917000 | -0.4333000 | 26 | 10 | 1 |
| -6.1044990 | 0.8958000 | -0.2916002 | 26 | 11 | 1 |
| -6.1044990 | 0.8958000 | -0.1499996 | 26 | 12 | 1 |
| -6.1044990 | 0.8958000 | -0.0061998 | 26 | 13 | 1 |
| -6.1044990 | 0.8875000 | 0.1563005 | 26 | 14 | 1 |
| -6.1044990 | 0.8583000 | 0.3312996 | 26 | 15 | 1 |
| -6.1044990 | 0.7813000 | 0.4875002 | 26 | 16 | 1 |
| -6.1044990 | 0.6563000 | 0.5749994 | 26 | 17 | 1 |
| -6.1044990 | 0.5104000 | 0.6250000 | 26 | 18 | 1 |
| -6.1044990 | 0.3458000 | 0.6416995 | 26 | 19 | 1 |
| -6.1044990 | 0.1667000 | 0.6520000 | 26 | 20 | 1 |
| -6.1044990 | 0.0000000 | 0.6583005 | 26 | 21 | 1 |
| -7.6044990 | 0.0000000 | -1.0521000 | 27 | 1 | 1 |
| -7.6044990 | 0.1333000 | -1.0416000 | 27 | 2 | 1 |
| -7.6044990 | 0.2646000 | -1.0186990 | 27 | 3 | 1 |
| -7.6044990 | 0.3854000 | -0.9728994 | 27 | 4 | 1 |
| -7.6044990 | 0.5000000 | -0.9061995 | 27 | 5 | 1 |
| -7.6044990 | 0.5933000 | -0.8125000 | 27 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| -7.6044990 | 0.6583000 | -0.6999998 | 27 | 7 | 1 |
| -7.6044990 | 0.6917000 | -0.5707998 | 27 | 8 | 1 |
| -7.6044990 | 0.7342000 | -0.4478998 | 27 | 9 | 1 |
| -7.6044990 | 0.7083000 | -0.3332996 | 27 | 10 | 1 |
| -7.6044990 | 0.7083000 | -0.2186995 | 27 | 11 | 1 |
| -7.6044990 | 0.7083000 | -0.1104002 | 27 | 12 | 1 |
| -7.6044990 | 0.7083000 | 0.0063000 | 27 | 13 | 1 |
| -7.6044990 | 0.6979000 | 0.1332998 | 27 | 14 | 1 |
| -7.6044990 | 0.6563000 | 0.2582998 | 27 | 15 | 1 |
| -7.6044990 | 0.6000000 | 0.3792000 | 27 | 16 | 1 |
| -7.6044990 | 0.5167000 | 0.4667005 | 27 | 17 | 1 |
| -7.6044990 | 0.4167000 | 0.5354004 | 27 | 18 | 1 |
| -7.6044990 | 0.2379000 | 0.5812998 | 27 | 19 | 1 |
| -7.6044990 | 0.1458000 | 0.6083002 | 27 | 20 | 1 |
| -7.6044990 | 0.0000000 | 0.6146002 | 27 | 21 | 1 |
| -9.3335030 | 0.0000000 | -0.8646002 | 28 | 1 | 1 |
| -9.3335030 | 0.1083000 | -0.8563002 | 28 | 2 | 1 |
| -9.3335030 | 0.2083000 | -0.8362003 | 28 | 3 | 1 |
| -9.3335030 | 0.2917000 | -0.7479000 | 28 | 4 | 1 |
| -9.3335030 | 0.3542000 | -0.6666002 | 28 | 5 | 1 |
| -9.3335030 | 0.4063000 | -0.5832996 | 28 | 6 | 1 |
| -9.3335030 | 0.4137000 | -0.4958000 | 28 | 7 | 1 |
| -9.3335030 | 0.4625000 | -0.4124994 | 28 | 8 | 1 |
| -9.3335030 | 0.4750000 | -0.3332996 | 28 | 9 | 1 |
| -9.3335030 | 0.4896000 | -0.2521000 | 28 | 10 | 1 |
| -9.3335030 | 0.4896000 | -0.1771002 | 28 | 11 | 1 |
| -9.3335030 | 0.4896000 | -0.1020994 | 28 | 12 | 1 |
| -9.3335030 | 0.4896000 | -0.0228996 | 28 | 13 | 1 |
| -9.3335030 | 0.4833000 | 0.0667000 | 28 | 14 | 1 |
| -9.3335030 | 0.4688000 | 0.1625004 | 28 | 15 | 1 |
| -9.3335030 | 0.4333000 | 0.2563000 | 28 | 16 | 1 |
| -9.3335030 | 0.3875000 | 0.3333998 | 28 | 17 | 1 |
| -9.3335030 | 0.3137000 | 0.3979006 | 28 | 18 | 1 |
| -9.3335030 | 0.2333000 | 0.4520998 | 28 | 19 | 1 |
| -9.3335030 | 0.1167000 | 0.5021000 | 28 | 20 | 1 |
| -9.3335030 | 0.0000000 | 0.5103998 | 28 | 21 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 1 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 2 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 3 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 4 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 5 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 6 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 7 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 8 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 9 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 10 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 11 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 12 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 13 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 14 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 15 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 16 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 17 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 18 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 19 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 20 | 1 |
| -12.2085000 | 0.0000000 | 0.0938005 | 29 | 21 | 1 |
| 0.0 | 0.0 | 0.0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 1 | 2 | -1 |
| 3 | 4 | 1 | 2 | -1 |
| 4 | 5 | 1 | 2 | -1 |
| 5 | 6 | 1 | 2 | -1 |
| 6 | 7 | 1 | 2 | -1 |
| 7 | 8 | 1 | 2 | -1 |
| 8 | 9 | 1 | 2 | -1 |
| 18 | 19 | 5 | 6 | 1 |
| 19 | 20 | 5 | 6 | 1 |
| 20 | 21 | 5 | 6 | 1 |
| 0 | 0 | 0 | 0 | 0 |

ORIGINAL PAGE IS
OF POOR QUALITY

# SAMPLE OUTPUT - FLOWBODY

```
NCSU FLOWBODY PROGRAM: POTENTIAL FLOW + BOUNDARY LAYER + NONUNIFORM SLIPSTREAM FLOW + INTERIOR MASS FLOW

TEST CESSNA 182 WITH NW01 AND NW20 YIELDING 900 PANELS -- FUSELAGE ONLY

NO. OF QUADS.   = 900
NO. OF SECTIONS=   1
MAX. NO. OF ITERATIONS S.FLOW 100

VINF =  0.1000000E+03   VO =  0.1000000E+03   RHO =  0.2370000E-02   PEPA =  0.1700000E+03   HVP = -0.1000000E+00

SPC =  0.2291270E+00   DEP =  0.7420000E+00   CRHA =  0.2400000E+00   TINF =  0.5200000E+03   ZOA =  0.5000000E+10

IWRITE =  0   ISCHI =  0   ISCHB =  0   IMATCH =  0   IOO = 13   ITEST =  0   NODE =   0

COORDINATE CONVERSION FACTOR,    1.0000000.

1 PLANE OF SYMMETRY
CONVERGENCE CRITERIA , 0.00010

 1 VORTEX SYSTEM(S) SPECIFIED
                                 ISPACE= 2
                                  XCMP(1)=   3.00000   AEXP(1)=  -0.20000
                                  XCMP(2)=   4.00000   AEXP(2)=  -0.00000      XMR= -10.00000

FOR VORTEX SYSTEM: 1
   XCVS  12.00000  YCVS   0.0      ZCVS  -0.48710
   RDMS   2.30000  CAO    0.0      PHIO  90.00000
   NCAN  100  NVRMIN    0  NMSO    4  NPTSO   21
   OFACTO 0.1000000E+00

SECTION    1

     BODY-CENTERLINE DIRECTION COSINES( -0.9998 ,  0.0    ,  0.0219 )

  REFERENCE INDUCED VELOCITY =  0.2932882112E+00

  MATCHING OF AIRSTREAM POWER TO DEVELOPED ENGINE POWER(SHP) REQUIRED OPACT TO BE  0.3001039RE+00 TIMES THE ORIGINAL.

  RING VORTEX INFORMATION:

  NUMBER        CENTER COORDINATES          RADIUS   CIRCULATION
                X         Y          Z

     1      12.00000   0.0      -0.48470   2.90000    0.8109724
     2       6.81140   0.0      -0.39470   3.10092    0.6983279
     3       6.11087   0.0      -0.30007   3.49953    0.5900001
     4       P.79844   0.0      -0.2352P   3.33007    0.6112303
     5      -1.12290   0.0      -0.14934   2.95243    0.6916420
     6      -2.01190   0.0      -0.06405   2.74250    0.7442537
     7      -8.87067   0.0       0.02099   2.50312    0.7902981
     8     -12.07970   0.0       0.10404   2.30000    0.8109724
     9     -10.44029   0.0       0.18740   2.30000    0.8109724
    10      -20.P0920   0.0       0.27094   2.30000    0.8109724

  M      X1         X2         X3         X4         XC         XN         A          CZ4
  N      Y1         Y2         Y3         Y4         YC         YN         P.         CY4
  O      Z1         Z2         Z3         Z4         ZC         ZN         CZ1        CZ6

QUESTIONABLE POINT -POOR FIT      0.4138E+00
WARNING LONG THIN QUAD.
  1   0.11942E+02   0.11942E+02   0.11942E+02   0.11942E+02   0.11942E+02   0.77420E+00   0.11074E+00  -0.99443E+00
  1   0.0           0.0           0.18330E+00   0.0           0.61100E-01   0.60177E+00   0.99299E+00  -0.21142E+00
  1  -0.48710E+00  -0.48710E+00  -0.15970E+01  -0.16374E+01  -0.12281E+01  -0.40422E+00   0.77401E+00   0.79040E+00

QUESTIONABLE POINT -POOR FIT      0.4002E-02
  1   0.11942E+02   0.11942E+02   0.10879E+02   0.10879E+02   0.11201E+02   0.30072E+00   0.14014E+00  -0.19940E+00
  2   0.0           0.14330E+00   0.22080E+00   0.0           0.10202E+00   0.90512E-01   0.30297E+00  -0.10318E+00
  2  -0.14390E+01  -0.15979E+01  -0.19229E+01  -0.18271E+01  -0.17774E+01  -0.91912E+00   0.74970E+00  -0.37730E+00

QUESTIONABLE POINT -POOR FIT      0.8378E-01
WARNING LONG THIN QUAD.
  2   0.11942E+02   0.11942E+02   0.11942E+02   0.11942E+02   0.11942E+02   0.79990E+00   0.10889E+00  -0.49822E+00
  1   0.0           0.0           0.39830E+00   0.18330E+00   0.10053E+00   0.61209E+00   0.90690E+00  -0.53905E+00
  2  -0.48710E+00  -0.48710E+00  -0.19271E+01  -0.15979E+01  -0.11800E+01  -0.23079E+00   0.28810E+00  -0.11448E+01

QUESTIONABLE POINT -POOR FIT      0.5298E-02
  [illegible]
  [illegible]
  [illegible]

QUESTIONABLE POINT -POOR FIT      0.2002E+00
WARNING LONG THIN QUAD.
  [illegible]
  [illegible]
  [illegible]

        .                          .                          .
        .                          .                          .
        .                          .                          .

QUESTIONABLE POINT -POOR FIT      0.4141E+02
WARNING LONG THIN QUAD.
  [illegible]
  [illegible]
  [illegible]

WARNING LONG THIN QUAD.
  [illegible]
  [illegible]
  [illegible]

QUESTIONABLE POINT -POOR FIT      0.2092E+01
WARNING LONG THIN QUAD.
  [illegible]
  [illegible]
  [illegible]
```

DESIGNATION OF NORMAL VELOCITY ON QUADRILATERALS

| | | | | |
|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 1 HAS BEEN DESIGNATED TO HAVE A NONZERO INWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 5 HAS BEEN DESIGNATED TO HAVE A NONZERO INWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 7 HAS BEEN DESIGNATED TO HAVE A NONZERO INWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 9 HAS BEEN DESIGNATED TO HAVE A NONZERO INWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 11 HAS BEEN DESIGNATED TO HAVE A NONZERO INWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 13 HAS BEEN DESIGNATED TO HAVE A NONZERO INWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 15 HAS BEEN DESIGNATED TO HAVE A NONZERO INWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 115 HAS BEEN DESIGNATED TO HAVE A NONZERO OUTWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 117 HAS BEEN DESIGNATED TO HAVE A NONZERO OUTWARD NORMAL COMPONENT OF VELOCITY. |
| [illegible] | [illegible] | [illegible] | [illegible] | PANEL 119 HAS BEEN DESIGNATED TO HAVE A NONZERO OUTWARD NORMAL COMPONENT OF VELOCITY. |

PROBABLE ERROR IN INPUT - SOLID ANGLE = [illegible]

INDUCED VELOCITY BY RING VORTICES AT ORIGINAL-BODY PANEL CENTROIDS:

| PANEL NUMBER | PANEL CENTROID LOCATION X | Y | Z | VELOCITY COMPONENTS X | Y | Z |
|---|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 991 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 992 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 993 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 994 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 995 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 996 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 997 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 998 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 999 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

SOURCE DENSITY SOLUTION

[illegible] CESSNA 182 WITH [illegible] AND [illegible] YIELDING [illegible] PANELS — FUSELAGE ONLY

X VELOCITY=-1.0 Y VELOCITY= 0.0 Z VELOCITY= 0.0

ITERATIVE MATRIX SOLUTION INFORMATION

| ITERATION | SUM OF CHANGES | A | B1 | B2 |
|---|---|---|---|---|
| 1 | [illegible] | | | |
| 2 | [illegible] | | | |
| 3 | [illegible] | | | |
| 4 | [illegible] | | | |
| 5 | [illegible] | | | |
| | | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | | | |
| 7 | [illegible] | | | |
| 8 | [illegible] | | | |
| 9 | [illegible] | | | |
| 10 | [illegible] | | | |
| | | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | | | |
| 12 | [illegible] | | | |
| 13 | [illegible] | | | |
| 14 | [illegible] | | | |
| 15 | [illegible] | | | |
| | | A EXTRAPOLATION [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | | | |

ORIGINAL PAGE IS
OF POOR QUALITY

COMPUTATION OF VELOCITIES AND PRESSURE COEFFICIENTS AT THE PANEL CENTROIDS

BEST CESSNA 182 WITH ... 400 PANELS — FUSELAGE ONLY PAGE = 1

| PT. | XC | YC | ZC | VX | VY | VZ | ABS.V | CP | SOURCE | V NORMAL | AREA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| . | . | | | | | . | | | | | . |
| . | . | | | | | . | | | | | . |
| . | . | | | | | . | | | | | . |
| 391 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 392 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 393 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 394 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 395 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 396 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 397 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 398 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 399 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 400 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

PRESSURE LIFT AND DRAG COEFFICIENTS

| | |
|---|---|
| PRESSURE CL = | 0.00690 |
| PRESSURE CD = | 0.02065 |
| REFERENCE AREA = | 170.00000 |
| REYNOLDS NUMBER = | 0.2279E+08 |

| PT | [illegible] | [illegible] | [illegible] | CP | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | PT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 1 |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 2 |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 3 |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 4 |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 5 |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 6 |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 7 |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 8 |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 9 |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 10 |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 11 |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 12 |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 13 |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 14 |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 15 |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 16 |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 17 |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 18 |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 19 |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 20 |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 21 |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 22 |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 23 |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 24 |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 25 |
| | . | | | | | . | | | | . | |
| | . | | | | | . | | | | . | |
| | . | | | | | . | | | | . | |
| 391 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 391 |
| 392 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 392 |
| 393 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 393 |
| 394 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 394 |
| 395 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 395 |
| 396 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 396 |
| 397 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 397 |
| 398 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 398 |
| 399 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 399 |
| 400 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 400 |

*** CALCULATION OF INTERIOR PRESSURE COEFFICIENTS AND NORMAL VELOCITIES ***

AVERAGE CP ON INLET PANELS = [illegible] INLET AREA = [illegible]
AVERAGE CP ON EXHAUST PANELS= [illegible] EXHAUST AREA= [illegible]

CPI = [illegible]
CPIS= [illegible]

ADJUSTMENT DUE TO HEATING OF FLOW: TEMPERATURE RATIO = [illegible]
EFFECTIVE ORIFICE AREA= [illegible]

CPI = [illegible]
CPIS= [illegible]

NORMAL VELOCITY AT PANEL 1 = [illegible]

NORMAL VELOCITY AT PANEL 5 = [illegible]

NORMAL VELOCITY AT PANEL 7 = [illegible]

NORMAL VELOCITY AT PANEL 9 = [illegible]

NORMAL VELOCITY AT PANEL 11 = [illegible]

NORMAL VELOCITY AT PANEL 13 = [illegible]

NORMAL VELOCITY AT PANEL 15 = [illegible]

NORMAL VELOCITY AT PANEL 115 = [illegible]

NORMAL VELOCITY AT PANEL 117 = [illegible]

NORMAL VELOCITY AT PANEL 119 = [illegible]

ORIGINAL PAGE IS
OF POOR QUALITY

SOURCE DENSITY SOLUTION

REST CESSNA 182 WITH HHD1 AND HHD2 YIELDING 500 PANELS -- FUSELAGE ONLY

X VELOCITY=-1.0 Y VELOCITY= 0.0 Z VELOCITY= 0.0

ITERATIVE MATRIX SOLUTION INFORMATION

| ITERATION | SUM OF CHANGES | A | B1 | B2 |
|---|---|---|---|---|
| 1 | [illegible] | | | |
| 2 | [illegible] | | | |
| 3 | [illegible] | | | |
| 4 | [illegible] | | | |
| 5 | [illegible] | | | |
| | | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | | | |
| 7 | [illegible] | | | |
| 8 | [illegible] | | | |
| 9 | [illegible] | | | |
| 10 | [illegible] | | | |
| | | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | | | |
| 12 | [illegible] | | | |
| 13 | [illegible] | | | |
| 14 | [illegible] | | | |
| 15 | [illegible] | | | |
| | A EXTRAPOLATION | [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | | | |

COMPUTATION OF VELOCITIES AND PRESSURE COEFFICIENTS AT THE PANEL CENTROIDS

REST CESSNA 182 WITH HHD1 AND HHD2 YIELDING 500 PANELS -- FUSELAGE ONLY PAGE 1

X FLOW

| NO. | XC | YC | ZC | VX | VY | VZ | ABS.V | CP | SOURCE | V NORMAL | AREA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| . | . | | | | . | | | | | | . |
| . | . | | | | . | | | | | | . |
| . | . | | | | . | | | | | | . |
| 491 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 492 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 493 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 494 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 495 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 496 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 497 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 498 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 499 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 500 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

PRESSURE LIFT AND DRAG COEFFICIENTS

```
********************************
     PRESSURE CL =      0.00248
     PRESSURE CD =      0.01798
  REFERENCE AREA =    174.00000
 REYNOLDS NUMBER =  0.2379E+08
********************************
```

| PT | XMN | YMN | ZMN | CP | VXDXMN | VYDYMN | VZDZMN | AREA | 2*CP*AREA*NX*ZMN | 2*CP*AREA*NZ*XMN | PT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.77828 | 0.40277 | -0.40022 | 0.00094 | -0.04033 | 0.09140 | 0.09075 | 0.111 | -0.0019792 | 0.0008129 | 1 |
| 2 | 0.39072 | 0.47001 | -0.91912 | -0.16790 | -0.30063 | 0.00016 | 0.30038 | 0.149 | 0.0000370 | -0.0190009 | 2 |
| 3 | 0.79990 | 0.41309 | -0.22079 | 0.57886 | -0.32113 | 0.27722 | 0.04380 | 0.109 | -0.0201030 | 0.0092309 | 3 |
| 4 | 0.47708 | 0.18000 | -0.29174 | -0.12230 | -0.41381 | 0.08091 | 0.00321 | 0.158 | 0.0330716 | -0.0144079 | 4 |
| 5 | 0.80044 | 0.36800 | -0.37099 | 0.23208 | -0.77741 | 0.03208 | 0.02119 | 0.181 | -0.0130236 | 0.0021090 | 5 |
| 6 | 0.40090 | 0.29749 | -0.23473 | -0.00014 | -0.62014 | 0.03797 | 0.30087 | 0.155 | 0.0189238 | -0.0090771 | 6 |
| 7 | 0.80229 | 0.42707 | -0.19003 | 0.00087 | -0.70049 | 0.10129 | 0.01200 | 0.184 | -0.0150011 | 0.0024737 | 7 |
| 8 | 0.40774 | 0.40718 | -0.77772 | -0.01200 | -0.42030 | 0.00173 | -0.33084 | 0.100 | 0.0030024 | -0.0010302 | 8 |
| 9 | 0.80119 | -0.08804 | -0.47276 | 0.37988 | -0.07027 | -0.00012 | 0.80447 | 0.181 | -0.0363000 | 0.0078814 | 9 |
| 10 | 0.40804 | 0.57107 | -0.40980 | 0.01234 | -0.42343 | 0.14940 | 0.29398 | 0.157 | -0.0027130 | 0.0010962 | 10 |
| 11 | 0.80020 | 0.13004 | -0.43403 | 0.23013 | -0.70070 | 0.02377 | -0.00022 | 0.113 | -0.0229909 | 0.0403309 | 11 |
| 12 | 0.44741 | 0.02024 | -0.42196 | -0.00090 | -0.41187 | 0.20439 | 0.14042 | 0.100 | 0.0081204 | -0.0000009 | 12 |
| 13 | 0.49117 | -0.03941 | -0.73129 | 0.80430 | -0.41170 | -0.30923 | 0.17390 | 0.112 | -0.0014742 | 0.0092317 | 13 |
| 14 | 0.43040 | 0.70973 | -0.18284 | -0.04197 | -0.39336 | 0.29092 | 0.00373 | 0.100 | 0.0079407 | -0.0037784 | 14 |
| 15 | 0.67770 | 0.12740 | -0.72705 | 0.03490 | -0.40730 | 0.02048 | -0.00094 | 0.129 | -0.0019107 | 0.0794981 | 15 |
| 16 | 0.30920 | 0.78292 | -0.21619 | -0.00024 | -0.30703 | 0.32790 | 0.03994 | 0.107 | 0.0142163 | -0.0107219 | 16 |
| 17 | 0.72000 | 0.03112 | -0.94974 | 0.31387 | -0.24809 | 0.00970 | 0.22789 | 0.191 | -0.0094927 | 0.0304322 | 17 |
| 18 | 0.32020 | 0.47904 | -0.40002 | -0.14987 | -0.31947 | 0.33242 | -0.01302 | 0.173 | 0.0201001 | -0.0161241 | 18 |
| 19 | 0.37990 | 0.30401 | -0.29422 | 0.23061 | -0.26450 | 0.07979 | 0.18440 | 0.171 | -0.0703990 | 0.0230499 | 19 |
| 20 | 0.20067 | 0.02810 | -0.20302 | -0.10220 | -0.29040 | 0.27905 | -0.02048 | 0.174 | 0.0180011 | -0.0193004 | 20 |
| 21 | 0.20174 | 0.40907 | -0.80007 | 0.00410 | -0.10930 | 0.09176 | 0.09338 | 0.103 | -0.0307264 | 0.0050092 | 21 |
| 22 | 0.15307 | 0.98290 | -0.10119 | -0.19019 | -0.10964 | 0.17794 | -0.01299 | 0.171 | 0.0009023 | -0.0104929 | 22 |
| 23 | 0.17327 | 0.40042 | -0.77744 | 0.03757 | -0.10812 | 0.12097 | 0.00993 | 0.206 | -0.0119397 | 0.0020943 | 23 |
| 24 | 0.09904 | 0.90408 | 0.03740 | -0.10374 | -0.10399 | 0.10002 | 0.00347 | 0.178 | -0.0024481 | -0.3002709 | 24 |
| 25 | 0.32016 | 0.01098 | 0.49130 | 0.00005 | -0.29493 | 0.30171 | 0.21318 | 0.213 | 0.0290313 | 0.0111224 | 25 |
| . | | | | | . | | | | | . | |
| . | | | | | . | | | | | . | |
| . | | | | | . | | | | | . | |
| 552 | -0.14007 | 0.00000 | 0.67763 | -0.29523 | 0.16701 | -0.10507 | -0.00100 | 0.131 | -0.0309400 | 0.0113798 | 552 |
| 553 | -0.10182 | 0.02108 | 0.77712 | -0.41060 | 0.11047 | -0.02087 | -0.09000 | 0.187 | -0.1193001 | 0.0190024 | 553 |
| 554 | -0.16311 | 0.75240 | 0.00725 | -0.29738 | 0.10129 | -0.00329 | -0.09743 | 0.137 | -0.0360723 | 0.0116729 | 554 |
| 555 | -0.08740 | 0.04052 | 0.49040 | -0.41490 | 0.10009 | -0.00304 | -0.10272 | 0.197 | -0.1459027 | 0.0146791 | 555 |
| 556 | -0.13774 | 0.59127 | 0.79402 | -0.30042 | 0.15521 | -0.03914 | -0.11000 | 0.147 | -0.0781020 | 0.0171092 | 556 |
| 557 | -0.07777 | 0.28194 | 0.49070 | -0.41772 | 0.00629 | 0.00195 | -0.08707 | 0.242 | -0.1930047 | 0.0167709 | 557 |
| 558 | -0.13102 | 0.40774 | 0.90374 | -0.30318 | 0.14029 | -0.01993 | -0.12341 | 0.184 | -0.1009007 | 0.0167939 | 558 |
| 559 | -0.06000 | 0.07943 | 0.90099 | -0.41081 | 0.07238 | 0.00032 | -0.07230 | 0.220 | -0.1809037 | 0.0114070 | 559 |
| 560 | -0.12970 | 0.15309 | 0.47470 | -0.30473 | 0.14522 | -0.00326 | -0.14209 | 0.179 | -0.1099458 | 0.0132745 | 560 |

CALCULATION OF ON-BODY STREAMLINES

REST CESSNA 182 WITH ... AND ... YIELDING 560 PANELS -- FUSELAGE ONLY
FOR V INFINITY=-1.0,0.0,0.0, COMPUTE 560 STREAMLINES STARTING AT EACH PANEL CENTROID POINT

LINE PASSING THROUGH QUADRILATERAL 2

| I | X | Y | Z | CP | SL | VABS |
|---|---|---|---|---|---|---|
| 1 | 11.51400 | 0.00210 | -1.63287 | 0.07221 | 0.0 | 0.67394 |
| 2 | 11.20092 | 0.18704 | -1.77749 | -0.16790 | 0.37032 | 1.08040 |
| 3 | 10.87201 | 0.00307 | -1.91910 | -0.37940 | 0.73907 | 1.17201 |
| 4 | 10.28090 | 0.02904 | -2.11670 | -0.57300 | 1.42024 | 1.23493 |
| 5 | 9.67032 | 0.07075 | -2.23784 | -0.80002 | 2.10776 | 1.30508 |

TRANSITION AT S = 0.00614 FOR STEP NUMBER 7

| S | V | RUDS | HMEAN | DELTAS | THETAS | TAU | CFI |
|---|---|---|---|---|---|---|---|
| 0.0 | 199.78400 | 49.71204 | 2.55400 | 0.0 | 0.0 | 5000.00000 | 1000.00000 |
| 0.37631 | 179.01041 | 48.26318 | 2.43074 | 0.30000 | 0.00001 | 0.05100 | 0.00170 |
| 0.75907 | 197.64090 | 32.40013 | 2.42251 | 0.00107 | 0.00002 | 0.04419 | 0.00197 |
| 1.69424 | 200.77419 | 13.72420 | 1.49000 | 0.00000 | 0.00002 | 0.03537 | 0.00110 |
| 2.18423 | 219.47999 | 40.01703 | 1.60000 | 0.00287 | 0.00057 | 0.17299 | 0.33070 |

LINE PASSING THROUGH QUADRILATERAL 4

| I | X | Y | Z | CP | SL | VABS |
|---|---|---|---|---|---|---|
| 1 | 11.24100 | 0.21309 | -1.52910 | 0.12210 | 0.0 | 0.93003 |
| 2 | 11.10409 | 0.20290 | -1.73900 | -0.13230 | 0.30042 | 1.05962 |
| 3 | 10.87030 | 0.20709 | -1.90773 | -0.30057 | 0.79000 | 1.17770 |
| 4 | 10.21030 | 0.20000 | -2.12070 | -0.57000 | 1.49303 | 1.29974 |
| 5 | 9.00092 | -0.24702 | -2.14700 | -0.67103 | 1.67432 | 1.29240 |

TRANSITION AT S = 0.00118 FOR STEP NUMBER 9

| S | V | RUDS | HMEAN | DELTAS | THETAS | TAU | CFI |
|---|---|---|---|---|---|---|---|
| 0.0 | 149.40000 | 40.09713 | 2.99400 | 0.0 | 0.0 | 5000.00000 | 1000.00000 |
| 0.30042 | 180.09000 | 94.01829 | 2.41073 | 0.30009 | 0.00001 | 0.03290 | 0.00192 |
| 0.79040 | 190.63170 | 39.03221 | 2.41277 | 0.30103 | 0.10332 | 0.06451 | 0.00190 |
| 1.49307 | 200.41701 | 19.22234 | 1.49000 | 0.30004 | 0.00132 | 0.03777 | 0.10176 |
| 1.67432 | 230.87909 | -34.40470 | 1.64000 | 0.30142 | 0.00384 | 0.14900 | 0.33041 |

.
.
.

ORIGINAL PAGE IS
OF POOR QUALITY

LINE PASSING THROUGH QUADRILATERAL 559

| I | X | Y | Z | CP | S | UEDS |
|---|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | 0.0 | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 26 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 27 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 28 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

TRANSITION AT S = [illegible] FOR STEP NUMBER [illegible]

| S | V | DUDS | HMEAN | DELTAS | THETAS | TAU | CFI |
|---|---|---|---|---|---|---|---|
| 0.0 | [illegible] | [illegible] | [illegible] | 0.0 | 0.0 | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

SUMMARY OF BOUNDARY LAYER INFORMATION FOR QUADRILATERALS

| NQUAD | X | Y | Z | DSTAR | SKIN |
|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| ⋮ | | ⋮ | | ⋮ | |
| 451 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 452 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 453 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 454 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 455 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 456 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 457 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 458 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 459 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 460 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

FRICTION DRAG COEFFICIENT

```
******************************
     FRICTION CD =    [illegible]
  REFERENCE AREA =    [illegible]
 REYNOLDS NUMBER =    [illegible]
     BODY LENGTH =    [illegible]
******************************
```

ORIGINAL PAGE IS
OF POOR QUALITY

BEGIN WAKE BODY GEOMETRY

AVERAGE SLOPE = [illegible] AVERAGE PANEL AREA = [illegible] AVERAGE X-CENTROID = [illegible] END OF BODY AT X = [illegible]

SECTION 1000

| N | X1 | X2 | X3 | X4 | XC | XN | A | C24 |
|---|---|---|---|---|---|---|---|---|
| N | Y1 | Y2 | Y3 | Y4 | YC | YN | PL | C25 |
| P | Z1 | Z2 | Z3 | Z4 | ZC | ZN | C21 | C26 |
| WARNING LONG THIN QUAD. | | | | | | | | |
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 1 | 0.0 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| WARNING LONG THIN QUAD. | | | | | | | | |
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| WARNING LONG THIN QUAD. | | | | | | | | |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| WARNING LONG THIN QUAD. | | | | | | | | |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| WARNING LONG THIN QUAD. | | | | | | | | |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

| N | X1 | X2 | X3 | X4 | XC | XN | A | C24 |
|---|---|---|---|---|---|---|---|---|
| N | Y1 | Y2 | Y3 | Y4 | YC | YN | PL | C25 |
| P | Z1 | Z2 | Z3 | Z4 | ZC | ZN | C21 | C26 |
| WARNING LONG THIN QUAD. | | | | | | | | |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 4 | [illegible] | [illegible] | 0.0 | 0.0 | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

PROBABLE ERROR IN INPUT - SOLID ANGLE = [illegible]

INDUCED VELOCITY BY RING VERTICES AT WAKE-BODY PANEL CENTROIDS:

| PANEL NUMBER | PANEL CENTROID LOCATION X | Y | Z | VELOCITY COMPONENTS X | Y | Z |
|---|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| . | | | . | | | . |
| . | | | . | | | . |
| . | | | . | | | . |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

SOURCE DENSITY SOLUTION

BEST CESSNA 182 WITH [illegible] YIELDING [illegible] PANELS — FUSELAGE ONLY

X VELOCITY=1.0    Y VELOCITY= 0.0    Z VELOCITY= 0.0

ITERATIVE MATRIX SOLUTION INFORMATION

| ITERATION | SUM OF CHANGES | A | B1 | B2 |
|---|---|---|---|---|
| 1 | [illegible] | | | |
| 2 | [illegible] | | | |
| 3 | [illegible] | | | |
| 4 | [illegible] | | | |
| 5 | [illegible] | | | |
| | | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | | | |
| 7 | [illegible] | | | |
| 8 | [illegible] | | | |
| 9 | [illegible] | | | |
| 10 | [illegible] | | | |
| | | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | | | |
| 12 | [illegible] | | | |
| 13 | [illegible] | | | |
| 14 | [illegible] | | | |
| 15 | [illegible] | | | |
| | | A EXTRAPOLATION [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | | | |
| 17 | [illegible] | | | |
| 18 | [illegible] | | | |
| 19 | [illegible] | | | |
| 20 | [illegible] | | | |
| | | A EXTRAPOLATION [illegible] | [illegible] | [illegible] |
| 21 | [illegible] | | | |
| 22 | [illegible] | | | |
| 23 | [illegible] | | | |
| 24 | [illegible] | | | |
| 25 | [illegible] | | | |
| | | A EXTRAPOLATION [illegible] | [illegible] | [illegible] |
| 26 | [illegible] | | | |

COMPUTATION OF VELOCITIES AND PRESSURE COEFFICIENTS AT THE PANEL CENTROIDS

BEST CESSNA 182 WITH [illegible] YIELDING [illegible] PANELS -- FUSELAGE ONLY    PAGE = 1

[illegible]

| PT. | XC | YC | ZC | VX | VY | VZ | ABS.V | CP | SOURCE | V NORMAL | AREA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| | . | | | | | . | | | | | . |
| | . | | | | | . | | | | | . |
| | . | | | | | . | | | | | . |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

PRESSURE LIFT AND DRAG COEFFICIENTS

| | |
|---|---|
| PRESSURE CL = | [illegible] |
| PRESSURE CD = | [illegible] |
| REFERENCE AREA = | [illegible] |
| REYNOLDS NUMBER = | [illegible] |

ORIGINAL PAGE IS
OF POOR QUALITY

| ST | XMP | YMP | ZMP | CP | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | ST |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 1 |
| 2 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 2 |
| 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 3 |
| 4 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 4 |
| 5 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 5 |
| 6 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 6 |
| 7 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 7 |
| 8 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 8 |
| 9 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 9 |
| 10 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 10 |
| 11 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 11 |
| 12 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 12 |
| 13 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 13 |
| 14 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 14 |
| 15 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 15 |
| 16 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 16 |
| 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 17 |
| 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 18 |
| 19 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 19 |
| 20 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 20 |
| 21 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 21 |
| 22 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 22 |
| 23 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 23 |
| 24 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 24 |
| 25 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 25 |

•
•
•

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 580 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 580 |
| 581 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 581 |
| 582 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 582 |
| 583 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 583 |
| 584 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 584 |
| 585 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 585 |
| 586 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 586 |
| 587 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 587 |
| 588 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 588 |
| 589 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 589 |
| 590 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 590 |
| 591 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 591 |
| 592 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 592 |
| 593 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 593 |
| 594 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 594 |
| 595 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 595 |
| 596 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 596 |
| 597 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 597 |
| 598 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 598 |
| 599 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 599 |
| 600 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 600 |

TOTAL BODY COEFFICIENTS

| | | |
|---|---|---|
| TOTAL BODY CL | = | [illegible] |
| TOTAL BODY CD | = | [illegible] |
| REFERENCE AREA | = | [illegible] |
| REYNOLDS NUMBER | = | [illegible] |
| BODY LENGTH | = | [illegible] |

# USER'S INSTRUCTIONS - GRIDPLOT PROGRAM

The program is written in FORTRAN IV and is designed to execute in single precision on an IBM 370/165 computer with an average execution time of 1 minute for each data set. An average execution requires approximately 320,000 bytes of core storage. The program accepts multiple data sets.

Given a data set describing the half-body* under consideration, the program may be instructed to perform the following:

(a) Generate a properly-indexed data set with additional body (M,N) stations,

(b) Modify the input data by a simple averaging technique, a linear-interpolation technique, and/or user-specified data-point change information,

(c) Plot various orthographic, perspective, and/or stereoscopic views of the input data and/or the modified data,

(d) Refine the grid network by either an equal-line augmentation scheme or a user-specified line augmentation scheme,

(e) Punch properly-indexed data sets of the input and/or modified data set(s) for input into the NCSU PLOT program of NASA CR-2523 and/or into the FLOWBODY program of this report, and

(f) Convert the data into different units.

*Since the body is considered to be symmetrical about the X-Z plane, only half of the body is needed to describe the entire body.

The program requires the specification of the following input in the indicated order:

CARD 1:

(a) The maximum number N of N-stations present in any input data set to be tried:

N is a right-adjusted positive integer number occupying columns 1-5.

(b) The maximum number M of M-stations present in any input data set to be tried:

M is a right-adjusted positive integer number occupying columns 6-10.

(c) The maximum number NADD of additional N-stations present in any input data set to be tried:

NADD is a right-adjusted positive integer number occupying columns 11-15.

(d) The maximum number MADD of additional M-stations present in any input data set to be tried:

MADD is a right-adjusted positive integer number occupying columns 16-20.

(e) The horizontal (x-direction) length PXL of the plotting picture:

PXL is a single-precision floating-point number occupying columns 21-30 in an F10.0 field. The units of PXL must be appropriate to the installation. At NCSU, PXL must be in inches.

(f) The width (y-direction) PYL of the plotting picture:

PYL is a single-precision floating-point number occupying columns 31-40 in a F10.0 field. The units of PYL must be appropriate to the installation. At NCSU, PYL must be in inches.

Important: Only one "Card 1" is permitted per execution.

CARD 2:

(a) The read unit number IDS:

IDS is a right-adjusted integer number occupying columns 1-5 and specifying that the data is to be read from cards, magnetic tape, disk, etc. The user must supply the suitable job control cards for the tape and/or disk reads. The IDS parameter controls only the reading of CARD 3, CARD 4, and the Body Description cards.

(b) The desired number INPTM of additional interior M-stations:

INPTM is a right-adjusted integer number occupying columns 6-10. INPTM may be negative, zero, or positive. If INPTM is negative, no grid refinement of the M-station is performed. If INPTM is zero, no equal-line augmentation is performed but allows for refinement by the user-specified line-augmentation scheme. If INPTM is positive, the equal-line augmentation scheme may (to be explained later) be used with the number of additional M-stations between each two successive input M-stations equal to INPTM.

(c) The desired number INPTN of additional interior N-stations:

INPTN is a right-adjusted integer number occupying columns 11-15. INPTN may be negative, zero, or positive. If INPTN is negative, no grid refinement of the N-stations is performed. If INPTN is zero, no equal-line augmentation is performed but allows for refinement by the user-specified line-augmentation scheme. If INPTN is positive, the equal-line augmentation scheme may (to be explained later) be used with the number of additional N-stations between each two successive input M-stations equal to INPTN.

(d) The punch option IPUNCH:

IPUNCH is a right-adjusted nonnegative integer number, occupying columns 16-20, that specifies the punching of the data set with the additional body (M,N) stations. If IPUNCH = 0, no cards are punched. If IPUNCH = 1, cards are punched.

(e) The input-data plot option IPLOT1:

IPLOT1 is a right-adjusted nonnegative integer number occupying columns 21-25. If IPLOT1 = 0, no plots are produced. If IPLOT1 = 1, plots are produced.

(f) The modified-data plot option IPLOT2:

IPLOT2 is a right-adjusted nonnegative integer number occupying columns 26-30. If IPLOT2 = 0, no plots are produced. If IPLOT2 = 1, plots are produced.

(g) The input-data punch option LPCH1:

LPCH1 is a right-adjusted nonnegative integer number, occupying columns 31-35, that specifies the punching of the input data in a compatible form for the PLOT program given in NASA CR-2523 [Reference 3]. If LPCH1 = 0, no cards are punched. If LPCH1 = 1, cards are punched.

(h) The modified-data punch option LPCH2:

LPCH2 is a right-adjusted nonnegative integer number, occupying columns 36-40, that specifies the punching of the modified data in a compatible form for the PLOT program given in NASA CR-2523 [Reference 3]. If LPCH2 = 0, no cards are punched. If LPCH2 = 1, cards are punched.

(i) The write option IWRITE:

IWRITE is a right-adjusted nonnegative integer number, occupying columns 41-45, that specifies the amount of desired output. If IWRITE = 0, maximum printout is produced. If IWRITE = 2, minimum printout is produced. If IWRITE = 1, the amount of printout is between the minimum and maximum.

(j) The conversion factor CF:

CF is a single-precision floating-point number, occupying columns 51-60 in an F10.0 field, that may be used to convert the data of CARD 5 through CARD (K + 4) from one set of units to another.

CARD 3:

The title array TITLE:

The 80 characters of the array TITLE are used for identifying output. The reading of TITLE is controlled by the read unit number IDS.

CARD 4:

The number NQE of quadrilaterals of the input half-body data:

NQE is a right-adjusted integer number occupying columns 1-4. The reading of NQE is controlled by the read unit number IDS.

## ** Body Description Cards **

Each card contains the information to specify one half-body point. Each card contains

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-12 | XI | x-coordinate |
| 13-24 | YI | y-coordinate |
| 25-36 | ZI | z-coordinate |
| 37-40 | NI | N-station index (See Figure 4) |
| 41-44 | MI | M-station index (See Figure 4) |
| 45-48 | NS | Body number |

XI, YI, and ZI are single-precision floating-point numbers in F-fields. NI, MI, and NS are right-adjusted integer numbers. NS should be a constant for a given data set, which must be greater than zero but not equal to 1000. A blank card must be supplied at the end of these cards to signal the end of the body description cards. The reading of the body description cards are controlled by the read unit number IDS.

## ** Point Modification Cards **

1. Additional Input Point Change Information:

A single card contains the information to change one input point to the given values.

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-20 | XP | New X-value at (N,M) |
| 21-40 | YP | New Y-value at (N,M) |
| 41-60 | ZP | New Z-value at (N,M) |
| 61-65 | N | Reference N-station ($\leq$ max NI) |
| 66-70 | M | Reference M-station ($\leq$ max MI) |

XP, YP, and ZP are single-precision floating-point numbers in F-fields. N and M are right-adjusted integer numbers that denote the N and M station for the application of XP, YP, and ZP. A blank card must be supplied to serve either of two purposes. If no additional point change information is to be supplied, the blank card terminates the attempt to read more cards. If point change information is supplied, the blank card signals the end of this information.

2. Simple Averaging of Points or Linear Interpolation:

A single card contains the information to change one input point in a prescribed manner. The user may specify a two-point average, four-point average, two-point linear interpolation, or four-point linear interpolation.

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-5 | M1 | 1st reference M-station [$0 \le M1 \le max(MI)$] |
| 6-10 | M2 | 2nd reference M-station [$1 \le M2 \le max(MI)+1$] |
| 11-15 | N1 | 1st reference N-station [$0 \le N1 \le max(NI)$] |
| 16-20 | N2 | 2nd reference N-station [$1 \le N2 \le max(NI)+1$] |
| 21-25 | IMETH | Method: IMETH = 0 → simple average<br>IMETH = 1 → linear interpolation |

It should be noted that points are changed by the order in which the data cards are encountered. It must be true that

$$|M2 - M1| = 0 \text{ or } 2$$

and

$$|N2 - N1| = 0 \text{ or } 2 .$$

Consider a representative (plane) portion of the grid of the original input data below. Suppose it is desired the modify the original input point (e). Three schemes of each method are available (See Figure 5):

1. Two-point scheme of the points (a) and (c): M2 = (m+1), M1 = (m-1), N2 = (n+2), N1 = (n)
2. Two-point scheme of the points (b) and (d): M2 = (m+2), M1 = (m), N2 = (n+1), N1 = (n+1)
3. Four-point scheme of the points (a), (b), (c), and (d): M2 = (m+2), M1 = (m), N2 = (n+2), N1 = (n).

ORIGINAL PAGE IS
OF POOR QUALITY

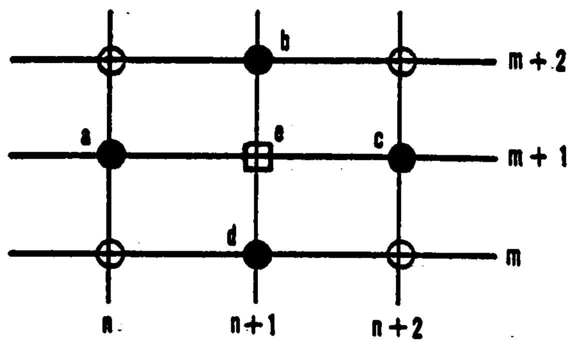



Figure 5: Illustration of points used in averaging and interpolating schemes

M2, M1, N2, and N1 are right-adjusted integer numbers. A blank card must be supplied to serve either of two purposes. If no averaging (or linear interpolation) information is to be supplied, the blank card terminates the attempt to read more cards. If averaging (or linear interpolation) information is supplied, the blank card signals the end of this information.

** Console Message LABEL Card **

The 80 characters of the array LABEL is used to provide information to the operator on the console typewriter about specific forms for plotting. Important: Only one LABEL card is permitted per execution.

** Plot Cards **

A single card contains all the necessary information for one plot. The available options and the necessary inptu for each are described in the succeeding sections. Reading of these cards are controlled by nonzero values of IPLOT1 and/or IPLOT2. Although the various plot cards, applicable to both the input and modified data, are presently discussed, only the plot card(s) pertaining to the plotting of the input data (i.e., if IPLOT1 $\neq$ 0) must be included at this time.

Orthographic projections. - For orthographic projections, the card should be set up as follows (See Figure 6):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1 | HORZ | "X", "Y", or "Z" for horizontal axis |
| 3 | VERT | "X", "Y", or "Z" for vertical axis |
| 5 to 7 | TEST1 | Word "OUT" for deletion of hidden lines; otherwise, leave blank |
| 8 to 12 | PHI | Roll angle, degrees (See Figure 7) |
| 13 to 17 | THETA | Pitch angle, degrees (See Figure 7) |
| 18 to 22 | PSI | Yaw angle, degrees (See Figure 7) |
| 48 to 52 | PLOTSZ | PLOTSZ determines the size of plot (scale factor is computed using PLOTSZ and maximum dimension of configuration) |
| 53 to 55 | TYPE | Word "ORT" |
| 72 | KODE | If KODE = 0, continue reading plot cards. If KODE = 1, after processing this plot, continue with the remaining non-plotting portion of the program |

An attempt is made to center the given configuration within the specified field. If the desired plot size is greater than 28 inches, centering is attempted within 28 inches so care must be taken in choosing the view. Minimum values are adjusted so that body axis lines with no rotation angles coincide with grid lines on the plotter paper. Therefore, the plotter pen should always be positioned exactly 1 inch from the side of the plotting space and on the intersection of heavy grid lines at the start of plotting.

Plan, front, and side views (stacked). - For plan, front, and side views, the card should be set up as follows (See Figure 8):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 8 to 12 | PHI | y-origin on paper of plan view, inches |
| 13 to 17 | THETA | y-origin on paper of side view, inches |
| 18 to 22 | PSI | y-origin on paper of front view, inches |

| Columns | FORTRAN Name | Description |
|---|---|---|
| 48 to 52 | PLOTSZ | PLOTSZ determines size of plot (a scale factor is computed using PLOTSZ and maximum dimension of configuration) |
| 53 to 55 | TYPE | Word "VU3" |
| 72 | KODE | If KODE = 0, continue reading plot cards<br>If KODE = 1, after processing this plot, continue with the remaining non-plotting portion of the program |

<u>Perspective views</u>. - For perspective views, the card should be set up as follows (See Figure 9):

| Columns | FORTRAN Name | Description |
|---|---|---|
| 8 to 12 | PHI | x of view point (location of viewer) in data coordinate system |
| 13 to 17 | THETA | y of view point in data coordinate system |
| 18 to 22 | PSI | z of view point in data coordinate system |
| 23 to 27 | XF | x of focal point (determines direction and focus) in data coordinate system |
| 28 to 32 | YF | y of focal point in data coordinate system |
| 33 to 37 | ZF | z of focal point in data coordinate system |
| 38 to 42 | DIST | Distance from eye to viewing plane, inches |
| 43 to 47 | FMAG | Viewing-plane magnification factor; it controls size of projected image |
| 48 to 52 | PLOTSZ | Diameter of viewing plane, inches; DIST and PLOTSZ together determine a cone which is field of vision; PLOTSZ value is also relative to type of viewer which is to be used. |
| 53 to 55 | TYPE | Word "PER" |
| 72 | KODE | If KODE = 0, continue reading plot cards<br>If KODE = 1, after processing this plot, continue with the remaining non-plotting portion of the program |

Stereo frames suitable for viewing in a stereoscope. - For stereo frames suitable for viewing in a stereoscope, the input is identical to that for the perspective views except that the word "STE" is used in columns 53 to 55. (See Figure 10).

IMPORTANT: If IPLOT1 ≠ 0, at least one plot card must be supplied. Similarly, if IPLOT2 ≠ 0, at least one plot card must be supplied.

** User-Specified Line-Augmentation Cards **

A single card contains the information to specify additional M-lines or N-lines between the "referenced" input M- or N-line.

| Columns | FORTRAN Name | Description |
|---|---|---|
| 1-4 | NAME | Line of reference (M or N or END) |
| 6-10 | K1 | Reference line number |
| 11-15 | NL | Number of additional lines |

NAME is a character array, occupying columns 1-4 in an A4 field. If NAME = M, the number NL of additional lines is supplied between M-line (K1) and M-line (K1+1). If NAME = N, the number NL of additional lines is supplied between N-line (K1) and N-line (K1+1). Cards of this type is continued until NAME = END is encountered. Since NAME is a character array, the specification of M, N, or END must be left-adjusted.

** Plot Cards for Modified Data **

The plot cards described earlier must be now included for the plotting of the modified body, if and only if IPLOT2 ≠ 0.

Specification of the cards above represent a complete set of data for a particular body. Additional data sets are programmed similarly starting again at CARD 2. A blank card should be supplied at the end of the last data set to terminate the program.

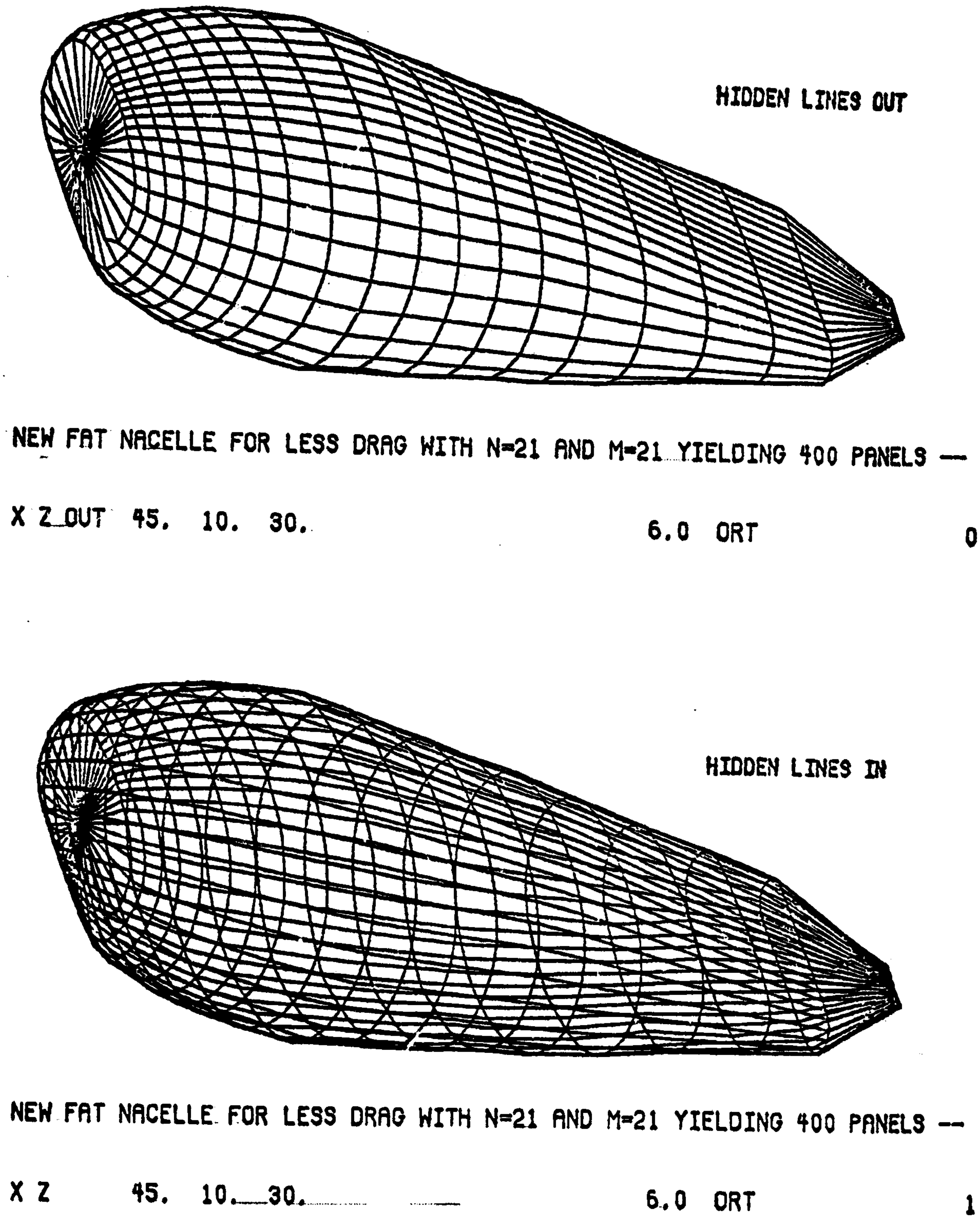



Figure 6: Example of orthographic projection

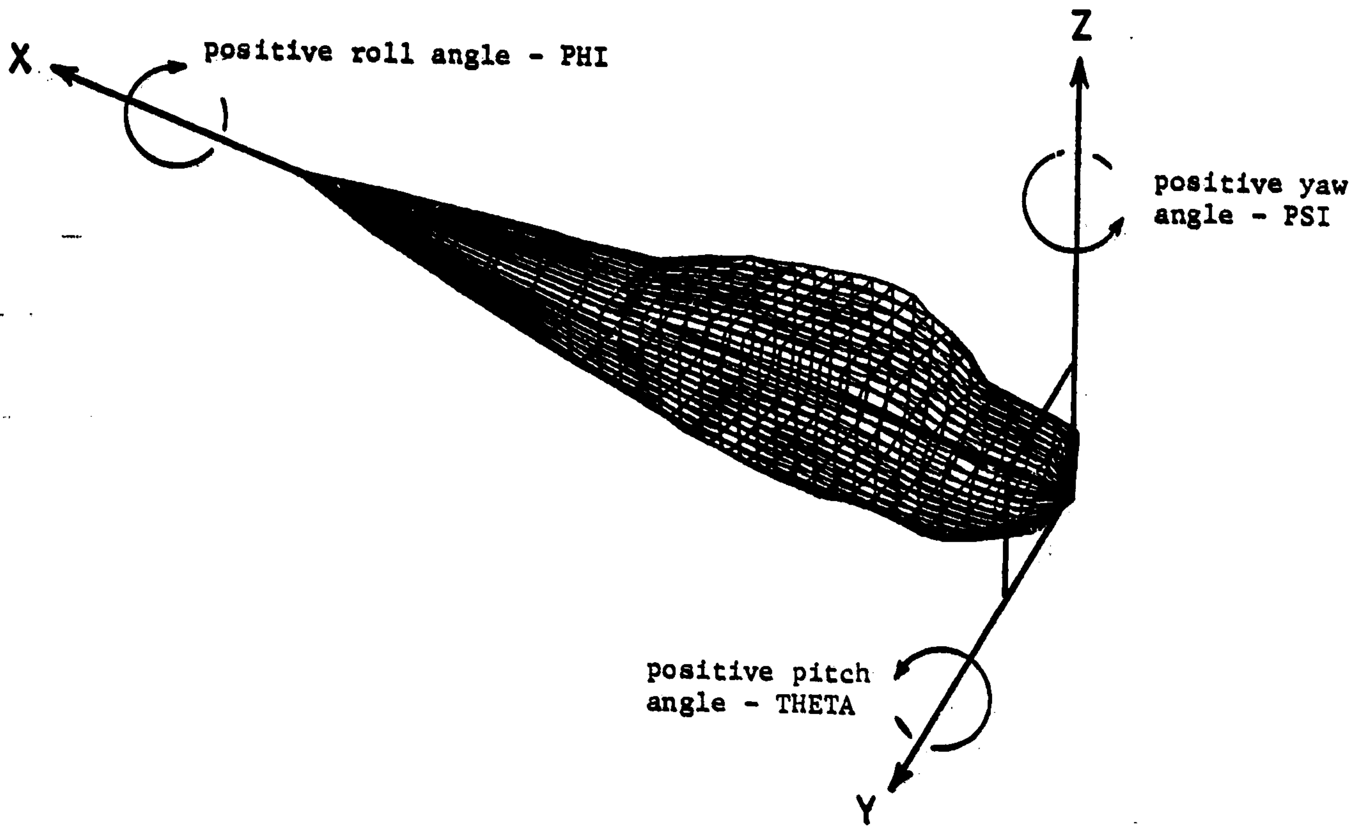



Figure 7: Orientation of body with respect to body reference axes for plotting angles

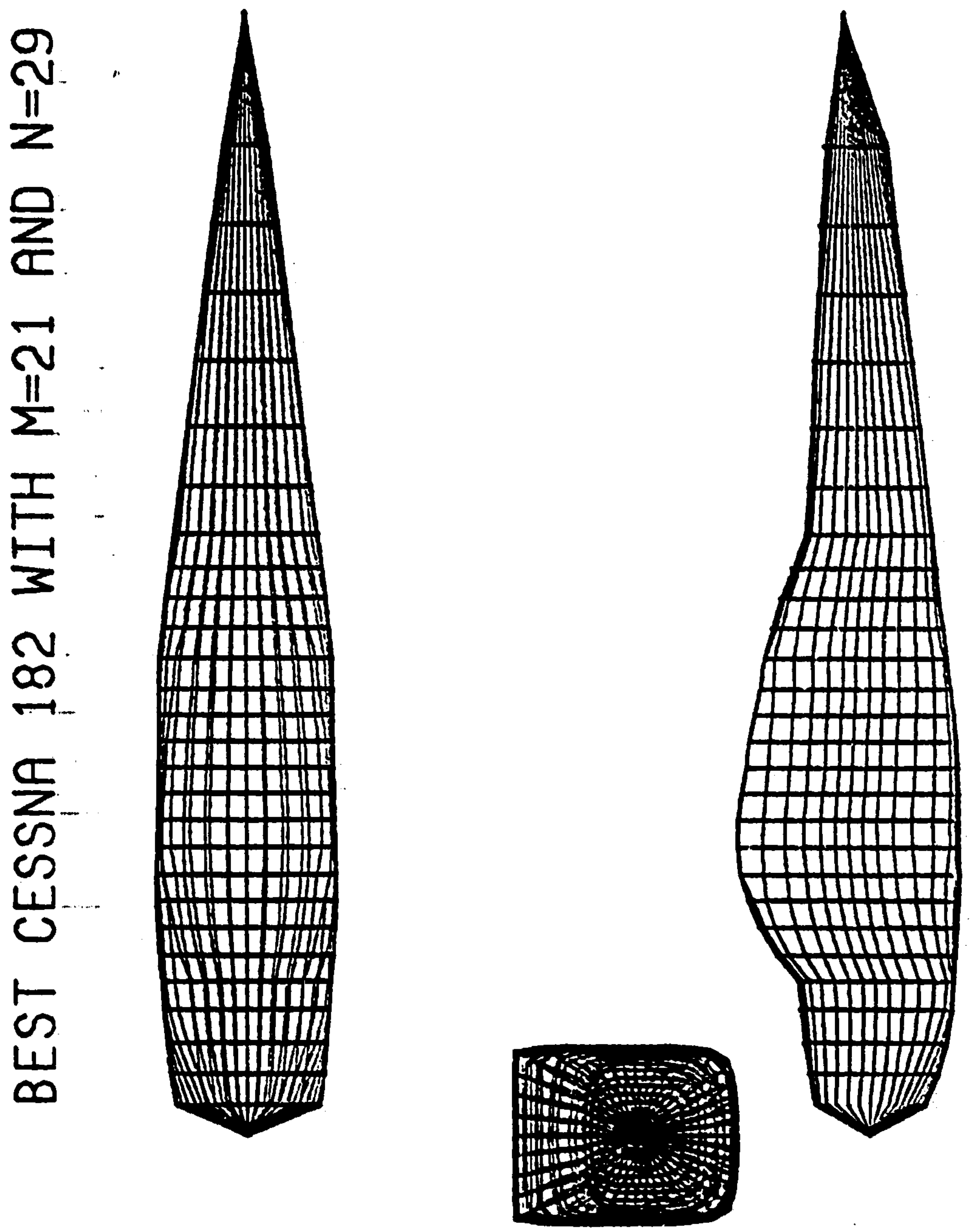
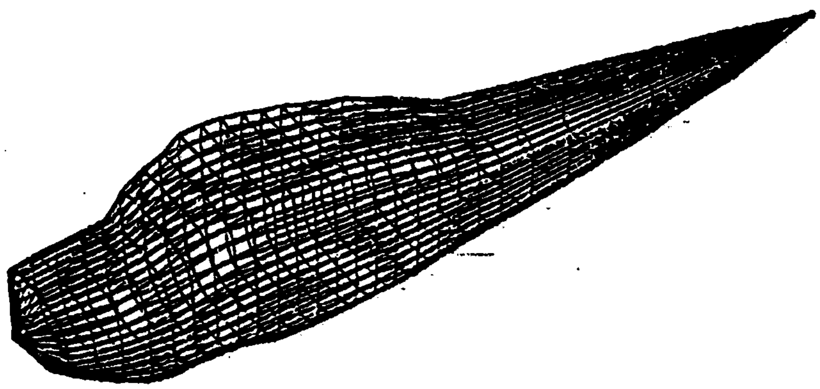



Figure 8: Example of 3-view plot



BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY

-20. -50. 50. 12. 0.0 0.0 14. 1.0 8.0 PER 1

Figure 9: Example of perspective view

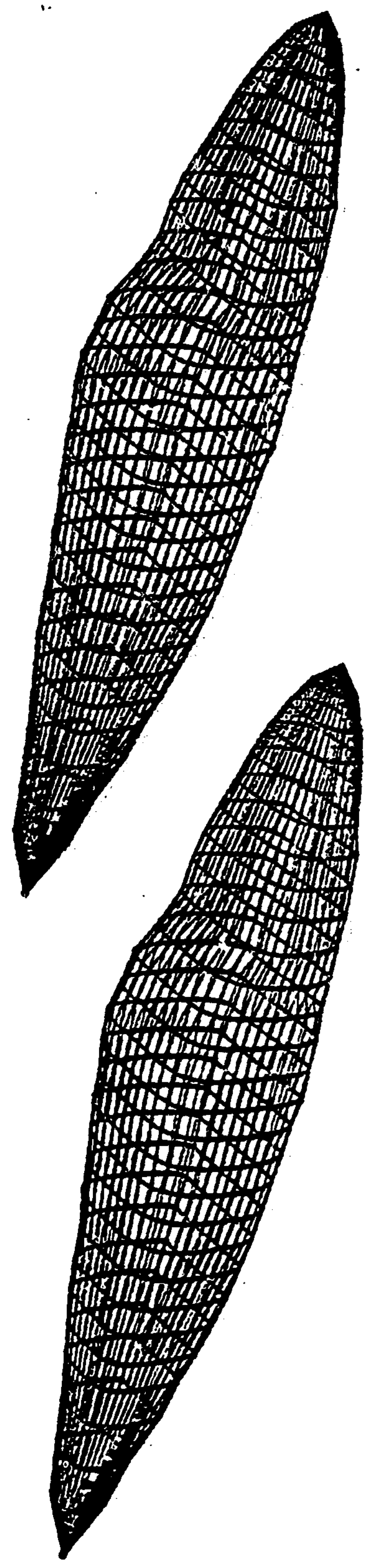


Figure 10: Example of stereo frames of stereoscopic view

# PROGRAM LISTING - GRIDPLOT

```
C       PROGRAMMER:           STAN R. FOX
C                             2406 BROUGHTON BDLG
C                             DEPT ME ENGRG. NCSU
C                             RALEIGH, N.C.  27650
C                             PHONE: 919/737-2374
C
C
C       FOR A COMPLETE DESCRIPTION OF NECESSARY INPUT AND THE AVAILABLE
C       OPTIONS. CONSULT THF USFR'S INSTRUCTIONS.
C
C
        DIMENSION P(29,21,3),X(29),Y(29),Z(29),XN(89),YN(89),Q(29,81,3),R(
       189,81,3),XFUS(89),SFUS(89,81,2),BS(2,81,3),C(4,29),D(29),DIAG(29),
       1MP(21,2),NP(29,2)
        DIMENSION XAME1(4),XAME2(4),LABEL(80)
        COMMCN /INOUT/JREAD,JWRITE,JPUNCH,KFILE1,KFILE2,KFILE3,KFILE4,IWRI
       1TE
        COMMCN /TRANS/TITLE(20)
        COMMON /CODE/ICODE,IBP1,IBP2
        COMMCN /PCODE/LCODE
        DATA XAME1/4HINPU,4HT PL,4HOT D,4HATA /,XAME2/4HNEW ,4HPLOT,4H DAT
       1,4HA    /
C
        JREAD=1
        JPUNCH=2
        JWRITE=3
        KFILE1=4
        KFILE2=5
        KFILE3=6
        KFILE4=7
        NCOUNT=1
C***    PEAD DATA CARDS
        READ (JREAD,1) N,M,NADD,MADD,PXL,PYL
      1 FORMAT (4I5,2F10.0)
        J1=M
        J2=N
        N1=N+IABS(NADD)
        M1=M+IABS(MADD)
        J3=MAX0(N1,M1)
        J4=MAX0(N,M)
      2 READ (JREAD,3) IDS,INPTM,INPTN,IPUNCH,IPLOT1,IPLOT2,LPCH1,LPCH2,IW
       1RITE,CF
```

```
    3 FORMAT (9I5,5X,F10.5)
      ICAN=0
      IF (CF.EQ.0.0E0)CF=1.0E0
      IF ((IDS.GT.0.AND.IDS.NE.JPUNCH).AND.(IDS.NE.JWRITE.AND.IDS.NE.KFI
     1L1).AND.(IDS.NE.KFILE2.AND.IDS.NE.KFILE3).AND.IDS.NE.KFILE4) GO TO
     1 5
      IF (IDS.LE.0) GO TO 97
      WRITE (JWRITE,4)
    4 FORMAT (1X,///,3X,28HERROR IN IDS ... TERMINATING,//)
      GO TO 97
    5 INPTN=IABS(INPTN)
      INPTM=IABS(INPTM)
      INPTNS=INPTN
      INPTMS=INPTM
      READ (IDS,6) (TITLE(I),I=1,20)
    6 FORMAT (20A4)
      WRITE (JWRITE,7) (TITLE(I),I=1,20)
    7 FORMAT (1H1,//,5X,8HINPUT***,//,10X,20A4,/)
      WRITE (JWRITE,8) IDS,INPTM,INPTN,IPUNCH,IPLOT1,IPLOT2,LPCH1,LPCH2,
     1IWRITE,CF
    8 FORMAT (1X,/,10X,4HIDS=,I3,3X,6HINPTM=,I3,4X,6HINPTN=,I3,4X,7HIPUN
     1CH=,I3,4X,7HIPLOT1=,I3,3X,7HIPLOT2=,I3,/,10X,6HLPCH1=,I3,2X,6HLPCH
     12=,I3,3X,7HIWRITE=,I3,3X,3HCF=,E16.9,/)
      READ (IDS,9) NQE
    9 FORMAT (I4)
      WRITE (JWRITE,10) NQE
   10 FORMAT (10X,25HNUMBER OF QUADRILATERALS,,I4,//)
      IF (IWRITE.EQ.0) WRITE (JWRITE,11)
   11 FORMAT (15X,2HXI,12X,2HYI,12X,2HZI,8X,2HNI,5X,2HMI,5X,2HNS,/)
      READ (IDS,12) XI,YI,ZI,NI,MI,NS
   12 FORMAT (3F12.8,3I4)
      MMAX=MI
      MMIN=MI
      NMAX=NI
      NMIN=NI
      NSS=NS
      L=1
      GO TO 14
   13 READ (IDS,12) XI,YI,ZI,NI,MI,NS
      IF (NS.NE.NSS) GO TO 16
      L=L+1
   14 IF (IWRITE.EQ.0) WRITE (JWRITE,15) XI,YI,ZI,NI,MI,NS
```

```
   15 FORMAT (10X,3(F12.8,2X),3(I5,2X))
      P(NI,MI,1)=XI*CF
      P(NI,MI,2)=YI*CF
      P(NI,MI,3)=ZI*CF
      MMAX=MAX0(MMAX,MI)
      MMIN=MIN0(MMIN,MI)
      NMAX=MAX0(NMAX,NI)
      NMIN=MIN0(NMIN,NI)
      GO TO 13
   16 NQ=(MMAX-MMIN)*(NMAX-NMIN)
      IF (NQ.EQ.NQE) GO TO 18
      WRITE (JWRITE,17) NQ,NQE
   17 FORMAT (1X,///,5X,30HNUMBER OF QUADRILATERALS READ(,I4,56H) DOES N
     1OT EQUAL THE NUMBER OF SPECIFIED QUADRILATERALS(,I4,1H))
      ICAN=1
      GO TO 96
C***  INPUT READING COMPLETED, CHECK FOR ERRORS
   18 IF (((MMAX.EQ.(2*(MMAX/2)).OR.MMAX.LT.4).OR.(MMIN.EQ.(2*(MMIN/2)).
     1OR.MMIN.NE.1)).OR.((NMAX.EQ.(2*(NMAX/2)).OR.NMAX.LT.4).OR.(NMIN.EQ
     1.(2*(NMIN/2)).OR.NMIN.NE.1))) GO TO 19
      GO TO 21
   19 WRITE (JWRITE,20) MMAX,MMIN,NMAX,NMIN
   20 FORMAT (1X,//,5X,23HERROR DETECTED IN INPUT,3X,6HMMAX =,I3,3X,6HMM
     1IN =,I3,/,31X,6HNMAX =,I3,3X,6HNMIN =,I3)
      ICAN=1
      GO TO 96
C***  ENACT POINT MODIFICATIONS
   21 CALL PNT1(J1,J2,P,JREAD,JWRITE,CF)
      CALL PNT2(J1,J2,P,JREAD,JWRITE,MMAX,NMAX)
      IF (NCOUNT.NE.1) GO TO 23
      IF (IPLOT1.EQ.0.AND.IPLOT2.EQ.0) GO TO 23
      CALL PICSIZ(PXL,PYL)
      READ (JREAD,22) (LABEL(NN),NN=1,80)
   22 FORMAT (80A1)
      CALL PENMSG(LABEL)
   23 IF (IPLOT1.EQ.0.AND.LPCH1.EQ.0) GO TO 38
      LCODE=1
C***  CONVERT TO PLOT AXIS SYSTEM
      ZMAX=-1000.0
      ZMIN=1000.0
      DO 24 NN=1,NMAX
      XFUS(NN)=P(1,1,1)-P(NN,1,1)
```

```
      DO 24 MM=1,MMAX
      IF (MM.GT.1.AND.MM.LT.MMAX) GO TO 24
      IF (ZMAX.LT.P(NN,MM,3))ZMAX=P(NN,MM,3)
      IF (ZMIN.GT.P(NN,MM,3))ZMIN=P(NN,MM,3)
   24 CONTINUE
      ZORG=ABS(ZMIN)+ABS(ZMAX)
      DO 25 NN=1,NMAX
      DC 25 MM=1,MMAX
   25 P(NN,MM,3)=P(NN,MM,3)+ZORG
C***  INPUT DATA
      IF (LPCH1.FQ.0) GC TC 33
      WRITE (JPUNCH,26) (XAME1(I),I=1,4)
   26 FORMAT (4A4)
      WRITE (JPUNCH,27) (XFUS(NN),NN=1,NMAX)
   27 FORMAT (10F7.4)
      DO 28 NN=1,NMAX
      WRITE (JPUNCH,27) (P(NN,MM,2),MM=1,MMAX)
   28 WRITE (JPUNCH,27) (P(NN,MM,3),MM=1,MMAX)
      IF (IWRITE.GT.1) GO TO 33
      WRITE (JWRITE,29)
   29 FORMAT (1X,///,10X,16HINPUT PLCT DATA:,//)
      WRITE (JWRITE,30) (XFUS(NN),NN=1,NMAX)
   30 FORMAT (10X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,1
     1X,F7.4,1X,F7.4,1X,F7.4)
      WRITE (JWRITE,31)
   31 FORMAT (1X)
      DO 32 NN=1,NMAX
      WRITE (JWRITE,30) (P(NN,MM,2),MM=1,MMAX)
      WRITE (JWRITE,31)
      WRITE (JWRITE,30) (P(NN,MM,3),MM=1,MMAX)
   32 WRITE (JWRITE,31)
   33 IF (IPLOT1.EQ.0) GO TO 36
      REWIND KFILE1
      WRITE (KFILE1)(XFUS(NN),NN=1,NMAX)
      DO 34 NN=1,NMAX
      WRITE (KFILE1)(P(NN,MM,2),MM=1,MMAX)
   34 WRITE (KFILE1)(P(NN,MM,3),MM=1,MMAX)
      WRITE (JWRITE,35)
   35 FORMAT (1X,//,1X,127('-'))
      CALL XYZPLT(MMAX,NMAX,XFUS,SFUS)
   36 DO 37 NN=1,NMAX
      DO 37 MM=1,MMAX
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   37 P(NN,MM,3)=P(NN,MM,3)-ZORG                                       GRD 206
   38 IF (INPTM.LT.0.AND.INPTN.LT.0.AND.IPLOT2.EQ.0.AND.LPCH2.EQ.0) GO T GRD 207
     1O 96                                                              GRD 208
      CALL TEST(MAC1,MAC2,MP,NP,J1,J2,KM1,KM2,KN1,KN2,MMAX,NMAX)        GRD 209
      KN=NMAX                                                           GRD 210
      IF ((INPTM.EQ.0.AND.INPTN.EQ.0).AND.(MAC1.EQ.0.AND.MAC2.EQ.0)) GO GRD 211
     1TO 79                                                             GRD 212
C***  FOR EACH N-STATION, CURVE FIT THE M-STATIONS                      GRD 213
      IBP1=0                                                            GRD 214
      IBP2=0                                                            GRD 215
      K=0                                                               GRD 216
      DO 54 N=1,NMAX                                                    GRD 217
      IF (N.EQ.1.OR.N.EQ.NMAX) GO TO 39                                 GRD 218
      GO TO 46                                                          GRD 219
   39 IF (MAC1.NE.0) GO TO 40                                           GRD 220
      NPTS=MMAX+(MMAX-1)*INPTM                                          GRD 221
      GO TO 41                                                          GRD 222
   40 NPTS=MAC1                                                         GRD 223
C***  FOR FIRST AND LAST N-STATIONS                                     GRD 224
   41 DO 42 M=1,NPTS                                                    GRD 225
   42 Q(N,M,1)=P(N,1,1)                                                 GRD 226
      IQ=1                                                              GRD 227
      IP=1                                                              GRD 228
      Q(N,IQ,2)=P(N,IP,2)                                               GRD 229
      Q(N,IQ,3)=P(N,IP,3)                                               GRD 230
   43 IQMX=IQ+1+INPTM                                                   GRD 231
      IP=IP+1                                                           GRD 232
      IF (IP.GT.MMAX) GO TO 54                                          GRD 233
      IF (MAC1.EQ.0) GO TO 44                                           GRD 234
      ICODE=0                                                           GRD 235
      IF ((IP-1).GE.KM1)ICODE=1                                         GRD 236
      IF (ICODE.EQ.0)INPTM=MP(IP-1,1)                                   GRD 237
      IF (ICODE.EQ.1)INPTM=MP(IP-KM1,2)                                 GRD 238
      IF (ICODE.EQ.0)IBP1=IBP1+INPTM                                    GRD 239
      IF (ICODE.EQ.1)IBP2=IBP2+INPTM                                    GRD 240
      IQMX=IQ+1+INPTM                                                   GRD 241
   44 X1=(P(N,IP,2)-P(N,IP-1,2))/FLOAT(INPTM+1)                         GRD 242
      X2=(P(N,IP,3)-P(N,IP-1,3))/FLOAT(INPTM+1)                         GRD 243
      IQP1=IQ+1                                                         GRD 244
      DO 45 ML=IQP1,IQMX                                                GRD 245
      Q(N,ML,2)=Q(N,ML-1,2)+X1                                          GRD 246
   45 Q(N,ML,3)=Q(N,ML-1,3)+X2                                          GRD 247
```

```
      IQ=IQMX
      GO TO 43
   46 MID=(MMAX+1)/2
      IF (MMAX.LT.7)MID=MMAX
      KNS=MID+(MID-1)*INPTMS
      SLP1=P(N,1,2)/P(N,1,3)
      SLP2=(P(N,MID,2)-P(N,1,2))/(P(N,MID,3)-P(N,1,3))
      A=ATAN((SLP2-SLP1)/(1.0+SLP1*SLP2))
      M1=1
      M2=MID
      ICODE=0
      MADD=0
      MD=0
      Y0=P(N,MID,2)
      Z0=P(N,MID,3)
   47 DO 48 M=M1,M2
      Z(M+MADD)=(P(N,M,3)-Z0)*COS(A)+(P(N,M,2)-Y0)*SIN(A)
      Y(M+MADD)=(P(N,M,2)-Y0)*COS(A)-(P(N,M,3)-Z0)*SIN(A)
   48 CONTINUE
      IF (MAC1.EQ.0) GO TO 49
      KNS=KM2+IBP2
      IF (ICODE.EQ.0)KNS=KM1+IBP1
   49 CALL PCS(MID,Y,Z,XN,YN,INPTM,KN,C,D,DIAG,JWRITE,J4,J3,MAC1,MP,J1,K
     1)
      IF (KN.NE.KNS) GO TO 50
      GO TO 52
   50 WRITE (JWRITE,51) N,ICODE,KN,KNS
   51 FORMAT (1X,52HPCS DID NOT FIND CORRECT NO. OF POINTS AT N-STATION
     1,I3,12H WITH ICODE=,I2,/,1X,6HFOUND ,I3,18H POINTS(SHOULD BE ,I3,1
     1H))
      GO TO 96
   52 DO 53 M=1,KN
      Q(N,M+MD,1)=P(N,1,1)
      Q(N,M+MD,2)=XN(M)*SIN(A)+YN(M)*COS(A)+Y0
      Q(N,M+MD,3)=XN(M)*COS(A)-YN(M)*SIN(A)+Z0
   53 CONTINUE
      IF (ICODE.NE.0.OR.MID.EQ.MMAX) GO TO 54
      SLP1=P(N,MMAX,2)/P(N,MMAX,3)
      SLP2=(P(N,MMAX,2)-P(N,MID,2))/(P(N,MMAX,3)-P(N,MID,3))
      A=ATAN((SLP2-SLP1)/(1.0+SLP1*SLP2))
      ICODE=1
      M1=MID
```

```
      M2=MMAX
      MADD=1-MID
      MD=KN-1
      GO TO 47
   54 INPTM=INPTMS
C***  SET INITIAL AND FINAL VALUES
      DO 55 LL=1,2
      N=1
      IF (LL.EQ.2)N=NMAX
      DO 55 M=1,NPTS
      DO 55 JJ=1,3
   55 BS(LL,M,JJ)=Q(N,M,JJ)
      MMAX=NPTS
C***  FOR EACH M-STATION, CURVE FIT EACH SET OF N-STATIONS
      MID=(NMAX+1)/2
      IF (NMAX.LT.7)MID=NMAX
      KNS=MID+(MID-1)*INPTN
      KS=NMAX+(NMAX-1)*INPTN
      IF (MAC2.EQ.0) GO TO 57
      K=1
      IBP1=0
      IBP2=0
      JD2=NMAX
      IF (NMAX.GE.7)JD2=(NMAX+1)/2
      DO 56 J=1,NMAX
      IF (J.LT.JD2)IBP1=IBP1+NP(J,1)
      IF (J.GE.JD2)IBP2=IBP2+NP(J-JD2+1,2)
   56 CONTINUE
   57 DO 64 M=1,MMAX
      X0=Q(MID,M,1)
      Y0=Q(MID,M,2)
      Z0=Q(MID,M,3)
      SAX=0.0
      ICODE=0
      M1=1
      M2=MID
      MADD=0
      MD=0
   58 SLP1=(Q(M1,M,2)-Q(M2,M,2))/(Q(M1,M,1)-Q(M2,M,1))
      SLP2=(Q(M1,M,3)-Q(M2,M,3))/(Q(M1,M,1)-Q(M2,M,1))
      ALP1=ATAN((SLP1-SAX)/(1.0+SAX*SLP1))
      ALP2=ATAN((SLP2-SAX)/(1.0+SAX*SLP2))
```

```
                                                                          GRD 332
      DO 59 N=M1,M2                                                       GRD 333
      IP=M2+M1-N                                                          GRD 334
      X(N+MADD)=(Q(IP,M,1)-X0)*COS(ALP1)+(Q(IP,M,2)-Y0)*SIN(ALP1)         GRD 335
      Y(N+MADD)=(Q(IP,M,2)-Y0)*COS(ALP1)-(Q(IP,M,1)-X0)*SIN(ALP1)         GRD 336
   59 CONTINUE                                                            GRD 337
      IF (MAC2.EQ.0) GO TO 60                                             GRD 338
      KNS=KN2+IBP2                                                        GRD 339
      IF (ICODE.EQ.0)KNS=KN1+IBP1                                         GRD 340
   60 CALL PCS(MID,Y,X,XN,YN,INPTN,KN,C,D,DIAG,JWRITE,J4,J3,MAC2,NP,J2,K  GRD 341
     1)                                                                   GRD 342
      IF (KN.NE.KNS) GO TO 77                                             GRD 343
      DO 61 N=1,KN                                                        GRD 344
      IP=KN+1-N+MD                                                        GRD 345
      R(IP,M,1)=XN(N)*COS(ALP1)-YN(N)*SIN(ALP1)+X0                        GRD 346
   61 R(IP,M,2)=XN(N)*SIN(ALP1)+YN(N)*COS(ALP1)+Y0                        GRD 347
      DO 62 N=M1,M2                                                       GRD 348
      IP=M2+M1-N                                                          GRD 349
      X(N+MADD)=(Q(IP,M,1)-X0)*COS(ALP2)+(Q(IP,M,3)-Z0)*SIN(ALP2)         GRD 350
      Z(N+MADD)=(Q(IP,M,3)-Z0)*COS(ALP2)-(Q(IP,M,1)-X0)*SIN(ALP2)         GRD 351
   62 CONTINUE                                                            GRD 352
      CALL PCS(MID,Z,X,XN,YN,INPTN,KN,C,D,DIAG,JWRITE,J4,J3,MAC2,NP,J2,K  GRD 353
     1)                                                                   GRD 354
      IF (KN.NE.KNS) GO TO 77                                             GRD 355
      DO 63 N=1,KN                                                        GRD 356
      IP=KN+1-N+MD                                                        GRD 357
      R(IP,M,3)=XN(N)*SIN(ALP2)+YN(N)*COS(ALP2)+Z0                        GRD 358
   63 CONTINUE                                                            GRD 359
      IF (ICODE.EQ.1.OR.MID.EQ.NMAX) GO TO 64                             GRD 360
      ICODE=1                                                             GRD 361
      M1=MID                                                              GRD 362
      M2=NMAX                                                             GRD 363
      MADD=1-MID                                                          GRD 364
      MD=KN-1                                                             GRD 365
      GO TO 58                                                            GRD 366
   64 CONTINUE                                                            GRD 367
      KN=KS                                                               GRD 368
      IF (MAC2.EQ.0) GO TO 65                                             GRD 369
      KN=MAC2                                                             GRD 370
   65 DO 67 LL=1,2                                                        GRD 371
      N=1                                                                 GRD 372
      IF (LL.EQ.2)N=KN                                                    GRD 373
      DO 66 M=1,MMAX
```

```
      DO 66 JJ=1,3
   66 R(N,M,JJ)=BS(LL,M,JJ)
   67 CONTINUE
      DO 76 N=1,KN
      X0=0.0
      DO 68 M=1,MMAX
      X0=X0+R(N,M,1)
   68 CONTINUE
      X0=X0/FLOAT(MMAX)
      IF (N.EQ.1) GO TO 70
      IF (X0.LT.R(N-1,1,1)) GO TO 70
      X0=0.0
      LK=0
      DO 69 M=1,MMAX
      IF (R(N,M,1).GT.R(N-1,1,1)) GO TO 69
      LK=LK+1
      X0=X0+R(N,M,1)
   69 CONTINUE
      IF (LK.EQ.0)ICAN=1
      IF (ICAN.EQ.1) GO TO 72
      X0=X0/FLOAT(LK)
   70 DO 71 M=1,MMAX
   71 R(N,M,1)=X0
      IF (N.EQ.1) GO TO 76
      IF (X0.GT.R(N-1,1,1)) GO TO 72
      GO TO 76
   72 NM1=N-1
      IF (ICAN.EQ.1) GO TO 74
      WRITE (JWRITE,73) N,NM1
   73 FORMAT (1X,///,10X,29HAVERAGE X-VALUE AT N-STATION ,I2,46H IS GREA
     1TER THAN AVERAGE X-VALUE AT N-STATION ,I2,38H.  MUST BE LESS THAN
     1OR EXACTLY EQUAL.,//)
      GO TO 96
   74 WRITE (JWRITE,75) N,NM1
   75 FORMAT (1X,///,10X,35HALL COMPUTED X-VALUES AT N-STATION ,I2,51H A
     1RE GREATER THAN THE AVERAGE X-VALUE AT N-STATION ,I2,//)
      GO TO 96
   76 CONTINUE
      GO TO 81
   77 WRITE (JWRITE,78) M,ICODE,KN,KNS
   78 FORMAT (1X,52HPCS DID NOT FIND CORRECT NO. OF POINTS AT M-STATION
     1,I3,12H WITH ICODE=,I2,/,1X,6HFOUND ,I3,18H POINTS(SHOULD BE ,I3,1
```

```
     1H))
      ICAN=1
      GO TO 96
   79 DO 80 N=1,KN
      DO 80 M=1,MMAX
      DO 80 J=1,3
   80 R(N,M,J)=P(N,M,J)
C***  NOW PRINT RESULTS
   81 NNQE=(MMAX-1)*(KN-1)
      IF (IWRITE.NE.2) WRITE (JWRITE,82) MMAX,KN,NNQE
   82 FORMAT (1H1,//,5X,11HNEW DATA***,//,5X,6HMMAX =,I3,3X,6HNMAX =,I3,
     12X,5H---->,I5,15H QUADRILATERALS,//,15X,2HXI,12X,2HYI,12X,2HZI,8X,
     12HNI,5X,2HMI,5X,2HNS,/)
      IF (IPUNCH.NE.0) WRITE (JPUNCH,83) NNQE
   83 FORMAT (I4)
      NS=1
      DO 85 N=1,KN
      DO 85 M=1,MMAX
      DO 84 J=1,3
   84 IF (ABS(R(N,M,J)).LT.1.0E-04)R(N,M,J)=0.0E0
      IF (IWRITE.NE.2) WRITE (JWRITE,15) (R(N,M,J),J=1,3),N,M,NS
      IF (IPUNCH.NE.0) WRITE (JPUNCH,12) (R(N,M,J),J=1,3),N,M,NS
   85 CONTINUE
      IF (IPLOT2.EQ.0.AND.LPCH2.EQ.0) GO TO 96
      LCODE=2
C***  CONVERT TO PLOT AXIS SYSTEM
      ZMAX=-1000.0
      ZMIN=1000.0
      DO 87 NN=1,KN
      XFUS(NN)=R(1,1,1)-R(NN,1,1)
      DO 86 MM=1,MMAX
      IF (MM.GT.1.AND.MM.LT.MMAX) GO TO 86
      IF (ZMAX.LT.R(NN,MM,3))ZMAX=R(NN,MM,3)
      IF (ZMIN.GT.R(NN,MM,3))ZMIN=R(NN,MM,3)
   86 CONTINUE
   87 CONTINUE
      ZORG=ABS(ZMIN)+ABS(ZMAX)
      DO 88 NN=1,KN
      DO 88 MM=1,MMAX
   88 R(NN,MM,3)=R(NN,MM,3)+ZORG
C***  NEW DATA
      IF (LPCH2.EQ.0) GO TO 92
```

```
      WRITE (JPUNCH,26) (XAME2(I),I=1,4)
      WRITE (JPUNCH,27) (XFUS(NN),NN=1,KN)
      DO 89 NN=1,KN
      WRITE (JPUNCH,27) (R(NN,MM,2),MM=1,MMAX)
   89 WRITE (JPUNCH,27) (R(NN,MM,3),MM=1,MMAX)
      IF (IWRITE.GT.1) GO TO 92
      WRITE (JWRITE,90)
   90 FORMAT (1X,///,10X,14HNEW PLOT DATA:,//)
      WRITE (JWRITE,30) (XFUS(NN),NN=1,KN)
      WRITE (JWRITE,31)
      DO 91 NN=1,KN
      WRITE (JWRITE,30) (R(NN,MM,2),MM=1,MMAX)
      WRITE (JWRITE,31)
      WRITE (JWRITE,30) (R(NN,MM,3),MM=1,MMAX)
   91 WRITE (JWRITE,31)
   92 IF (IPLOT2.EQ.0) GO TO 94
      REWIND KFILE4
      WRITE (KFILE4)(XFUS(NN),NN=1,KN)
      DO 93 NN=1,KN
      WRITE (KFILE4)(R(NN,MM,2),MM=1,MMAX)
   93 WRITE (KFILE4)(R(NN,MM,3),MM=1,MMAX)
      WRITE (JWRITE,35)
      CALL XYZPLT(MMAX,KN,XFUS,SFUS)
C***  RESET Z-VALUES
   94 DO 95 NN=1,KN
      DO 95 MM=1,MMAX
   95 R(NN,MM,3)=R(NN,MM,3)-ZORG
   96 IF (IPLOT1.NE.0.OR.IPLOT2.NE.0)NCOUNT=NCOUNT+1
      IF (IDS.NE.JREAD) REWIND IDS
      GO TO 2
      NCOUNT=NCOUNT+1
   97 IF (NCOUNT.GT.1) CALL PICSIZ(0.0,0.0)
      STOP
      END

      SUBROUTINE PNT1(J2,J1,R,JREAD,JWRITE,CF)
C
C***  POINT MODIFICATION BY ADDITIONAL INPUT POINT CHANGE INFORMATION
C
      DIMENSION R(J1,J2,3)
```

```
C
      K=0
    1 READ (JREAD,2) XP,YP,ZP,N,M
    2 FORMAT (3F20.0,2I5)
      IF (N.LE.0.OR.M.LE.0) GO TO 5
      K=K+1
      IF (K.EQ.1) WRITE (JWRITE,3)
    3 FORMAT (1X,///,5X,63HPOINT MODIFICATION BY ADDITIONAL INPUT POINT
     1CHANGE INFORMATION,//,15X,2HXI,12X,2HYI,12X,2HZI,8X,2HNI,5X,2HMI,/
     1)
      WRITE (JWRITE,4) XP,YP,ZP,N,M
    4 FORMAT (10X,3(F12.8,2X),2(I5,2X))
      R(N,M,1)=XP*CF
      R(N,M,2)=YP*CF
      R(N,M,3)=ZP*CF
      GO TO 1
    5 RETURN
      END


      SUBROUTINE PNT2(J2,J1,R,JREAD,JWRITE,MMAX,NMAX)
C
C***  POINT MODIFICATION BY SIMPLE AVERAGE OR LINEAR INTERPOLATION
C
      DIMENSION R(J1,J2,3)
C
      K=0
      ICODE1=0
      ICODE2=0
    1 READ (JREAD,2) M1,M2,N1,N2,IMETH
    2 FORMAT (5I5)
      IF ((M1.LE.0.AND.M2.LE.0).AND.(N1.LE.0.AND.N2.LE.0)) GO TO 31
      IF ((M1.LE.0.AND.M2.LE.0).OR.(N1.LE.0.OR.N2.LE.0)) GO TO 1
      IF ((M1.LE.0.OR.M2.LE.0).OR.(N1.LE.0.OR.N2.LE.0)) 
      K=K+1
      IF (K.EQ.1) WRITE (JWRITE,3)
    3 FORMAT (1X,///,5X,38HPOINT MODIFICATION BY SPECIFIED METHOD)
      JS=M2
      IF (M1.LE.M2) GO TO 4
      M2=M1
      M1=JS
    4 JS=N2
```

```
      IF (N1.LE.N2) GO TO 5
      N2=N1
      N1=JS
    5 L1=M2-M1
      L2=N2-N1
      IF (M2-MMAX-1)  6,6,1
    6 IF (1-M1)  7,7,1
    7 IF (N2-NMAX-1)  8,8,1
    8 IF (1-N1)  9,9,1
    9 IF (((L1.EQ.1.OR.L1.GT.2).OR.(L2.EQ.1.OR.L2.GT.2)).OR.(L1.EQ.0.AND
     1.L2.EQ.0)) GO TO 1
      IF (IMETH.LT.0)IMETH=0
      IF (IMETH.GT.1)IMETH=1
      AYN=0.0E0
      AZN=0.0E0
      AYM=0.0E0
      AZM=0.0E0
      IM=IMETH+1
      IF (L1.NE.0.AND.L2.NE.0) GO TO 19
      IF (L1.EQ.0.AND.L2.NE.0) GO TO 13
      IF (M2.GT.MMAX) GO TO 10
      IF (M1.LT.1) GO TO 11
      AYM=(R(N1,M1,2)+R(N1,M2,2))/2.0E0
      AZM=(R(N1,M1,3)+R(N1,M2,3))/2.0E0
      GO TO 12
   10 AYM=R(N1,M1+1,2)
      AZM=R(N1,M1,3)
      GO TO 12
   11 AYM=R(N1,M2-1,2)
      AZM=R(N1,M2,3)
   12 N=N1
      M=M1+1
      D=1.0E0
      GO TO 28
   13 GO TO (14,15),IM
   14 AYN=(R(N1,M1,2)+R(N2,M1,2))/2.0E0
      AZN=(R(N1,M1,3)+R(N2,M1,3))/2.0E0
      GO TO 18
   15 IF (N2.GT.NMAX) GO TO 16
      IF (N1.LT.1) GO TO 17
      A=(R(N1,M1,2)-R(N2,M1,2))/(R(N1,M1,1)-R(N2,M1,1))
      B=R(N1,M1,2)-A*R(N1,M1,1)
```

```
      AYN=A*R(N1+1,M1,1)+B
      A=(R(N1,M1,3)-R(N2,M1,3))/(R(N1,M1,1)-R(N2,M1,1))
      B=R(N1,M1,3)-A*R(N1,M1,1)
      AZN=A*R(N1+1,M1,1)+B
      GO TO 18
   16 AYN=R(N1,M1,2)
      AZN=R(N1,M1,3)
      ICODE1=1
      GO TO 18
   17 AYN=R(N2,M1,2)
      AZN=R(N2,M1,3)
      ICODE2=1
   18 N=N1+1
      M=M1
      D=1.0E0
      GO TO 28
   19 IF (M2.GT.MMAX) GO TO 20
      IF (M1.LT.1) GO TO 21
      AYM=(R(N1+1,M1,2)+R(N1+1,M2,2))/2.0E0
      AZM=(R(N1+1,M1,3)+R(N1+1,M2,3))/2.0E0
      GO TO 22
   20 AYM=R(N1+1,M1+1,2)
      AZM=R(N1+1,M1,3)
      GO TO 22
   21 AYM=R(N1+1,M2-1,2)
      AZM=R(N1+1,M2,3)
   22 GO TO (23,24),IM
   23 IF (N2.GT.NMAX) GO TO 25
      IF (N1.LT.1) GO TO 26
      AYN=(R(N1,M1+1,2)+R(N2,M1+1,2))/2.0E0
      AZN=(R(N1,M1+1,3)+R(N2,M1+1,3))/2.0E0
      GO TO 27
   24 IF (N2.GT.NMAX) GO TO 25
      IF (N1.LT.1) GO TO 26
      A=(R(N1,M1+1,2)-R(N2,M1+1,2))/(R(N1,M1+1,1)-R(N2,M1+1,1))
      B=R(N1,M1+1,2)-A*R(N1,M1+1,1)
      AYN=A*R(N1+1,M1+1,1)+B
      A=(R(N1,M1+1,3)-R(N2,M1+1,3))/(R(N1,M1+1,1)-R(N2,M1+1,1))
      B=R(N1,M1+1,3)-A*R(N1,M1+1,1)
      AZN=A*R(N1+1,M1+1,1)+B
      GO TO 27
   25 AYN=R(N1,M1+1,2)
```

```
                                                                              PNT 106
      AZN=R(N1,M1+1,3)                                                        PNT 107
      ICODE1=1                                                                PNT 108
      GO TO 27                                                                PNT 109
   26 AYN=R(N2,M1+1,2)                                                        PNT 110
      AZN=R(N2,M1+1,3)                                                        PNT 111
      ICODE2=1                                                                PNT 112
   27 N=N1+1                                                                  PNT 113
      M=M1+1                                                                  PNT 114
      D=2.0E0                                                                 PNT 115
   28 AVY=(AYM+AYN)/D                                                         PNT 116
      AVZ=(AZM+AZN)/D                                                         PNT 117
      IF (IMETH.EQ.0) WRITE (JWRITE,29) N,M,R(N,M,2),AVY,R(N,M,3),AVZ         PNT 118
   29 FORMAT (1X,/,5X,'POINT(N=',I2,',M=',I2,') Y=',1PE14.7,' -->',1PE14      PNT 119
     1.7,/,22X,'Z=',1PE14.7,' -->',1PE14.7,5X,14HSIMPLE AVERAGE)              PNT 120
      IF (IMETH.EQ.1) WRITE (JWRITE,30) N,M,R(N,M,2),AVY,R(N,M,3),AVZ         PNT 121
   30 FORMAT (1X,/,5X,'POINT(N=',I2,',M=',I2,') Y=',1PE14.7,' -->',1PE14      PNT 122
     1.7,/,22X,'Z=',1PE14.7,' -->',1PE14.7,5X,20HLINEAR INTERPOLATION)        PNT 123
      R(N,M,2)=AVY                                                            PNT 124
      R(N,M,3)=AVZ                                                            PNT 125
      GO TO 1                                                                 PNT 126
   31 IF (ICODE2.NE.0) WRITE (JWRITE,32)                                      PNT 127
   32 FORMAT (1X,///,5X,67HWARNING...DATA NOT COMPATIBLE AS INPUT TO 'FL      PNT 128
     1OWBODY' PROGRAM DUE TO,/,15X,39HPOINT MODIFICATION(S) AT FRONT OF       PNT 129
     1BODY.)                                                                  PNT 130
      IF (ICODE1.NE.0) WRITE (JWRITE,33)                                      PNT 131
   33 FORMAT (1X,///,5X,67HWARNING...DATA NOT COMPATIBLE AS INPUT TO 'FL      PNT 132
     1OWBODY' PROGRAM DUE TO,/,15X,38HPOINT MODIFICATION(S) AT REAR OF B      PNT 133
     1ODY.)                                                                   PNT 134
      RETURN                                                                  PNT 135
      END

      SUBROUTINE TEST(M1,M2,MF,NP,J1,J2,KM1,KM2,KN1,KN2,MX,NX)                TST   1
C                                                                             TST   2
C***  SUBROUTINE TEST ALLOWS THE SPECIFICATION OF ARBITRARY ADDITIONAL         TST   3
C***  STATIONS                                                                TST   4
C                                                                             TST   5
      DIMENSION MP(J1,2),NP(J2,2)                                             TST   6
      COMMON /INOUT/JREAD,JWRITE,JPUNCH,KK(5)                                 TST   7
      DATA M/4HM   /,N/4HN   /,MN/4HEND /                                     TST   8
C                                                                             TST   9
```

```
C***  INITIALIZE PARAMETERS                                             TST  10
      M1=0                                                              TST  11
      M2=0                                                              TST  12
      KM1=0                                                             TST  13
      KM2=0                                                             TST  14
      KN1=0                                                             TST  15
      KN2=0                                                             TST  16
      L1=0                                                              TST  17
      L2=0                                                              TST  18
      WRITE (JWRITE,1)                                                  TST  19
    1 FORMAT (1H1,///,13X,43HADDITIONAL ARBITRARY STATION SPECIFICATIONS TST  20
     1,//,10X,48HREFERENCE|___ M-STATIONS ___|___ N-STATIONS ___|,/,11X, TST  21
     19HSTATION |,18X,1H|,18X,1H|,/,19X,39H|1ST HALF  2ND HALF|1ST HALF  TST  22
     1 2ND HALF|,/)                                                     TST  23
      DO 2 J=1,J1                                                       TST  24
      MP(J,1)=0                                                         TST  25
    2 MP(J,2)=0                                                         TST  26
      DO 3 J=1,J2                                                       TST  27
      NP(J,1)=0                                                         TST  28
    3 NP(J,2)=0                                                         TST  29
      IF (MX.GE.7) GO TO 4                                              TST  30
      KM1=MX                                                            TST  31
    4 IF (NX.GE.7) GO TO 5                                              TST  32
      KN1=NX                                                            TST  33
    5 IF (MX.LT.7) GO TO 6                                              TST  34
      KM1=(MX+1)/2                                                      TST  35
      KM2=KM1                                                           TST  36
    6 IF (NX.LT.7) GO TO 7                                              TST  37
      KN1=(NX+1)/2                                                      TST  38
      KN2=KN1                                                           TST  39
C***  READ DATA CARDS                                                   TST  40
    7 READ (JREAD,8) NAME,K1,NL                                         TST  41
    8 FORMAT (A4,1X,2I5)                                                TST  42
      IF (NAME.EQ.MN) GO TO 11                                          TST  43
      IF (NAME.EQ.M) GO TO 9                                            TST  44
      IF (NAME.EQ.N) GO TO 10                                           TST  45
      GO TO 12                                                          TST  46
    9 IF (K1.LE.0.OR.K1.GE.MX) GO TO 7                                  TST  47
      IF (K1.LT.KM1)MP(K1,1)=IABS(NL)                                   TST  48
      IF (K1.GE.KM1)MP(K1-KM1+1,2)=IABS(NL)                             TST  49
      L1=L1+IABS(NL)                                                    TST  50
      GO TO 7                                                           TST  51
```

```
10 IF (K1.LE.0.OR.K1.GE.NX) GO TO 7                                     TST  52
   IF (K1.LT.KN1)NP(K1,1)=IABS(NL)                                      TST  53
   IF (K1.GE.KN1)NP(K1-KN1+1,2)=IABS(NL)                                TST  54
   L2=L2+IABS(NL)                                                       TST  55
   GO TO 7                                                              TST  56
11 M1=MX+L1                                                             TST  57
   M2=NX+L2                                                             TST  58
   IF ((2*(M1/2)).EQ.M1.OR.(2*(M2/2)).EQ.M2) GO TO 14                   TST  59
   GO TO 16                                                             TST  60
12 WRITE (JWRITE,13)                                                    TST  61
13 FORMAT (1X,///,1X,57HERROR... DATA CARD IN TEST ROUTINE HAS NO DIR   TST  62
  1ECTION LABEL,///)                                                    TST  63
   CALL EXIT                                                            TST  64
   RETURN                                                               TST  65
14 WRITE (JWRITE,15) M1,M2                                              TST  66
15 FORMAT (1X,///,1X,38HWARNING... TOTAL NUMBER OF M-STATIONS(,I3,16H   TST  67
  1) OR N-STATIONS(,I3,22H) IS NOT AN ODD NUMBER,//)                    TST  68
16 IF (M1.EQ.MX)M1=0                                                    TST  69
   IF (M2.EQ.NX)M2=0                                                    TST  70
   K=KM1                                                                TST  71
   IF (KN1.GT.K)K=KN1                                                   TST  72
   DO 23 J=1,K                                                          TST  73
   IF (J.LE.KM1.AND.J.LE.KN1) GO TO 17                                  TST  74
   IF (J.LE.KM1.AND.J.GT.KN1) GO TO 19                                  TST  75
   IF (J.GT.KM1.AND.J.LE.KN1) GO TO 21                                  TST  76
17 WRITE (JWRITE,18) J,MP(J,1),MP(J,2),NP(J,1),NP(J,2)                  TST  77
18 FORMAT (12X,I3,7X,I3,7X,I3,6X,I3,7X,I3)                              TST  78
   GO TO 23                                                             TST  79
19 WRITE (JWRITE,20) J,MP(J,1),MP(J,2)                                  TST  80
20 FORMAT (12X,I3,7X,I3,7X,I3)                                          TST  81
   GO TO 23                                                             TST  82
21 WRITE (JWRITE,22) J,NP(J,1),NP(J,2)                                  TST  83
22 FORMAT (12X,I3,26X,I3,7X,I3)                                         TST  84
23 CONTINUE                                                             TST  85
   WRITE (JWRITE,24)                                                    TST  86
24 FORMAT (1X,////)                                                     TST  87
   RETURN                                                               TST  88
   END                                                                  TST  89


   SUBROUTINE PCS(K,F,X,XY,Y,INPTS,KN,C,D,DIAG,JWRITE,J4,J3,M1,N,J,KK PCS   1
```

```
     1)
C
C***  CONTROL ROUTINE FOR PIECEWISE CUBIC SPLINE
C
      DIMENSION F(J4),X(J4),XY(J3),Y(J3),C(4,J4),D(J4),DIAG(J4),N(J,2)
      COMMON /CODE/ICODE,IBP1,IBP2
C
      KN=K+(K-1)*INPTS
      JP1D2=K
      IF (M1.EQ.0) GO TO 1
      IF (ICODE.EQ.0)KN=JP1D2+IBP1
      IF (ICODE.EQ.1)KN=JP1D2+IBP2
    1 IF (KN.EQ.K) GO TO 2
      DF=1.0E-4
      GO TO 4
    2 DO 3 I=1,K
      Y(I)=F(I)
    3 XY(I)=X(I)
      GO TO 9
    4 C(2,1)=(F(2)-F(1))/(X(2)-X(1))
      C(2,K)=(F(K)-F(K-1))/(X(K)-X(K-1))
      DO 5 I=1,K
    5 C(1,I)=F(I)
      CALL SPLINE(K,X,C,D,DIAG,J4)
      XX=X(1)
      XXH=X(K)
      XINC=0.0E0
      M=0
      L=1
    6 M=M+1
      IF (M.GT.KN) GO TO 10
      XX=XX+XINC
      IF (ABS(XX-XXH).LT.DF)XX=XXH
      IF (XX.EQ.XXH) GO TO 7
      IF (ABS(XX-X(L)).LT.DF)XX=X(L)
      IF (ABS(XX-X(L+1)).LT.DF)XX=X(L+1)
      IF (XX.EQ.X(L+1).AND.XX.NE.X(K))L=L+1
    7 XY(M)=XX
      Y(M)=PCUBIC(XX,X,C,K,J4)
      IF (XX.EQ.XXH) GO TO 9
      IF (M1.EQ.0) GO TO 8
      IF (KK.EQ.0)INPTS=N(L,ICODE+1)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF (KK.EQ.1.AND.(JP1D2-L).NE.0)INPTS=N(JP1D2-L,ICODE+1)
    8 IF (L.LT.K)XINC=(X(L+1)-X(L))/(INPTS+1)
      GO TO 6
    9 RETURN
   10 WRITE (JWRITE,11) M,KN
   11 FORMAT (3X,16HERROR IN PCS: M=,I4,6H > KN=,I4)
      CALL EXIT
      RETURN
      END

      SUBROUTINE SPLINE(NP1,XI,C,D,DIAG,J)
C
C***  PIECEWISE CUBIC SPLINE
C
      DIMENSION XI(J),C(4,J),D(J),DIAG(J)
C
      D(1)=0.0E0
      DIAG(1)=1.0E0
      N=NP1-1
      DO 1 M=2,NP1
      D(M)=XI(M)-XI(M-1)
    1 DIAG(M)=(C(1,M)-C(1,M-1))/D(M)
      DO 2 M=2,N
      C(2,M)=3.0E0*(D(M)*DIAG(M+1)+D(M+1)*DIAG(M))
    2 DIAG(M)=2.0E0*(D(M)+D(M+1))
      DO 3 M=2,N
      G=-D(M+1)/DIAG(M-1)
      DIAG(M)=DIAG(M)+G*D(M-1)
    3 C(2,M)=C(2,M)+G*C(2,M-1)
      NJ=NP1
      DO 4 M=2,N
      NJ=NJ-1
    4 C(2,NJ)=(C(2,NJ)-D(NJ)*C(2,NJ+1))/DIAG(NJ)
      DO 5 I=1,N
      DX=XI(I+1)-XI(I)
      DF1=(C(1,I+1)-C(1,I))/DX
      DF3=C(2,I)+C(2,I+1)-2.0E0*DF1
      C(3,I)=(DF1-C(2,I)-DF3)/DX
    5 C(4,I)=DF3/DX/DX
      RETURN
```

```
      END                                                                SPL  31

      FUNCTION PCUBIC(XBAR,XI,C,NP1,J)                                   PCB   1
C                                                                        PCB   2
C***  FUNCTION EVALUATION OF PIECEWISE CUBIC SPLINE                      PCB   3
C                                                                        PCB   4
      DIMENSION XI(J),C(4,J)                                             PCB   5
C                                                                        PCB   6
      I=1                                                                PCB   7
      DX=XBAR-XI(I)                                                      PCB   8
      IF (DX)  1,4,3                                                     PCB   9
    1 IF (I.EQ.1) GO TO 4                                                PCB  10
      I=I-1                                                              PCB  11
      DX=XBAR-XI(I)                                                      PCB  12
      IF (DX)  1,4,4                                                     PCB  13
    2 I=I+1                                                              PCB  14
      DX=DDX                                                             PCB  15
    3 IF (I.EQ.(NP1-1)) GO TO 4                                          PCB  16
      DDX=XBAR-XI(I+1)                                                   PCB  17
      IF (DDX)  4,2,2                                                    PCB  18
    4 PCUBIC=C(1,I)+DX*(C(2,I)+DX*(C(3,I)+DX*C(4,I)))                    PCB  19
      RETURN                                                             PCB  20
      END                                                                PCB  21

      SUBROUTINE XYZPLT(M,N,XFUS,SFUS)                                   XYZ   1
C                                                                        XYZ   2
C***  CONTROL ROUTINE FOR CONFIGURATION PLOTS                            XYZ   3
C                                                                        XYZ   4
      DIMENSION FIRST(7),XFUS(N),SFUS(N,M,2)                             XYZ   5
      COMMON /PASS/ABC(20),ABCDE(20),HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF XYZ   6
     1,ZF,DIST,FMAG,PLOTSZ,TYPF,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID, XYZ   7
     1ZMID,BIGD,KODE,ISP                                                 XYZ   8
      COMMON /INOUT/JREAD,JWRITE,JPUNCH,KFILE1,KFILE2,KFILE3,KFILE4,IWRI XYZ   9
     1TE                                                                 XYZ  10
      COMMON /PCODE/LCODE                                                XYZ  11
      COMMON /TRANS/TITLF(20)                                            XYZ  12
      DATA FIRST/4HNEW ,4HBODY,4H WIT,4HH IM,4HPRCV,4HED G,4HRID /,TAME/ XYZ  13
     14H    /                                                            XYZ  14
```

```
C
      DO 1 I=1,20
      IF (LCCDE.EQ.1)ABC(I)=TITLE(I)
      IF (LCODE.FQ.2.AND.I.LE.7)ABC(I)=FIRST(I)
    1 IF (LCODE.EQ.2.AND.I.GT.7)ABC(I)=TAME
      WRITE (JWRITE,2) (ABC(I),I=1,20)
    2 FORMAT (28X25HCONFIGURATION DESCRIPTION//1X20A4/)
C***  INPUT CONFIGURATION DESCRIPTION AND INITIALIZE
      CALL CBC10(M,N,XFUS,SFUS)
C***  PLOT CONFIGURATION
      WRITE (JWRITE,3)
    3 FORMAT (//36X9HPLOT DATA//)
    4 READ (JREAD,5) (ABCDE(I),I=1,20)
    5 FORMAT (20A4)
      WRITE (KFILE3,5) (ABCDE(I),I=1,20)
      REWIND KFILE3
      READ (KFILE3,6) HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FMAG,P
     1LOTSZ,TYPE,KODE
      REWIND KFILF3
    6 FORMAT (2A2,A3,9F5.3,A3,16X,I1)
      WRITE (JWRITE,7) HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FMAG,
     1PLOTSZ,TYPE,KODE
    7 FORMAT (1X,2A2,A3,9(1X,F10.5),A3,16X,I1)
      CALL CBC20(M,N)
      IF (KODF.EQ.0) GO TC 4
      WRITE (JWRITE,8)
    8 FORMAT (1X,//,1X,127('-'))
      RETURN
      END


      SUBROUTINE CBC1J(NRADX,NFORX,XFUS,SFUS)
C
C***  INPUTS AND INITIALIZES CONFIGURATION DESCRIPTICN
C
      COMMCN /PASS/ABC(20),ABCDE(20),HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF
     1,ZF,DIST,FMAG,PLOTSZ,TYPE,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,
     1ZMID,BIGD,KODE,ISP
      COMMCN /INOUT/JREAD,JWRITE,JPUNCH,KFILF1,KFILE2,KFILF3,KFILE4,IWRI
     1TE
      DIMENSICN XFUS(NFORX),SFUS(NFORX,NRADX,2)
```

```
      COMMON /AV/ALRT(212,3,2),VECRT(211,3)
      COMMON /PCODE/LCODE
C
      REWIND KFILE1
      REWIND KFILE2
      REWIND KFILE4
      LREAD=KFILE1
      IF (LCODE.NE.1)LREAD=KFILE4
      NRAD=NRADX
      NFUSOR=NFORX
      N=NFUSOR
      READ (LREAD)(XFUS(I),I=1,N)
      IF (IWRITE.EQ.0) WRITE (JWRITE,1) (XFUS(I),I=1,N)
    1 FORMAT (1X,F10.5,1X,F10.5,1X,F10.5,1X,F10.5,1X,F10.5,1X,F10.5,1X,F
     110.5,1X,F10.5,1X,F10.5,1X,F10.5)
      IF (IWRITE.EQ.0) WRITE (JWRITE,2)
    2 FORMAT (1X,/)
      DO 4 NN=1,N
      DO 3 K=1,2
      READ (LREAD)(SFUS(NN,MM,K),MM=1,NRAD)
      IF (IWRITE.EQ.0) WRITE (JWRITE,1) (SFUS(NN,MM,K),MM=1,NRAD)
    3 CONTINUE
      IF (IWRITE.EQ.0) WRITE (JWRITE,2)
    4 CONTINUE
      XMIN=XFUS(1)
      XMAX=XFUS(1)
      YMAX=SFUS(1,1,1)
      ZMIN=SFUS(1,1,2)
      ZMAX=SFUS(1,1,2)
      XMIN=AMIN1(XMIN,XFUS(1))
      XMAX=AMAX1(XMAX,XFUS(NFUSOR))
      DO 5 NN=1,NFUSOR
      DO 5 NR=1,NRAD
      YMAX=AMAX1(YMAX,SFUS(NN,NR,1))
      ZMIN=AMIN1(ZMIN,SFUS(NN,NR,2))
    5 ZMAX=AMAX1(ZMAX,SFUS(NN,NP,2))
C***  SETUP 1ST LINE IN STREAMWISE DIRECTION
      NL1=NFUSOR-1
      NAN=NRAD
      DO 6 N=1,NFUSOR
      ALRT(N,1,2)=XFUS(N)
      ALRT(N,2,2)=SFUS(N,1,1)
```

```
    6 ALRT(N,3,2)=SFUS(N,1,2)
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NFUSOR),N3=1,3)
      DO 9 NN=2,NAN
      DO 7 N=1,NFUSOR
      DO 7 N3=1,3
      ALRT(N,N3,1)=ALRT(N,N3,2)
    7 CONTINUE
      DO 8 N=1,NFUSOR
      ALRT(N,2,2)=SFUS(N,NN,1)
    8 ALRT(N,3,2)=SFUS(N,NN,2)
      CALL SURCL(NFUSOR)
      WRITE (KFILE2)((VECRT(N,N3),N=1,NL1),N3=1,3)
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NFUSOR),N3=1,3)
    9 CONTINUE
C***  SETUP 1ST LINE AROUND BODY
      NL1=NAN-1
      DO 10 N=1,NAN
      ALRT(N,1,2)=XFUS(1)
      ALRT(N,2,2)=SFUS(1,N,1)
   10 ALRT(N,3,2)=SFUS(1,N,2)
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NAN),N3=1,3)
      DO 13 NN=2,NFUSOR
      DO 11 N=1,NAN
      DO 11 N3=1,3
      ALRT(N,N3,1)=ALRT(N,N3,2)
   11 CONTINUE
      DO 12 N=1,NAN
      ALRT(N,1,2)=XFUS(NN)
      ALRT(N,2,2)=SFUS(NN,N,1)
   12 ALRT(N,3,2)=SFUS(NN,N,2)
      CALL SURCC(NAN)
      WRITE (KFILE2)((VECRT(N,N3),N=1,NL1),N3=1,3)
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NAN),N3=1,3)
   13 CONTINUE
C***  FIND MAXIMUM DISTANCE AND MIDPOINT
      YMIN=-YMAX
      XDIS=XMAX-XMIN
      YDIS=YMAX-YMIN
      ZDIS=ZMAX-ZMIN
      BIGD=AMAX1(XDIS,YDIS,ZDIS)
      XMID=.5*(XMAX-XMIN)+XMIN
      YMID=0.
```

```
      ZMID=.5*(ZMAX-ZMIN)+ZMIN                                          C10  95
      RETURN                                                            C10  96
      END                                                               C10  97


      SUBROUTINE SURCL(NPT)                                             SCL   1
C                                                                       SCL   2
C***  COMPUTES SURFACE UNIT NORMALS                                     SCL   3
C                                                                       SCL   4
      COMMON /AV/FLINE(212,3,2),FVEC(211,3)                             SCL   5
C                                                                       SCL   6
      DO 2 N=2,NPT                                                      SCL   7
      T1X=FLINE(N,1,2)-FLINE(N-1,1,1)                                   SCL   8
      T2X=FLINE(N-1,1,2)-FLINE(N,1,1)                                   SCL   9
      T1Y=FLINE(N,2,2)-FLINE(N-1,2,1)                                   SCL  10
      T2Y=FLINE(N-1,2,2)-FLINE(N,2,1)                                   SCL  11
      T1Z=FLINE(N,3,2)-FLINE(N-1,3,1)                                   SCL  12
      T2Z=FLINE(N-1,3,2)-FLINE(N,3,1)                                   SCL  13
      XNX=T2Y*T1Z-T1Y*T2Z                                               SCL  14
      YNY=T1X*T2Z-T2X*T1Z                                               SCL  15
      ZNZ=T2X*T1Y-T1X*T2Y                                               SCL  16
      FN=SQRT(XNX**2+YNY**2+ZNZ**2)                                     SCL  17
      IF (FN.EQ.0.) GO TO 1                                             SCL  18
      FVEC(N-1,1)=XNX/FN                                                SCL  19
      FVEC(N-1,2)=YNY/FN                                                SCL  20
      FVEC(N-1,3)=ZNZ/FN                                                SCL  21
      GO TO 2                                                           SCL  22
    1 FVEC(N-1,1)=0.                                                    SCL  23
      FVEC(N-1,2)=0.                                                    SCL  24
      FVEC(N-1,3)=0.                                                    SCL  25
    2 CONTINUE                                                          SCL  26
      RETURN                                                            SCL  27
      END                                                               SCL  28


      SUBROUTINE SURCC(NPT)                                             SCC   1
C                                                                       SCC   2
C***  COMPUTES SURFACE UNIT NORMALS                                     SCC   3
C                                                                       SCC   4
      COMMON /AV/FLINE(212,3,2),FVEC(211,3)                             SCC   5
```

```
C
      DO 2 N=2,NPT
      T1X=FLINE(N,1,2)-FLINE(N-1,1,1)
      T2X=FLINE(N,1,1)-FLINE(N-1,1,2)
      T1Y=FLINE(N,2,2)-FLINE(N-1,2,1)
      T2Y=FLINE(N,2,1)-FLINE(N-1,2,2)
      T1Z=FLINE(N,3,2)-FLINE(N-1,3,1)
      T2Z=FLINE(N,3,1)-FLINE(N-1,3,2)
      XNX=T2Y*T1Z-T1Y*T2Z
      YNY=T1X*T2Z-T2X*T1Z
      ZNZ=T2X*T1Y-T1X*T2Y
      FN=SQRT(XNX**2+YNY**2+ZNZ**2)
      IF (FN.EQ.0.) GO TO 1
      FVEC(N-1,1)=XNX/FN
      FVEC(N-1,2)=YNY/FN
      FVEC(N-1,3)=ZNZ/FN
      GO TO 2
    1 FVEC(N-1,1)=0.
      FVEC(N-1,2)=0.
      FVEC(N-1,3)=0.
    2 CONTINUE
      RETURN
      END

      SUBROUTINE CBC20(NRADX,NFORX)
C
C***  CONTROL ROUTINE FOR VARIOUS TYPES OF PLOTS OF A BODY
C
      COMMON /PASS/ABC(20),ABCDE(20),HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF
     1,ZF,DIST,FMAG,PLOTSZ,TYPE,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,
     1ZMID,BIGD,KODF,ISP
      COMMON /INOUT/JREAD,JWRITE,JPUNCH,KFILE1,KFILE2,KFILE3,KFILE4,IWRI
     1TE
      DIMENSION ORG(3)
      DATA TYPEP/3HPER/,TYPES/3HSTE/,TYPEV/3HVU3/
      DATA X1/1HX/,Y1/1HY/,Z1/1HZ/
C
      REWIND KFILE2
C***  SAVE MIN AND MAX
      XSAV=XMIN
```

```
      YSAV=YMIN
      ZSAV=ZMIN
      XMSAV=XMAX
      YMSAV=YMAX
      ZMSAV=ZMAX
      IF (TYPE.NE.TYPEV) GO TO 5
      SCALF=BIGD/PLOTSZ
      ORG(1)=PHI
      ORG(2)=THETA
      ORG(3)=PSI
      PHI=0.
      THETA=0.
      PSI=0.
      YBIG=ORG(1)
      YORG=FLOAT(IFIX(YMAX/SCALE))+ORG(1)
      IF (YBIG.GT.ORG(2)) GO TO 1
      YBIG=ORG(2)
      YORG=FLOAT(IFIX(ZMAX/SCALE))+ORG(2)
    1 IF (YBIG.GT.ORG(3)) GO TO 2
      YBIG=ORG(3)
      YORG=FLOAT(IFIX(ZMAX/SCALE))+ORG(3)
    2 CALL CALPLT(2.0,YORG,-3)
C***  NOTATE ON 3VIEW PLOTS
      NCHAR=IFIX(6.*PLOTSZ)
      IF (NCHAR.GT.80) GO TO 3
      X=0.
      GO TO 4
    3 NDIF=(NCHAR-80)/2
      X=FLOAT(NDIF)/6.
      NCHAR=80
    4 CALL NOTATE(X,0.,.2,ABC,0.,NCHAR)
      XMIN=0.
      YMIN=0.
      ZMIN=0.
      HORZ=X1
      VERT=Y1
      YORG=ORG(1)-YORG-1
      CALL CALPLT(0.,YORG,-3)
      CALL CBC21(NRADX,NFORX)
      REWIND KFILE2
      VERT=Z1
      YORG=ORG(2)-ORG(1)
```

```
      CALL CALPLT(0.,YORG,-3)
      CALL CBC21(NRADX,NFORX)
      REWIND KFILE2
      HORZ=Y1
      YORG=ORG(3)-ORG(2)
      YMIN=FLOAT(IFIX(YSAV/SCALE))*SCALE
      CALL CALPLT(0.,YORG,-3)
      CALL CBC21(NRADX,NFORX)
      X=FLOAT(IFIX(PLOTSZ+6.))
      Y=1.-ORG(3)
      GO TC 9
    5 IF (TYPE.EQ.TYPES) GO TO 7
C***  NOTATE ID CN PLOT
      X=0.
      NCHAR=IFIX(11.*PLOTSZ)+3
      IF (NCHAR.LE.80) GO TO 6
      NDIF=(NCHAR-80)/2
      X=FLOAT(NDIF)/11.
      NCHAR=80
    6 CALL NOTATE(X,.5,.1,ABC,0.,NCHAR)
      CALL NOTATE(X,0.,.1,ABCDE,0.,NCHAR)
    7 CALL CALPLT(0.,0.,-3)
      IF (TYPE.EQ.TYPEP.OR.TYPE.EQ.TYPES) GO TO 8
C***  ORTHOGRAPHIC
      CALL CBC21(NRADX,NFORX)
      X=FLOAT(IFIX(PLOTSZ+2.))
      Y=-0.
      GO TO 9
    8 ISP=1
      IF (TYPE.EQ.TYPES)ISP=2
C***  PERSPECTIVE OR STEREO
      CALL CBC22(NRADX,NFORX)
      X=PLOTSZ+2.
      IF (TYPE.EQ.TYPES)X=X+PLOTSZ
      Y=-0.
C***  END CF CCMPLETE PLOT
    9 CALL CALPLT(X,Y,-3)
C***  RESTORE MIN AND MAX
      XMIN=XSAV
      YMIN=YSAV
      ZMIN=ZSAV
      XMAX=XMSAV
```

```
      YMAX=YMSAV
      ZMAX=ZMSAV
      RETURN
      END

      SUBROUTINE CBC21(NRADX,NFORX)
C
C***  CONTROL ROUTINE FOR ORTHOGRAPHIC PROJECTIONS
C
      COMMON /PASS/ABC(20),ABCDE(20),HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF
     1,ZF,DIST,FMAG,PLOTSZ,TYPE,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,
     1ZMID,BIGD,KCDE,ISP
      DIMENSION C(3)
      COMMON /DPLT/ALINE(212,3),RLINE(212,2),RVEC(211,2),PLINE(212,2),A(
     12,3),NNUM(10)
      DATA XSEE/2HX /,YSEE/2HY /,XINTST/3HOUT/,CCNV/.017453293/
      DATA HVU3/3HVU3/
C
C***  INITIALIZE
      ITEST1=1
      DMAX=BIGD
      ITEST2=1
      IF (XINTST.NE.TEST1)ITEST1=0
      IF (PSI.EQ.0..AND.THETA.EQ.0..AND.PHI.EQ.0.)ITEST2=0
      SCALF=DMAX/PLOTSZ
      PHI=CONV*PHI
      THETA=CCNV*THETA
      PSI=CONV*PSI
      IF (TYPE.EC.HVU3) GO TO 1
      XDIS=XMAX-XMIN
      YDIS=YMAX-YMIN
      ZDIS=ZMAX-ZMIN
      XFIX=.5*(DMAX-XDIS)
      XMIN=XMIN-XFIX
      XMAX=XMAX+XFIX
      YFIX=.5*(DMAX-YDIS)
      YMIN=YMIN-YFIX
      YMAX=YMAX+YFIX
      ZFIX=.5*(DMAX-ZDIS)
      ZMIN=ZMIN-ZFIX
```

```
      ZMAX=ZMAX+ZFIX
C***  ADJUST MINIMUMS FOR GRID LINES
      XMIN=FLOAT(IFIX(XMIN/SCALE))*SCALE
      YMIN=FLOAT(IFIX(YMIN/SCALE))*SCALE
      ZMIN=FLOAT(IFIX(ZMIN/SCALE))*SCALE
C***  SETUP AXIS
    1 SINPSI=SIN(PSI)
      SINTHE=SIN(THETA)
      SINPHI=SIN(PHI)
      CCSPSI=CCS(PSI)
      COSTHE=COS(THETA)
      COSPHI=COS(PHI)
      IF (XSEE.NE.HORZ) GO TO 3
C***  USE X FOR HORIZONTAL VARIABLE
      IF (ITEST2.EQ.0) GO TO 2
      A(1,1)=COSTHE*COSPSI
      A(1,2)=-SINPSI*COSPHI+SINTHE*COSPSI*SINPHI
      A(1,3)=SINPSI*SINPHI+SINTHE*COSPSI*COSPHI
    2 HMIN=XMIN
      HMAX=XMAX
      HMID=XMID
      IHORZ=1
      GO TO 7
    3 IF (YSEE.NE.HORZ) GO TO 5
C***  USE Y FOR HORIZONTAL VARIABLE
      IF (ITEST2.EQ.0) GO TO 4
      A(1,1)=COSTHE*SINPSI
      A(1,2)=COSPSI*COSPHI+SINTHE*SINPSI*SINPHI
      A(1,3)=-COSPSI*SINPHI+SINTHE*SINPSI*COSPHI
    4 HMIN=YMIN
      HMAX=YMAX
      HMID=YMID
      IHORZ=2
      GO TO 7
C***  USE Z FOR HORIZONTAL VARIABLE
    5 IF (ITEST2.EQ.0) GO TO 6
      A(1,1)=-SINTHE
      A(1,2)=COSTHE*SINPHI
      A(1,3)=COSTHE*COSPHI
    6 HMIN=ZMIN
      HMAX=ZMAX
      HMID=ZMID
```

```
      IHCRZ=3
    7 IF (XSEE.NE.VERT) GO TO 9
C***  USE X FOR VERTICAL VARIABLE
      IF (ITEST2.EQ.0) GO TO 8
      A(2,1)=COSTHE*COSPSI
      A(2,2)=-SINPSI*COSPHI+SINTHE*COSPSI*SINPHI
      A(2,3)=SINPSI*SINPHI+SINTHE*COSPSI*COSPHI
    8 VMIN=XMIN
      VMAX=XMAX
      VMID=XMID
      IVERT=1
      GO TO 13
    9 IF (YSEE.NE.VERT) GO TO 11
C***  USE Y FOR VERTICAL VARIABLE
      IF (ITEST2.EQ.0) GO TO 10
      A(2,1)=COSTHE*SINPSI
      A(2,2)=COSPSI*COSPHI+SINTHE*SINPSI*SINPHI
      A(2,3)=-COSPSI*SINPHI+SINTHE*SINPSI*COSPHI
   10 VMIN=YMIN
      VMAX=YMAX
      VMID=YMID
      IVERT=2
      GO TO 13
C***  USE Z FOR VERTICAL VARIABLE
   11 IF (ITEST2.EQ.0) GO TO 12
      A(2,1)=-SINTHE
      A(2,2)=COSTHE*SINPHI
      A(2,3)=COSTHE*COSPHI
   12 VMIN=ZMIN
      VMAX=ZMAX
      VMID=ZMID
      IVERT=3
C***  CHECK PAPER PLANE
   13 IF (.NOT.((IHORZ.EQ.1.AND.IVERT.EQ.2).OR.(IVERT.EQ.1.AND.IHORZ.EQ.
     12))) GO TO 14
      ITEST=3
      C(1)=-SINTHE
      C(2)=COSTHE*SINPHI
      C(3)=COSTHE*COSPHI
      GO TO 16
   14 IF (.NOT.((IHORZ.EQ.1.AND.IVERT.EQ.3).OR.(IVERT.EQ.1.AND.IHORZ.EQ.
     13))) GO TO 15
```

```
      ITEST=2                                                           C21  120
      C(1)=COSTHE*SINPSI                                                C21  121
      C(2)=COSPSI*COSPHI+SINTHE*SINPSI*SINPHI                           C21  122
      C(3)=-COSPSI*SINPHI+SINTHE*SINPSI*COSPHI                          C21  123
      GO TO 16                                                          C21  124
   15 ITEST=1                                                           C21  125
      C(1)=COSTHE*COSPSI                                                C21  126
      C(2)=-SINPSI*COSPHI+SINTHE*COSPSI*SINPHI                          C21  127
      C(3)=SINPSI*SINPHI+SINTHE*COSPSI*COSPHI                           C21  128
C***  CENTER WITHIN PAGE SIZE IF SIZE GREATER THAN 28 INCHES             C21  129
   16 IF (PLOTSZ.GT.28..AND.TYPE.NE.EVU3)VMIN=-13.*SCALE+FLOAT(IFIX(VMID C21  130
     1/SCALE))*SCALE                                                    C21  131
C***  ROTATE MIDPOINT TO PLACE ROTATED VIEW CORRECTLY                    C21  132
      IF (ITEST2.EQ.0) GO TO 17                                         C21  133
      AMID1=A(1,1)*XMID+A(1,2)*YMID+A(1,3)*ZMID                         C21  134
      AMID2=A(2,1)*XMID+A(2,2)*YMID+A(2,3)*ZMID                         C21  135
      HMIN=HMIN-HMID+AMID1                                              C21  136
      VMIN=VMIN-VMID+AMID2                                              C21  137
C***  BEGIN PLOTTING LINES                                              C21  138
   17 CALL PLOTIT(NRADX,NFORX,ITEST,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN, C21  139
     1SCALE,C)                                                          C21  140
      CALL PLOTIT(NFORX,NRADX,ITEST,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN, C21  141
     1SCALE,C)                                                          C21  142
      RETURN                                                            C21  143
      END                                                               C21  144


      SUBROUTINE PLOTIT(NL,NPT,ITEST,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN PLT    1
     1,SCALE,C)                                                         PLT    2
C                                                                       PLT    3
C***  READS LINES OF POINTS DEFINING A SURFACE FROM DISK, MANIPULATES IN PLT    4
C***  SPECIFIED MANNER, AND PLOTS                                       PLT    5
C                                                                       PLT    6
      COMMON /INOUT/JREAD,JWRITE,JPUNCH,KFILE1,KFILE2,KFILE3,KFILE4,IWRI PLT    7
     1TE                                                                PLT    8
      DIMENSION VECRT(211,3,2),VECLF(211,3,2),XLINE(214,2),C(3)         PLT    9
      COMMON /DPLT/ALINF(212,3),RLINF(212,2),RVFC(211,2),PLINE(212,2),A( PLT   10
     12,3),NNUM(10)                                                     PLT   11
C                                                                       PLT   12
      NVFC=NPT-1                                                        PLT   13
      DO 31 N=1,NL                                                      PLT   14
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF (N.GT.1) GO TO 1
      KODE=3
      K1=2
      K2=2
      GO TO 4
    1 KODE=1
      K1=1
      K2=2
      DO 3 NV=1,NVEC
      DO 2 N3=1,3
      VECRT(NV,N3,1)=VECRT(NV,N3,2)
      VECLF(NV,N3,1)=VECLF(NV,N3,2)
    2 CONTINUE
    3 CONTINUE
    4 READ (KFILE2)((ALINE(NN,N3),NN=1,NPT),N3=1,3)
      IF (N.NE.NL) GO TO 5
      KODE=2
      K1=1
      K2=1
      GO TO 7
    5 READ (KFILE2)((VECRT(NN,N3,2),NN=1,NVEC),N3=1,3)
      DO 6 NN=1,NVEC
      VECLF(NN,1,2)=VECRT(NN,1,2)
      VECLF(NN,2,2)=-VECRT(NN,2,2)
    6 VECLF(NN,3,2)=VECRT(NN,3,2)
    7 DO 30 NN2=1,2
C***  LOOP FOR RIGHT AND LEFT SIDE OF BODY
      IF (NN2.EQ.1) GO TO 9
      DO 8 NN=1,NPT
    8 ALINE(NN,2)=-ALINE(NN,2)
    9 IF (ITEST1.EQ.1) GO TO 14
      IF (ITEST2.EQ.1) GO TO 11
C***  NO ROTATION OR VISIBILITY TEST
      DO 10 NN=1,NPT
      XLINE(NN,1)=ALINE(NN,IHORZ)
   10 XLINE(NN,2)=ALINE(NN,IVERT)
      GO TO 13
C***  ROTATE BUT NO VISIBILITY TEST
   11 CALL PTROT(NPT)
      DO 12 NN=1,NPT
      DO 12 N2=1,2
      XLINE(NN,N2)=RLINE(NN,N2)
```

```
   12 CONTINUE
C***  SCALE AND PLOT
   13 XLINE(NPT+1,1)=HMIN
      XLINE(NPT+1,2)=VMIN
      XLINE(NPT+2,1)=SCALE
      XLINE(NPT+2,2)=SCALE
      CALL LINE(XLINE(1,1),XLINE(1,2),NPT,1,0,0,0)
      GO TO 30
   14 IF (ITEST2.EQ.1) GO TO 19
C***  CHECK VISIBILITY BUT NO ROTATION
      DO 15 NN=1,NPT
      RLINE(NN,1)=ALINE(NN,IHORZ)
      RLINE(NN,2)=ALINE(NN,IVERT)
   15 CONTINUE
      DO 18 NN=1,NVEC
      DO 17 N2=1,2
      IF (NN2.EQ.2) GO TO 16
      RVEC(NN,N2)=VECRT(NN,ITEST,N2)
      GO TO 17
   16 RVEC(NN,N2)=VECLF(NN,ITEST,N2)
   17 CONTINUE
   18 CONTINUE
      GO TO 23
C***  ROTATE AND CHECK VISIBILITY
   19 CALL PTROT(NPT)
      IF (NN2.EQ.2) GO TO 21
      DO 20 N2=K1,K2
      CALL VECROT(NVEC,C,VECRT(1,1,N2),RVEC(1,N2))
   20 CONTINUE
      GO TO 23
   21 DO 22 N2=K1,K2
      CALL VECROT(NVEC,C,VECLF(1,1,N2),RVEC(1,N2))
   22 CONTINUE
C***  FIND VISIBLE LINES
   23 IF (ITEST.NE.2) GO TO 25
      DO 24 N2=K1,K2
      DO 24 M=1,NVEC
   24 RVEC(M,N2)=-RVEC(M,N2)
   25 CALL VISTST(KODF,NPT,NSET)
      IF (ITEST.NE.2) GO TO 27
      DO 26 N2=K1,K2
      DO 26 M=1,NVEC
```

```
   26 RVEC(M,N2)=-RVEC(M,N2)
   27 IF (NSET.EQ.0) GO TO 30
C***  SCALE AND PLOT
      NIT=0
      DO 29 N1=1,NSET
      NN=NNUM(N1)
      DO 28 NN1=1,NN
      NIT=NIT+1
      XLINE(NN1,1)=PLINE(NIT,1)
      XLINE(NN1,2)=PLINE(NIT,2)
   28 CONTINUE
      XLINE(NN+1,1)=HMIN
      XLINE(NN+1,2)=VMIN
      XLINE(NN+2,1)=SCALE
      XLINE(NN+2,2)=SCALE
      CALL LINE(XLINE(1,1),XLINE(1,2),NN,1,0,0,0)
   29 CONTINUE
   30 CONTINUE
   31 CONTINUE
      RETURN
      END


      SUBROUTINE VISTST(KODE,NPT,NSET)
C
C***  TESTS A LINE OF POINTS FOR VISIBILITY
C
      COMMON /DPLT/ALINE(212,3),RLINE(212,2),RVEC(211,2),PLINE(212,2),A(
     12,3),NNUM(10)
C
      IWRITE=0
      IF (NPT.EQ.81.OR.NPT.EQ.85)IWRITE=1
      NVEC=NPT-1
      NPLT=0
      NSET=0
      ICOUNT=0
      GO TO (1,2,3),KODE
    1 N1=1
      N2=2
      GO TO 4
    2 N1=1
```

```
      N2=1
      GO TO 4
    3 N1=2
      N2=2
    4 DO 13 N=1,NPT
      IF (N.EQ.1) GO TO 6
      IF (N.EQ.NPT) GO TO 8
      DO 5 NN=N1,N2
      IF ((RVEC(N-1,NN).GT.0.).OR.(RVEC(N,NN).GT.0.)) GO TO 12
    5 CONTINUE
      GO TO 10
    6 DO 7 NN=N1,N2
      IF (RVEC(1,NN).GT.0.) GO TO 12
    7 CONTINUE
      GO TO 10
    8 DO 9 NN=N1,N2
      IF (RVEC(NVEC,NN).GT.0.) GO TO 12
    9 CONTINUE
C***  POINT NOT VISIBLE
   10 IF (ICOUNT.LE.1) GO TO 11
      NSET=NSET+1
      NNUM(NSET)=ICOUNT
   11 ICOUNT=0
      GO TO 13
C***  POINT IS VISIBLE
   12 NPLT=NPLT+1
      ICOUNT=ICOUNT+1
      PLINE(NPLT,1)=RLINE(N,1)
      PLINE(NPLT,2)=RLINE(N,2)
   13 CONTINUE
      IF (ICOUNT.LE.1) GO TO 14
      NSET=NSET+1
      NNUM(NSET)=ICOUNT
   14 RETURN
      END

      SUBROUTINE PTROT(NPT)
C
C***  ROTATES AND PROJECTS A SET OF 3-D POINTS
C
```

```
      COMMON /DPLT/ALINE(212,3),RLINE(212,2),RVEC(211,2),PLINE(212,2),A( PTR   5
     12,3),NNUM(10)                                                     PTR   6
C                                                                       PTR   7
      DO 2 N=1,NPT                                                      PTR   8
      RLINE(N,1)=0.                                                     PTR   9
      RLINE(N,2)=0.                                                     PTR  10
      DO 1 I=1,2                                                        PTR  11
      DO 1 J=1,3                                                        PTR  12
      RLINE(N,I)=RLINE(N,I)+A(I,J)*ALINE(N,J)                           PTR  13
    1 CONTINUE                                                          PTR  14
    2 CONTINUE                                                          PTR  15
      RETURN                                                            PTR  16
      END                                                               PTR  17


      SUBROUTINE VECROT(NVEC,C,FVEC,RVEC)                               VEC   1
C                                                                       VEC   2
C***  TRANSFORMS VECTORS                                                VEC   3
C                                                                       VEC   4
      DIMENSION C(3),FVEC(211,3),RVEC(211)                              VEC   5
C                                                                       VEC   6
      DO 2 N=1,NVEC                                                     VEC   7
      SUM=0.                                                            VEC   8
      DO 1 NN=1,3                                                       VEC   9
    1 SUM=SUM+C(NN)*FVEC(N,NN)                                          VEC  10
    2 RVEC(N)=SUM                                                       VEC  11
      RETURN                                                            VEC  12
      END                                                               VEC  13


      SUBROUTINE CBC22(NRADX,NFORX)                                     C22   1
C                                                                       C22   2
C***  CONTROL ROUTINE FOR PERSPECTIVE AND STEREO                        C22   3
C                                                                       C22   4
      COMMON /PASS/ABC(20),ABCDF(20),HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF C22   5
     1,ZF,DIST,FMAG,PLOTSZ,TYPE,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID, C22   6
     1ZMID,BIGC,KODE,ISP                                                C22   7
      COMMON /INOUT/JREAD,JWRITE,JPUNCH,KFILE1,KFILE2,KFILE3,KFILE4,IWRI C22   8
     1TE                                                                C22   9
      DIMENSION XINIT(2),YINIT(2),ZINIT(2)                              C22  10
```

```
C
      XINIT(1)=PHI
      XINIT(2)=XF
      YINIT(1)=THETA
      YINIT(2)=YF
      ZINIT(1)=PSI
      ZINIT(2)=ZF
      CALL STERPT(XINIT,YINIT,ZINIT,0,1,0,3,PLOTSZ,DIST,FMAG)
C***  LOOP FOR RIGHT AND LEFT FRAMES
      DO 1 IC=1,ISP
      REWIND KFILE2
      NCI=-IC
C***  BEGIN PLOTTING LINES
      CALL PLTIT3(NRADX,NFORX,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FMAG,NC
     1I)
      CALL PLTIT3(NFORX,NRADX,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FMAG,NC
     1I)
    1 CONTINUE
      RETURN
      END


      SUBROUTINE STERPT(X,Y,Z,N,K,NC,IP,PAG,PLA,XPR)
C
C***  PLOTS STEREO        (PROGRAMMER - GEORGE C. SALLEY)
C
      DIMENSION VP(3),TRAN(3),SANG(3),CANG(3),ADJ(3),PT(4),XLP(2),ZLP(2)
      DIMENSION X(1),Y(1),Z(1)
      DIMENSION PLX(4),PLY(4),PLZ(2)
      DIMENSION PIX(4),PIY(4),PIZ(2)
      DIMENSION ILP(4),IPL(4)
      DATA PI,PI2,PI32,PI42/3.1415926,1.5707963,4.7123889,6.2831952/
      DATA PAP/1.125/
      DATA NPG/0/
      DATA NPT/1/
      DATA FRAME/9.80/
      DATA TURN/11.01/
C
      NL=1
      KK=K
      II=IP
```

```
      IF (NC)  16,1,22
    1 NP=N*K+1
      NR=NP+K
      PLIM=PAG/2.
      SF=XPR
      VPL=PLA
      DO 2 I=1,4
      PLX(I)=0.
      PLY(I)=0.
      PIX(I)=0.
      PIY(I)=0.
      IPL(I)=0
    2 ILP(I)=0
      DO 3 I=1,2
      PLZ(I)=0.
    3 PIZ(I)=0.
      VPX=X(NP)
      VPY=Y(NP)
      VPZ=Z(NP)
      FPX=X(NR)
      FPY=Y(NR)
      FPZ=Z(NR)
      VX=VPX-FPX
      VY=VPY-FPY
      VZ=VPZ-FPZ
      VP(2)=SQRT((VX**2)+(VY**2))
      VP(3)=SQRT((VZ**2)+(VP(2)**2))
      TRAN(1)=VPX-(VPL*(VX/VP(2)))
      TRAN(2)=VPY-(VPL*(VY/VP(2)))
      TRAN(3)=VPZ-(VPL*(VZ/VP(3)))
      VANG=ATAN((PAR/VP(3)))
      IF (VX)  11,4,7
    4 IF (VY)  6,60,5
    5 PANG=PI2
      GO TO 15
    6 PANG=PI32
      GO TO 15
    7 IF (VY)  10,8,9
    8 PANG=0.
      GO TO 15
    9 PANG=ATAN((VY/VX))
      GO TO 15
```

```
10 PANG=PI42-ATAN(((ABS(VY))/VX))                    STE  62
   GO TO 15                                          STE  63
11 IF (VY)  14,12,13                                 STE  64
12 PANG=PI                                           STE  65
   GO TO 15                                          STE  66
13 PANG=PI-ATAN((VY/(ABS(VX))))                      STE  67
   GO TO 15                                          STE  68
14 PANG=PI+ATAN(((ABS(VY))/(ABS(VX))))               STE  69
15 PANG=PI32-PANG                                    STE  70
   UANG=PANG-VANG                                    STE  71
   RANG=UANG+(2.*VANG)                               STE  72
   SANG(1)=SIN(UANG)                                 STE  73
   SANG(2)=SIN(RANG)                                 STE  74
   CANG(1)=COS(UANG)                                 STE  75
   CANG(2)=COS(RANG)                                 STE  76
   SANG(3)=VZ/VP(3)                                  STE  77
   CANG(3)=VP(2)/VP(3)                               STE  78
   VP(3)=VPL                                         STE  79
   XLP(1)=0.                                         STE  80
   ZLP(1)=0.                                         STE  81
   XLP(2)=FRAME                                      STE  82
   ZLP(2)=0.                                         STE  83
   ADJ(1)=PLIM                                       STE  84
   ADJ(2)=PLIM                                       STE  85
   ADJ(3)=ADJ(2)+FRAME                               STE  86
   IF (N)  60,60,22                                  STE  87
16 M=IABS(NC)                                        STE  88
   L=M                                               STE  89
   IF (NPG+NC)  23,17,23                             STE  90
17 IF (2+NC)  60,19,18                               STE  91
18 NPG=2                                             STE  92
   GO TO 20                                          STE  93
19 NPG=1                                             STE  94
20 DO 21 I=1,L                                       STE  95
   CALL CALPLT(TURN,0.,-3)                           STE  96
21 CONTINUE                                          STE  97
   CALL CALPLT(XLP(M),ZLP(M),3)                      STE  98
22 M=1                                               STE  99
   L=2                                               STE 100
23 DO 59 I=M,L                                       STE 101
   IF (NPG)  60,24,30                                STE 102
24 IF (NC)  25,29,29                                 STE 103
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
25 IF (NPT+NC)  30,26,30
26 IF (2+NC)  60,28,27
27 NPT=2
   GO TO 29
28 NPT=1
29 CALL CALPLT(XLP(I),ZLP(I),3)
30 DO 58 J=1,N
   PT(1)=((X(NO)-TRAN(1))*CANG(I))-((Y(NO)-TRAN(2))*SANG(I))
   PT(4)=((X(NO)-TRAN(1))*SANG(I))+((Y(NO)-TRAN(2))*CANG(I))
   PT(2)=((PT(4)*CANG(3))-((Z(NO)-TRAN(3))*SANG(3)))
   PT(3)=((PT(4)*SANG(3))+((Z(NO)-TRAN(3))*CANG(3)))
   IF (PT(2))  31,36,36
31 IF (ILP(I))  60,32,35
32 IF (II-3)  33,34,60
33 VX=PLX(I)-PT(1)
   VY=PLY(I)-PT(2)
   VZ=PLZ(I)-PT(3)
   VP(1)=SQRT((VX**2)+(VY**2))
   VP(2)=SQRT((VZ**2)+(VP(1)**2))
   VPL=PLY(I)/(VY/VP(1))
   PT(4)=PLX(I)-((VX/VP(1))*VPL)
   PLX(I)=PT(1)
   PT(1)=PT(4)
   PLY(I)=PT(2)
   PT(2)=0.
   PT(4)=PLZ(I)-((VZ/VP(2))*VPL)
   PLZ(I)=PT(3)
   PT(3)=PT(4)
   ILP(I)=1
   GO TO 41
34 ILP(I)=1
35 PLX(I)=PT(1)
   PLY(I)=PT(2)
   PLZ(I)=PT(3)
   GO TO 54
36 IF (ILP(I))  60,40,37
37 IF (II-3)  38,39,60
38 II=3
   IPL(I)=1
   PIX(I)=PT(1)
   PIY(I)=PT(2)
   PIZ(I)=PT(3)
```

```
      VX=PT(1)-PLX(I)
      VY=PT(2)-PLY(I)
      VZ=PT(3)-PLZ(I)
      VP(1)=SQRT((VX**2)+(VY**2))
      VP(2)=SQRT((VZ**2)+(VP(1)**2))
      VPL=PT(2)/(VY/VP(1))
      PT(2)=0.
      PT(1)=PT(1)-((VX/VP(1))*VPL)
      PT(3)=PT(3)-((VZ/VP(2))*VPL)
   39 ILP(I)=0
   40 PLX(I)=PT(1)
      PLY(I)=PT(2)
      PLZ(I)=PT(3)
   41 XP=(PT(1)+(PT(2)*(-PT(1))/(PT(2)+VP(3))))*SF
      ZP=(PT(3)+(PT(2)*(-PT(3))/(PT(2)+VP(3))))*SF
      VPL=SQRT((XP**2)+(ZP**2))
      IF (VPL-PLIM)  47,47,42
   42 IF (ILP(I+2))  60,43,46
   43 IF (II-3)  44,45,60
   44 PLX(I+2)=XP
      PLY(I+2)=ZP
      XP=PLIM*(XP/VPL)
      ZP=PLIM*(ZP/VPL)
      ILP(I+2)=1
      GO TO 53
   45 ILP(I+2)=1
   46 PLX(I+2)=XP
      PLY(I+2)=ZP
      GO TO 54
   47 IF (ILP(I+2))  60,51,48
   48 IF (II-3)  49,50,60
   49 II=3
      IPL(I+2)=1
      PLX(I+2)=XP
      PLY(I+2)=ZP
      VPL=SQRT((PLX(I+2)**2)+(PLY(I+2)**2))
      XP=PLIM*(PLX(I+2)/VPL)
      ZP=PLIM*(PLY(I+2)/VPL)
   50 ILP(I+2)=0
      GO TO 52
   51 IF ((SQRT(((PLX(I+2))**2)+((PLY(I+2))**2)))-PLIM)  52,52,48
   52 PLX(I+2)=XP
```

```
      PLY(I+2)=ZP
   53 XPT=XP+ADJ(I+1)
      YPT=ZP+ADJ(1)
      CALL CALPLT(XPT,YPT,II)
   54 II=2
      IF (IPL(I+2))  60,56,55
   55 IPL(I+2)=0
      XP=PIX(I+2)
      ZP=PIY(I+2)
      GO TO 51
   56 IF (IPL(I))  60,58,57
   57 IPL(I)=0
      PT(1)=PIX(I)
      PT(2)=PIY(I)
      PT(3)=PIZ(I)
      GO TO 40
   58 NC=NC+KK
      XLP(I)=XPT
      ZLP(I)=YPT
      NO=1
   59 II=IP
   60 RETURN
      END

      SUBROUTINE PLTIT3(NL,NPT,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FMAG,N
     1CI)
C
C***  READS LINES OF POINTS DEFINING A SURFACE FROM DISK
C
      COMMON /INOUT/JREAD,JWRITE,JPUNCH,KFILE1,KFILE2,KFILE3,KFILE4,IWRI
     1TF
      DIMENSION ALINE(214,3)
C
      ALINE(NPT+1,1)=PHI
      ALINE(NPT+2,1)=XF
      ALINE(NPT+1,2)=THETA
      ALINE(NPT+2,2)=YF
      ALINE(NPT+1,3)=PSI
      ALINE(NPT+2,3)=ZF
      DO 5 N=1,NL
```

```
      READ (KFILE2)((ALINE(NN,N3),NN=1,NPT),N3=1,3)
      IF (N.EQ.NL) GO TO 1
C***  SKIP VECTORS
      READ (KFILE2)VEC
C***  LOOP FOR RIGHT AND LEFT SIDE OF BODY
    1 DO 4 NN2=1,2
      IF (NN2.EQ.1) GO TO 3
      DO 2 NN=1,NPT
    2 ALINE(NN,2)=-ALINE(NN,2)
    3 CALL STERPT(ALINE(1,1),ALINE(1,2),ALINE(1,3),NPT,1,NCI,3,PLOTSZ,DI
     1ST,FMAG)
    4 CONTINUE
    5 CONTINUE
      RETURN
      END


      SUBROUTINE CALPLT(A,B,I)
C
C***  IMPORTANT???
C***
C***  THIS SUBROUTINE USES THE PLOTTER SOFTWARE AVAILABLE AT N. C. STATE
C***  UNIVERSITY (SUBROUTINES PLOT AND ORIGIN) TO REPRODUCE THE ACTIONS
C***  OF THE CALCOMP SOFTWARE SUBROUTINE CALPLT.
C***
C***  IF THE USER DOES NOT HAVE THE ORIGINAL CALCOMP ROUTINE CALPLT
C***  AVAILABLE TO HIM, HE MUST WRITE A DUMMY SUBROUTINE, USING THE
C***  PLOTTER SOFTWARE AVAILABLE AT HIS INSTALLATION, WHICH WILL
C***  REPRODUCE THE ACTION OF CALPLT.
C
      DIMENSION A(1),B(1)
C
      IF (I.LT.0) GO TO 1
      J=I-2
      CALL PLOT(A,B,J)
      RETURN
    1 J=IABS(I)
      J=J-2
      CALL PLOT(A,B,J)
      CALL ORIGIN(A,B,1.0)
      RETURN
```

```
      END
      SUBROUTINE NOTATE(A,B,C,D,E,I)
C
C***  IMPORTANT???
C***
C***  THIS SUBROUTINE USES THE PLOTTER SOFTWARE AVAILABLE AT N. C. STATE
C***  UNIVERSITY (SUBROUTINE SYMBOL) TO REPRODUCE THE ACTIONS OF THE
C***  CALCOMP SOFTWARE SUBROUTINE NOTATE.
C***
C***  IF THE USER DOES NOT HAVE THE ORIGINAL CALCOMP ROUTINE NOTATE
C***  AVAILABLE TO HIM, HE MUST WRITE A DUMMY SUBROUTINE, USING THE
C***  PLOTTER SOFTWARE AVAILABLE AT HIS INSTALLATION, WHICH WILL
C***  REPRODUCE THE ACTION OF NOTATE.
C
      DIMENSION D(1),DD(21)
      DATA STOP/4H   _/
C
      J=I/4
      XI=I
      XR=XI/4
      IF ((XR-J).GT.0.1)J=J+1
      DO 1 K=1,J
    1 DD(K)=D(K)
      DD(J+1)=STOP
      CALL SYMBOL(A,B,C,DD,E)
      RETURN
      END
      SUBROUTINE LINE(A,B,I,J,K,L,S)
C
C***  IMPORTANT???
C***
C***  THIS SUBROUTINE USES THE PLOTTER SOFTWARE AVAILABLE AT N. C. STATE
C***  UNIVERSITY (SUBROUTINE PLOT) TO REPRODUCE THE ACTIONS OF THE
C***  CALCOMP SOFTWARE SUBROUTINE LINE.
C***
C***  IF THE USER DOES NOT HAVE THE ORIGINAL CALCOMP ROUTINE LINE
```

```
C***  AVAILABLE TO HIM, HE MUST WRITE A DUMMY SUBROUTINE, USING THE
C***  PLCTTER SOFTWARE AVAILABLE AT HIS INSTALLATICN, WHICH WILL
C***  REPRODUCE THF ACTION OF LINE.
C
      DIMENSION A(1),B(1),X(212),Y(212)
C
      XMIN=A(I+1)
      XSCALE=A(I+2)
      YMIN=B(I+1)
      YSCALE=B(I+2)
      DO 1 II=1,I
      X(II)=(A(II)-XMIN)/XSCALE
    1 Y(II)=(B(II)-YMIN)/YSCALE
      CALL PLOT(X,Y,I)
      RETURN
      END
```

# SAMPLE INPUT - GRIDPLOT

```
   29     21     60     60        599.0          11.5
   11      1      0      0       1      1      0     0       0        1.0
CESSNA  182 FUSELAGE   (N=29, M=21)    POINT  MODIFIED
 560
 12.20849609   0.0             -0.42710018    1     1    1
 12.20849609   0.0             -0.42710018    1     2    1
 12.20849609   0.0             -0.42710018    1     3    1
 12.20849609   0.0             -0.42710018    1     4    1
 12.20849609   0.0             -0.42710018    1     5    1
 12.20849609   0.0             -0.42710018    1     6    1
 12.20849609   0.0             -0.42710018    1     7    1
 12.20849609   0.0             -0.42710018    1     8    1
 12.20849609   0.0             -0.42710018    1     9    1
 12.20849609   0.0             -0.42710018    1    10    1
 12.20849609   0.0             -0.42710018    1    11    1
 12.20849609   0.0             -0.42710018    1    12    1
 12.20849609   0.0             -0.42710018    1    13    1
 12.20849609   0.0             -0.42710018    1    14    1
 12.20849609   0.0             -0.42710018    1    15    1
 12.20849609   0.0             -0.42710018    1    16    1
 12.20849609   0.0             -0.42710018    1    17    1
 12.20849609   0.0             -0.42710018    1    18    1
 12.20849609   0.0             -0.42710018    1    19    1
 12.20849609   0.0             -0.42710018    1    20    1
 12.20849609   0.0             -0.42710018    1    21    1
 11.54179382   0.0             -1.63539696    2     1    1
 11.54179382   0.13330997      -1.59789753    2     2    1
 11.54179382   0.35830218      -1.52709770    2     3    1
 11.54179382   0.51460171      -1.44369793    2     4    1
 11.54179382   0.67080063      -1.34999657    2     5    1
 11.54179382   0.81869990      -1.25209513    2     6    1
 11.54179382   0.96880031      -1.12709713    2     7    1
 11.54179382   1.10829830      -0.99579996    2     8    1
 11.54179382   1.26459789      -0.84369946    2     9    1
 11.54179382   1.40829754      -0.65209943    2    10    1
 11.54179382   1.51659788      -0.42710018    2    11    1
 11.54179382   1.57079792      -0.18540013    2    12    1
```

| | | | | | |
|---|---|---|---|---|---|
| 11.54179382 | 1.56249905 | 0.07289881 | 2 | 13 | 1 |
| 11.54179382 | 1.49169731 | 0.32289863 | 2 | 14 | 1 |
| 11.54179382 | 1.32919884 | 0.52289993 | 2 | 15 | 1 |
| 11.54179382 | 1.10619926 | 0.67290115 | 2 | 16 | 1 |
| 11.54179382 | 0.88330030 | 0.73960114 | 2 | 17 | 1 |
| 11.54179382 | 0.66670161 | 0.77089977 | 2 | 18 | 1 |
| 11.54179382 | 0.44370186 | 0.77710056 | 2 | 19 | 1 |
| 11.54179382 | 0.21670252 | 0.78129959 | 2 | 20 | 1 |
| 11.54179382 | 0.0 | 0.78129864 | 2 | 21 | 1 |
| 10.87548828 | 0.0 | -1.87437125 | 3 | 1 | 1 |
| 10.87548828 | 0.21348673 | -1.85611057 | 3 | 2 | 1 |
| 10.87548828 | 0.42911345 | -1.81024837 | 3 | 3 | 1 |
| 10.87548828 | 0.63553590 | -1.74978733 | 3 | 4 | 1 |
| 10.87548828 | 0.85405535 | -1.66543198 | 3 | 5 | 1 |
| 10.87548828 | 1.04672909 | -1.54464340 | 3 | 6 | 1 |
| 10.87548828 | 1.22386456 | -1.37801647 | 3 | 7 | 1 |
| 10.87548828 | 1.35816669 | -1.18529987 | 3 | 8 | 1 |
| 10.87548828 | 1.47383976 | -0.97383112 | 3 | 9 | 1 |
| 10.87548828 | 1.56031990 | -0.73745412 | 3 | 10 | 1 |
| 10.87548828 | 1.62289333 | -0.48950770 | 3 | 11 | 1 |
| 10.87548828 | 1.64995766 | -0.25327464 | 3 | 12 | 1 |
| 10.87548828 | 1.64265728 | 0.03646851 | 3 | 13 | 1 |
| 10.87548828 | 1.58325468 | 0.30730814 | 3 | 14 | 1 |
| 10.87548828 | 1.43849087 | 0.54058963 | 3 | 15 | 1 |
| 10.87548828 | 1.21449389 | 0.71662617 | 3 | 16 | 1 |
| 10.87548828 | 0.98534536 | 0.81041336 | 3 | 17 | 1 |
| 10.87548828 | 0.74790800 | 0.85105801 | 3 | 18 | 1 |
| 10.87548828 | 0.52097390 | 0.87290001 | 3 | 19 | 1 |
| 10.87548828 | 0.24583626 | 0.88954067 | 3 | 20 | 1 |
| 10.87548828 | 0.0 | 0.89578819 | 3 | 21 | 1 |
| 10.20849609 | 0.0 | -2.11459732 | 4 | 1 | 1 |
| 10.20849609 | 0.24370094 | -2.11459827 | 4 | 2 | 1 |
| 10.20849609 | 0.50000101 | -2.09369564 | 4 | 3 | 1 |
| 10.20849609 | 0.76460040 | -2.05619621 | 4 | 4 | 1 |
| 10.20849609 | 1.03749943 | -1.98119736 | 4 | 5 | 1 |
| 10.20849609 | 1.27499771 | -1.83749676 | 4 | 6 | 1 |
| 10.20849609 | 1.47919346 | -1.62919712 | 4 | 7 | 1 |
| 10.20849609 | 1.63829330 | -1.37499905 | 4 | 8 | 1 |
| 10.20849609 | 1.68329906 | -1.10409737 | 4 | 9 | 1 |
| 10.20849609 | 1.71249366 | -0.82289964 | 4 | 10 | 1 |
| 10.20849609 | 1.72919941 | -0.55210012 | 4 | 11 | 1 |
| 10.20849609 | 1.72919941 | -0.28119946 | 4 | 12 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 10.20849609 | 1.72289848 | 0.0 | 4 | 13 | 1 |
| 10.20849609 | 1.67499828 | 0.29170024 | 4 | 14 | 1 |
| 10.20849609 | 1.54789829 | 0.55830002 | 4 | 15 | 1 |
| 10.20849609 | 1.32289791 | 0.76039708 | 4 | 16 | 1 |
| 10.20849609 | 1.08749771 | 0.88130093 | 4 | 17 | 1 |
| 10.20849609 | 0.82920057 | 0.93130016 | 4 | 18 | 1 |
| 10.20849609 | 0.55830109 | 0.96879864 | 4 | 19 | 1 |
| 10.20849609 | 0.27500153 | 0.99789906 | 4 | 20 | 1 |
| 10.20849609 | 0.0 | 1.01039886 | 4 | 21 | 1 |
| 9.47949219 | 0.0 | -2.20653439 | 5 | 1 | 1 |
| 9.47949219 | 0.26524454 | -2.20542717 | 5 | 2 | 1 |
| 9.47949219 | 0.54416764 | -2.18721867 | 5 | 3 | 1 |
| 9.47949219 | 0.83579701 | -2.15273476 | 5 | 4 | 1 |
| 9.47949219 | 1.14577866 | -2.08340359 | 5 | 5 | 1 |
| 9.47949219 | 1.41740608 | -1.93233395 | 5 | 6 | 1 |
| 9.47949219 | 1.61756706 | -1.68262959 | 5 | 7 | 1 |
| 9.47949219 | 1.72328186 | -1.38718700 | 5 | 8 | 1 |
| 9.47949219 | 1.76175117 | -1.08082676 | 5 | 9 | 1 |
| 9.47949219 | 1.77553558 | -0.78628421 | 5 | 10 | 1 |
| 9.47949219 | 1.78421116 | -0.50451654 | 5 | 11 | 1 |
| 9.47949219 | 1.78129768 | -0.29580021 | 5 | 12 | 1 |
| 9.47949219 | 1.77709770 | -0.00419992 | 5 | 13 | 1 |
| 9.47949219 | 1.73749823 | 0.29790026 | 5 | 14 | 1 |
| 9.47949219 | 1.61869312 | 0.58540034 | 5 | 15 | 1 |
| 9.47949219 | 1.39169788 | 0.80829787 | 5 | 16 | 1 |
| 9.47949219 | 1.14579964 | 0.94379997 | 5 | 17 | 1 |
| 9.47949219 | 0.88540184 | 1.00629807 | 5 | 18 | 1 |
| 9.47949219 | 0.59380156 | 1.04169750 | 5 | 19 | 1 |
| 9.47949219 | 0.29170138 | 1.06250000 | 5 | 20 | 1 |
| 9.47949219 | 0.0 | 1.09379959 | 5 | 21 | 1 |
| 8.87548828 | 0.0 | -2.28270912 | 6 | 1 | 1 |
| 8.87548828 | 0.26309351 | -2.28068352 | 6 | 2 | 1 |
| 8.87548828 | 0.58076197 | -2.26470566 | 6 | 3 | 1 |
| 8.87548828 | 0.89478779 | -2.23272228 | 6 | 4 | 1 |
| 8.87548828 | 1.23549366 | -2.16808796 | 6 | 5 | 1 |
| 8.87548828 | 1.53539348 | -2.01096938 | 6 | 6 | 1 |
| 8.87548828 | 1.73221207 | -1.72690201 | 6 | 7 | 1 |
| 8.87548828 | 1.81855011 | -1.39728332 | 6 | 8 | 1 |
| 8.87548828 | 1.82675457 | -1.06154537 | 6 | 9 | 1 |
| 8.87548828 | 1.82776833 | -0.75594687 | 6 | 10 | 1 |
| 8.87548828 | 1.82979393 | -0.46509069 | 6 | 11 | 1 |
| 8.87548828 | 1.82979393 | -0.18043518 | 6 | 12 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 8.87548828 | 1.82776833 | 0.11980814 | 6 | 13 | 1 |
| 8.87548828 | 1.80648994 | 0.44240475 | 6 | 14 | 1 |
| 8.87548828 | 1.74058628 | 0.77948773 | 6 | 15 | 1 |
| 8.87548828 | 1.45419788 | 0.86879939 | 6 | 16 | 1 |
| 8.87548828 | 1.19789886 | 1.01039791 | 6 | 17 | 1 |
| 8.87548828 | 0.92060075 | 1.07289791 | 6 | 18 | 1 |
| 8.87548828 | 0.61670130 | 1.09999847 | 6 | 19 | 1 |
| 8.87548828 | 0.30420119 | 1.12079906 | 6 | 20 | 1 |
| 8.87548828 | 0.0 | 1.13539866 | 6 | 21 | 1 |
| 8.31250000 | 0.0 | -2.32221413 | 7 | 1 | 1 |
| 8.31250000 | 0.30934238 | -2.32116604 | 7 | 2 | 1 |
| 8.31250000 | 0.63428217 | -2.30083370 | 7 | 3 | 1 |
| 8.31250000 | 0.96966606 | -2.25412846 | 7 | 4 | 1 |
| 8.31250000 | 1.31181812 | -2.14426041 | 7 | 5 | 1 |
| 8.31250000 | 1.59068489 | -1.93829727 | 7 | 6 | 1 |
| 8.31250000 | 1.78097320 | -1.63248825 | 7 | 7 | 1 |
| 8.31250000 | 1.86184788 | -1.29106236 | 7 | 8 | 1 |
| 8.31250000 | 1.87014583 | -0.94641578 | 7 | 9 | 1 |
| 8.31250000 | 1.87067032 | -0.63348174 | 7 | 10 | 1 |
| 8.31250000 | 1.87171841 | -0.33618915 | 7 | 11 | 1 |
| 8.31250000 | 1.87171841 | -0.04413038 | 7 | 12 | 1 |
| 8.31250000 | 1.87067032 | 0.26303864 | 7 | 13 | 1 |
| 8.31250000 | 1.85965729 | 0.59784329 | 7 | 14 | 1 |
| 8.31250000 | 1.82052994 | 0.96629173 | 7 | 15 | 1 |
| 8.31250000 | 1.69789791 | 1.31879616 | 7 | 16 | 1 |
| 8.31250000 | 1.39999866 | 1.48329735 | 7 | 17 | 1 |
| 8.31250000 | 1.05829906 | 1.53129763 | 7 | 18 | 1 |
| 8.31250000 | 0.71253254 | 1.56879616 | 7 | 19 | 1 |
| 8.31250000 | 0.34790224 | 1.58539963 | 7 | 20 | 1 |
| 8.31250000 | 0.0 | 1.59379364 | 7 | 21 | 1 |
| 7.70849609 | 0.0 | -2.36459827 | 8 | 1 | 1 |
| 7.70849609 | 0.33750254 | -2.36459732 | 8 | 2 | 1 |
| 7.70849609 | 0.69170105 | -2.33959675 | 8 | 3 | 1 |
| 7.70849609 | 1.04999924 | -2.27709675 | 8 | 4 | 1 |
| 7.70849609 | 1.39369965 | -2.11869717 | 8 | 5 | 1 |
| 7.70849609 | 1.64999866 | -1.86039539 | 8 | 6 | 1 |
| 7.70849609 | 1.83329868 | -1.53119755 | 8 | 7 | 1 |
| 7.70849609 | 1.90829945 | -1.17709732 | 8 | 8 | 1 |
| 7.70849609 | 1.91669941 | -0.82289970 | 8 | 9 | 1 |
| 7.70849609 | 1.91669941 | -0.50209981 | 8 | 10 | 1 |
| 7.70849609 | 1.91669941 | -0.19789970 | 8 | 11 | 1 |
| 7.70849609 | 1.91669941 | 0.10210031 | 8 | 12 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 7.70849609 | 1.91669941 | 0.41670042 | 8 | 13 | 1 |
| 7.70849609 | 1.91669941 | 0.76459897 | 8 | 14 | 1 |
| 7.70849609 | 1.90629864 | 1.16669846 | 8 | 15 | 1 |
| 7.70849609 | 1.82919788 | 1.61879635 | 8 | 16 | 1 |
| 7.70849609 | 1.54789925 | 1.84789658 | 8 | 17 | 1 |
| 7.70849609 | 1.18329906 | 1.91669846 | 8 | 18 | 1 |
| 7.70849609 | 0.79170042 | 1.95419598 | 8 | 19 | 1 |
| 7.70849609 | 0.38540173 | 1.96879768 | 8 | 20 | 1 |
| 7.70849609 | 0.0 | 1.96879864 | 8 | 21 | 1 |
| 7.12519836 | 0.0 | -2.40619659 | 9 | 1 | 1 |
| 7.12519836 | 0.35830206 | -2.39579678 | 9 | 2 | 1 |
| 7.12519836 | 0.73750222 | -2.37499905 | 9 | 3 | 1 |
| 7.12519836 | 1.11869907 | -2.30209732 | 9 | 4 | 1 |
| 7.12519836 | 1.45419884 | -2.09999561 | 9 | 5 | 1 |
| 7.12519836 | 1.69789886 | -1.80519717 | 9 | 6 | 1 |
| 7.12519836 | 1.86669922 | -1.45209789 | 9 | 7 | 1 |
| 7.12519836 | 1.93329906 | -1.08329868 | 9 | 8 | 1 |
| 7.12519836 | 1.93750000 | -0.72910017 | 9 | 9 | 1 |
| 7.12519836 | 1.93749905 | -0.40209949 | 9 | 10 | 1 |
| 7.12519836 | 1.93749905 | -0.09369940 | 9 | 11 | 1 |
| 7.12519836 | 1.93749905 | 0.20839971 | 9 | 12 | 1 |
| 7.12519836 | 1.93749905 | 0.52710056 | 9 | 13 | 1 |
| 7.12519836 | 1.93749905 | 0.88329893 | 9 | 14 | 1 |
| 7.12519836 | 1.93750000 | 1.29789829 | 9 | 15 | 1 |
| 7.12519836 | 1.91669941 | 1.80629730 | 9 | 16 | 1 |
| 7.12519836 | 1.65629864 | 2.09379673 | 9 | 17 | 1 |
| 7.12519836 | 1.26039886 | 2.15629673 | 9 | 18 | 1 |
| 7.12519836 | 0.84380150 | 2.19789696 | 9 | 19 | 1 |
| 7.12519836 | 0.41040224 | 2.21249771 | 9 | 20 | 1 |
| 7.12519836 | 0.0 | 2.21879673 | 9 | 21 | 1 |
| 6.56250000 | 0.0 | -2.42709732 | 10 | 1 | 1 |
| 6.56250000 | 0.37080187 | -2.42079735 | 10 | 2 | 1 |
| 6.56250000 | 0.76670182 | -2.40619659 | 10 | 3 | 1 |
| 6.56250000 | 1.16669941 | -2.33749771 | 10 | 4 | 1 |
| 6.56250000 | 1.51459380 | -2.11869621 | 10 | 5 | 1 |
| 6.56250000 | 1.76669788 | -1.80829716 | 10 | 6 | 1 |
| 6.56250000 | 1.90419865 | -1.42289829 | 10 | 7 | 1 |
| 6.56250000 | 1.93749905 | -1.02289772 | 10 | 8 | 1 |
| 6.56250000 | 1.93750000 | -0.66660017 | 10 | 9 | 1 |
| 6.56250000 | 1.93750000 | -0.33959949 | 10 | 10 | 1 |
| 6.56250000 | 1.93750000 | -0.03119946 | 10 | 11 | 1 |
| 6.56250000 | 1.93750000 | 0.27089965 | 10 | 12 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 6.56250000 | 1.93750000 | 0.58960056 | 10 | 13 | 1 |
| 6.56250000 | 1.93750000 | 0.94579887 | 10 | 14 | 1 |
| 6.56250000 | 1.93750000 | 1.36039829 | 10 | 15 | 1 |
| 6.56250000 | 1.91669750 | 1.86879730 | 10 | 16 | 1 |
| 6.56250000 | 1.73563576 | 2.25150967 | 10 | 17 | 1 |
| 6.56250000 | 1.33749866 | 2.36459637 | 10 | 18 | 1 |
| 6.56250000 | 0.88120276 | 2.36459637 | 10 | 19 | 1 |
| 6.56250000 | 0.42500114 | 2.36459637 | 10 | 20 | 1 |
| 6.56250000 | 0.0 | 2.36459732 | 10 | 21 | 1 |
| 5.95849609 | 0.0 | -2.40619755 | 11 | 1 | 1 |
| 5.95849609 | 0.38540173 | -2.39789581 | 11 | 2 | 1 |
| 5.95849609 | 0.78750134 | -2.38539696 | 11 | 3 | 1 |
| 5.95849609 | 1.20209885 | -2.34789658 | 11 | 4 | 1 |
| 5.95849609 | 1.62079906 | -2.21249580 | 11 | 5 | 1 |
| 5.95849609 | 1.85419846 | -1.83959675 | 11 | 6 | 1 |
| 5.95849609 | 1.90629864 | -1.37499905 | 11 | 7 | 1 |
| 5.95849609 | 1.92709923 | -0.97079933 | 11 | 8 | 1 |
| 5.95849609 | 1.92709923 | -0.61869991 | 11 | 9 | 1 |
| 5.95849609 | 1.92709923 | -0.30000013 | 11 | 10 | 1 |
| 5.95849609 | 1.92709923 | 0.01039886 | 11 | 11 | 1 |
| 5.95849609 | 1.92709923 | 0.31249994 | 11 | 12 | 1 |
| 5.95849609 | 1.92709923 | 0.62709999 | 11 | 13 | 1 |
| 5.95849609 | 1.92709827 | 0.98960036 | 11 | 14 | 1 |
| 5.95849609 | 1.90629864 | 1.39789867 | 11 | 15 | 1 |
| 5.95849609 | 1.88329792 | 1.88329697 | 11 | 16 | 1 |
| 5.95849609 | 1.82379887 | 2.42379735 | 11 | 17 | 1 |
| 5.95849609 | 1.34379768 | 2.42709637 | 11 | 18 | 1 |
| 5.95849609 | 0.88960165 | 2.42709732 | 11 | 19 | 1 |
| 5.95849609 | 0.42710114 | 2.42709637 | 11 | 20 | 1 |
| 5.95849609 | 0.0 | 2.42709732 | 11 | 21 | 1 |
| 5.37548923 | 0.0 | -2.40619755 | 12 | 1 | 1 |
| 5.37548923 | 0.38540268 | -2.39789677 | 12 | 2 | 1 |
| 5.37548923 | 0.78750128 | -2.38539696 | 12 | 3 | 1 |
| 5.37548923 | 1.19789982 | -2.33959579 | 12 | 4 | 1 |
| 5.37548923 | 1.61459627 | -2.20209599 | 12 | 5 | 1 |
| 5.37548923 | 1.83959770 | -1.82709599 | 12 | 6 | 1 |
| 5.37548923 | 1.90209866 | -1.36459827 | 12 | 7 | 1 |
| 5.37548923 | 1.91669941 | -0.96669954 | 12 | 8 | 1 |
| 5.37548923 | 1.91669941 | -0.61460012 | 12 | 9 | 1 |
| 5.37548923 | 1.91669941 | -0.29790020 | 12 | 10 | 1 |
| 5.37548923 | 1.91669941 | 0.01039386 | 12 | 11 | 1 |
| 5.37548923 | 1.91669941 | 0.31039906 | 12 | 12 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 5.37548923 | 1.91669941 | 0.62500006 | 12 | 13 | 1 |
| 5.37548923 | 1.91669941 | 0.98540038 | 12 | 14 | 1 |
| 5.37548923 | 1.89999866 | 1.39169693 | 12 | 15 | 1 |
| 5.37548923 | 1.67499905 | 1.87499905 | 12 | 16 | 1 |
| 5.37548923 | 1.81250000 | 2.41039753 | 12 | 17 | 1 |
| 5.37548923 | 1.34379768 | 2.42709732 | 12 | 18 | 1 |
| 5.37548923 | 0.88960266 | 2.42709732 | 12 | 19 | 1 |
| 5.37548923 | 0.42710203 | 2.42709827 | 12 | 20 | 1 |
| 5.37548923 | 0.0 | 2.42709827 | 12 | 21 | 1 |
| 4.79149055 | 0.0 | -2.40619659 | 13 | 1 | 1 |
| 4.79149055 | 0.37500191 | -2.39999676 | 13 | 2 | 1 |
| 4.79149055 | 0.77500260 | -2.36039639 | 13 | 3 | 1 |
| 4.79149055 | 1.18750000 | -2.34999561 | 13 | 4 | 1 |
| 4.79149055 | 1.59999847 | -2.22499657 | 13 | 5 | 1 |
| 4.79149055 | 1.82289791 | -1.84789562 | 13 | 6 | 1 |
| 4.79149055 | 1.89169884 | -1.39999676 | 13 | 7 | 1 |
| 4.79149055 | 1.90629864 | -1.00000000 | 13 | 8 | 1 |
| 4.79149055 | 1.90629864 | -0.65619940 | 13 | 9 | 1 |
| 4.79149055 | 1.90629864 | -0.33749950 | 13 | 10 | 1 |
| 4.79149055 | 1.90629864 | -0.03119953 | 13 | 11 | 1 |
| 4.79149055 | 1.90629864 | 0.26669967 | 13 | 12 | 1 |
| 4.79149055 | 1.90629864 | 0.56129972 | 13 | 13 | 1 |
| 4.79149055 | 1.90629864 | 0.93750012 | 13 | 14 | 1 |
| 4.79149055 | 1.88539866 | 1.33959866 | 13 | 15 | 1 |
| 4.79149055 | 1.86249828 | 1.81879616 | 13 | 16 | 1 |
| 4.79149055 | 1.77919960 | 2.32919693 | 13 | 17 | 1 |
| 4.79149055 | 1.31250095 | 2.34379673 | 13 | 18 | 1 |
| 4.79149055 | 0.87500197 | 2.34379768 | 13 | 19 | 1 |
| 4.79149055 | 0.42080206 | 2.34379768 | 13 | 20 | 1 |
| 4.79149055 | 0.0 | 2.34379768 | 13 | 21 | 1 |
| 4.22949314 | 0.0 | -2.38541508 | 14 | 1 | 1 |
| 4.22949314 | 0.36875725 | -2.38021374 | 14 | 2 | 1 |
| 4.22949314 | 0.75731053 | -2.35300515 | 14 | 3 | 1 |
| 4.22949314 | 1.16461849 | -2.33645725 | 14 | 4 | 1 |
| 4.22949314 | 1.57292271 | -2.21875000 | 14 | 5 | 1 |
| 4.22949314 | 1.80211544 | -1.80038494 | 14 | 6 | 1 |
| 4.22949314 | 1.87606144 | -1.42182732 | 14 | 7 | 1 |
| 4.22949314 | 1.89066124 | -1.02182865 | 14 | 8 | 1 |
| 4.22949314 | 1.89066124 | -0.68117833 | 14 | 9 | 1 |
| 4.22949314 | 1.89066124 | -0.36557472 | 14 | 10 | 1 |
| 4.22949314 | 1.89066124 | -0.06242180 | 14 | 11 | 1 |
| 4.22949314 | 1.89066124 | 0.23233014 | 14 | 12 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 4.22949314 | 1.89066124 | 0.54383254 | 14 | 13 | 1 |
| 4.22949314 | 1.88856316 | 0.89693540 | 14 | 14 | 1 |
| 4.22949314 | 1.86981106 | 1.29798317 | 14 | 15 | 1 |
| 4.22949314 | 1.84790953 | 1.77403545 | 14 | 16 | 1 |
| 4.22949314 | 1.74377728 | 2.25006580 | 14 | 17 | 1 |
| 4.22949314 | 1.28542233 | 2.26052094 | 14 | 18 | 1 |
| 4.22949314 | 0.85416979 | 2.26052094 | 14 | 19 | 1 |
| 4.22949314 | 0.41250896 | 2.26052189 | 14 | 20 | 1 |
| 4.22949314 | 0.0 | 2.26052189 | 14 | 21 | 1 |
| 3.66649151 | 0.0 | -2.36459827 | 15 | 1 | 1 |
| 3.66649151 | 0.36250210 | -2.36039829 | 15 | 2 | 1 |
| 3.66649151 | 0.73960114 | -2.33959770 | 15 | 3 | 1 |
| 3.66649151 | 1.14169884 | -2.32289696 | 15 | 4 | 1 |
| 3.66649151 | 1.54579926 | -2.21249580 | 15 | 5 | 1 |
| 3.66649151 | 1.78129768 | -1.87289715 | 15 | 6 | 1 |
| 3.66649151 | 1.86039925 | -1.44369698 | 15 | 7 | 1 |
| 3.66649151 | 1.87500000 | -1.04369831 | 15 | 8 | 1 |
| 3.66649151 | 1.87499905 | -0.70619953 | 15 | 9 | 1 |
| 3.66649151 | 1.87500000 | -0.39369953 | 15 | 10 | 1 |
| 3.66649151 | 1.87500000 | -0.09369946 | 15 | 11 | 1 |
| 3.66649151 | 1.87500000 | 0.19789881 | 15 | 12 | 1 |
| 3.66649151 | 1.87500000 | 0.50629985 | 15 | 13 | 1 |
| 3.66649151 | 1.87079906 | 0.85630023 | 15 | 14 | 1 |
| 3.66649151 | 1.85419941 | 1.25629711 | 15 | 15 | 1 |
| 3.66649151 | 1.83329773 | 1.72919846 | 15 | 16 | 1 |
| 3.66649151 | 1.70829868 | 2.17079735 | 15 | 17 | 1 |
| 3.66649151 | 1.25329887 | 2.17709637 | 15 | 18 | 1 |
| 3.66649151 | 0.83330154 | 2.17709732 | 15 | 19 | 1 |
| 3.66649151 | 0.40420246 | 2.17709637 | 15 | 20 | 1 |
| 3.66649151 | 0.0 | 2.17709732 | 15 | 21 | 1 |
| 3.08349705 | 0.0 | -2.30209827 | 16 | 1 | 1 |
| 3.08349705 | 0.34790134 | -2.30209827 | 16 | 2 | 1 |
| 3.08349705 | 0.71380150 | -2.29159641 | 16 | 3 | 1 |
| 3.08349705 | 1.09579945 | -2.26669598 | 16 | 4 | 1 |
| 3.08349705 | 1.48959923 | -2.17289829 | 16 | 5 | 1 |
| 3.08349705 | 1.74999905 | -1.86869526 | 16 | 6 | 1 |
| 3.08349705 | 1.84789744 | -1.46869659 | 16 | 7 | 1 |
| 3.08349705 | 1.87500000 | -1.07499695 | 16 | 8 | 1 |
| 3.08349705 | 1.87499905 | -0.73750007 | 16 | 9 | 1 |
| 3.08349705 | 1.87500000 | -0.42500007 | 16 | 10 | 1 |
| 3.08349705 | 1.87500000 | -0.12500000 | 16 | 11 | 1 |
| 3.08349705 | 1.87500000 | 0.16670033 | 16 | 12 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| 3.08349705 | 1.87500000 | 0.47500032 | 16 | 13 | 1 |
| 3.08349705 | 1.86869812 | 0.82499874 | 16 | 14 | 1 |
| 3.08349705 | 1.84579945 | 1.22289848 | 16 | 15 | 1 |
| 3.08349705 | 1.81250000 | 1.67709732 | 16 | 16 | 1 |
| 3.08349705 | 1.63749886 | 2.04999638 | 16 | 17 | 1 |
| 3.08349705 | 1.20829868 | 2.05209732 | 16 | 18 | 1 |
| 3.08349705 | 0.80210114 | 2.05209732 | 16 | 19 | 1 |
| 3.08349705 | 0.38540173 | 2.05209732 | 16 | 20 | 1 |
| 3.08349705 | 0.0 | 2.05209827 | 16 | 21 | 1 |
| 2.50050259 | 0.0 | -2.23404884 | 17 | 1 | 1 |
| 2.50050259 | 0.33432961 | -2.23404884 | 17 | 2 | 1 |
| 2.50050259 | 0.68965244 | -2.22653294 | 17 | 3 | 1 |
| 2.50050259 | 1.05792713 | -2.20741653 | 17 | 4 | 1 |
| 2.50050259 | 1.44976711 | -2.12919426 | 17 | 5 | 1 |
| 2.50050259 | 1.72084713 | -1.85512447 | 17 | 6 | 1 |
| 2.50050259 | 1.83138752 | -1.46869755 | 17 | 7 | 1 |
| 2.50050259 | 1.87112808 | -1.08856869 | 17 | 8 | 1 |
| 2.50050259 | 1.87500000 | -0.75235219 | 17 | 9 | 1 |
| 2.50050259 | 1.87500000 | -0.43955213 | 17 | 10 | 1 |
| 2.50050259 | 1.87500000 | -0.13955212 | 17 | 11 | 1 |
| 2.50050259 | 1.87500000 | 0.15210140 | 17 | 12 | 1 |
| 2.50050259 | 1.87500000 | 0.46044821 | 17 | 13 | 1 |
| 2.50050259 | 1.86869907 | 0.81044763 | 17 | 14 | 1 |
| 2.50050259 | 1.84486485 | 1.20834541 | 17 | 15 | 1 |
| 2.50050259 | 1.79794693 | 1.64794540 | 17 | 16 | 1 |
| 2.50050259 | 1.57821846 | 1.95382595 | 17 | 17 | 1 |
| 2.50050259 | 1.16243866 | 1.95494556 | 17 | 18 | 1 |
| 2.50050259 | 0.77099133 | 1.95494652 | 17 | 19 | 1 |
| 2.50050259 | 0.37084961 | 1.95494652 | 17 | 20 | 1 |
| 2.50050259 | 0.0 | 1.95494652 | 17 | 21 | 1 |
| 1.83349895 | 0.0 | -2.15347290 | 18 | 1 | 1 |
| 1.83349895 | 0.31716633 | -2.15347290 | 18 | 2 | 1 |
| 1.83349895 | 0.65212643 | -2.14451504 | 18 | 3 | 1 |
| 1.83349895 | 1.00186348 | -2.12660654 | 18 | 4 | 1 |
| 1.83349895 | 1.36593288 | -2.04374504 | 18 | 5 | 1 |
| 1.83349895 | 1.64167434 | -1.83775928 | 18 | 6 | 1 |
| 1.83349895 | 1.76654289 | -1.46454620 | 18 | 7 | 1 |
| 1.83349895 | 1.83141232 | -1.09633350 | 18 | 8 | 1 |
| 1.83349895 | 1.83334923 | -0.76662630 | 18 | 9 | 1 |
| 1.83349895 | 1.83334923 | -0.46142560 | 18 | 10 | 1 |
| 1.83349895 | 1.83334923 | -0.16872585 | 18 | 11 | 1 |
| 1.83349895 | 1.83334923 | 0.11775017 | 18 | 12 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 1.83349895 | 1.83334923 | 0.41772372 | 18 | 13 | 1 |
| 1.83349895 | 1.83019733 | 0.76252383 | 18 | 14 | 1 |
| 1.83349895 | 1.80782986 | 1.14897251 | 18 | 15 | 1 |
| 1.83349895 | 1.72712231 | 1.54792118 | 18 | 16 | 1 |
| 1.83349895 | 1.45575809 | 1.76131058 | 18 | 17 | 1 |
| 1.83349895 | 1.08546829 | 1.78687000 | 18 | 18 | 1 |
| 1.83349895 | 0.72509682 | 1.80042076 | 18 | 19 | 1 |
| 1.83349895 | 0.35207558 | 1.81237003 | 18 | 20 | 1 |
| 1.83349895 | 0.0 | 1.81602192 | 18 | 21 | 1 |
| 1.16650295 | 0.0 | -2.07022381 | 19 | 1 | 1 |
| 1.16650295 | 0.29787934 | -2.06915665 | 19 | 2 | 1 |
| 1.16650295 | 0.61081326 | -2.05946636 | 19 | 3 | 1 |
| 1.16650295 | 0.93213493 | -2.03582191 | 19 | 4 | 1 |
| 1.16650295 | 1.27509117 | -1.96117020 | 19 | 5 | 1 |
| 1.16650295 | 1.55385303 | -1.76150036 | 19 | 6 | 1 |
| 1.16650295 | 1.72566128 | -1.45928860 | 19 | 7 | 1 |
| 1.16650295 | 1.77522087 | -1.11089420 | 19 | 8 | 1 |
| 1.16650295 | 1.78044128 | -0.81106269 | 19 | 9 | 1 |
| 1.16650295 | 1.78044128 | -0.49478799 | 19 | 10 | 1 |
| 1.16650295 | 1.78044128 | -0.21001327 | 19 | 11 | 1 |
| 1.16650295 | 1.78044128 | 0.06855011 | 19 | 12 | 1 |
| 1.16650295 | 1.78044333 | 0.33993721 | 19 | 13 | 1 |
| 1.16650295 | 1.77675629 | 0.69266218 | 19 | 14 | 1 |
| 1.16650295 | 1.74670219 | 1.06272697 | 19 | 15 | 1 |
| 1.16650295 | 1.63500977 | 1.41562653 | 19 | 16 | 1 |
| 1.16650295 | 1.34488392 | 1.56927872 | 19 | 17 | 1 |
| 1.16650295 | 1.00608540 | 1.60303974 | 19 | 18 | 1 |
| 1.16650295 | 0.67638302 | 1.62560558 | 19 | 19 | 1 |
| 1.16650295 | 0.32963387 | 1.64544773 | 19 | 20 | 1 |
| 1.16650295 | 0.0 | 1.65019894 | 19 | 21 | 1 |
| 0.52050543 | 0.0 | -1.98959827 | 20 | 1 | 1 |
| 0.52050543 | 0.27920258 | -1.98749828 | 20 | 2 | 1 |
| 0.52050543 | 0.57080275 | -1.97709560 | 20 | 3 | 1 |
| 0.52050543 | 0.86460030 | -1.94789600 | 20 | 4 | 1 |
| 0.52050543 | 1.13750000 | -1.88119698 | 20 | 5 | 1 |
| 0.52050543 | 1.46879959 | -1.71669674 | 20 | 6 | 1 |
| 0.52050543 | 1.66669941 | -1.45419788 | 20 | 7 | 1 |
| 0.52050543 | 1.72079945 | -1.12499809 | 20 | 8 | 1 |
| 0.52050543 | 1.72919846 | -0.85410005 | 20 | 9 | 1 |
| 0.52050543 | 1.72919941 | -0.52739949 | 20 | 10 | 1 |
| 0.52050543 | 1.72919941 | -0.24999388 | 20 | 11 | 1 |
| 0.52050543 | 1.72919941 | 0.02089977 | 20 | 12 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 0.52050543 | 1.72919941 | 0.26459879 | 20 | 13 | 1 |
| 0.52050543 | 1.72499943 | 0.62499988 | 20 | 14 | 1 |
| 0.52050543 | 1.68750000 | 0.97919953 | 20 | 15 | 1 |
| 0.52050543 | 1.54579926 | 1.28749752 | 20 | 16 | 1 |
| 0.52050543 | 1.23749924 | 1.38329697 | 20 | 17 | 1 |
| 0.52050543 | 0.92920077 | 1.42499733 | 20 | 18 | 1 |
| 0.52050543 | 0.62920201 | 1.45629692 | 20 | 19 | 1 |
| 0.52050543 | 0.30830300 | 1.48329735 | 20 | 20 | 1 |
| 0.52050543 | 0.0 | 1.48959827 | 20 | 21 | 1 |
| -0.14549518 | 0.0 | -1.92709827 | 21 | 1 | 1 |
| -0.14549518 | 0.25000101 | -1.92709827 | 21 | 2 | 1 |
| -0.14549518 | 0.52080262 | -1.92079639 | 21 | 3 | 1 |
| -0.14549518 | 0.78330141 | -1.88539696 | 21 | 4 | 1 |
| -0.14549518 | 1.08329868 | -1.83119678 | 21 | 5 | 1 |
| -0.14549518 | 1.35419941 | -1.69789791 | 21 | 6 | 1 |
| -0.14549518 | 1.55209923 | -1.46459770 | 21 | 7 | 1 |
| -0.14549518 | 1.62500000 | -1.16869831 | 21 | 8 | 1 |
| -0.14549518 | 1.64579868 | -0.87919992 | 21 | 9 | 1 |
| -0.14549518 | 1.64579868 | -0.60830009 | 21 | 10 | 1 |
| -0.14549518 | 1.64579868 | -0.34369928 | 21 | 11 | 1 |
| -0.14549518 | 1.64579868 | -0.06539969 | 21 | 12 | 1 |
| -0.14549518 | 1.64579868 | 0.18130010 | 21 | 13 | 1 |
| -0.14549518 | 1.63749790 | 0.48750013 | 21 | 14 | 1 |
| -0.14549518 | 1.59379864 | 0.81669909 | 21 | 15 | 1 |
| -0.14549518 | 1.41249943 | 1.06039810 | 21 | 16 | 1 |
| -0.14549518 | 1.11460318 | 1.12919712 | 21 | 17 | 1 |
| -0.14549518 | 0.83330125 | 1.16039944 | 21 | 18 | 1 |
| -0.14549518 | 0.57080185 | 1.19169617 | 21 | 19 | 1 |
| -0.14549518 | 0.27920246 | 1.21669769 | 21 | 20 | 1 |
| -0.14549518 | 0.0 | 1.23959732 | 21 | 21 | 1 |
| -0.87449425 | 0.0 | -1.84369659 | 22 | 1 | 1 |
| -0.87449425 | 0.22080147 | -1.84369659 | 22 | 2 | 1 |
| -0.87449425 | 0.45830166 | -1.83959770 | 22 | 3 | 1 |
| -0.87449425 | 0.69999969 | -1.82919693 | 22 | 4 | 1 |
| -0.87449425 | 0.97920042 | -1.78959751 | 22 | 5 | 1 |
| -0.87449425 | 1.23959923 | -1.68119812 | 22 | 6 | 1 |
| -0.87449425 | 1.41669941 | -1.47289848 | 22 | 7 | 1 |
| -0.87449425 | 1.50829887 | -1.21249771 | 22 | 8 | 1 |
| -0.87449425 | 1.53749943 | -0.94789970 | 22 | 9 | 1 |
| -0.87449425 | 1.54999924 | -0.69789958 | 22 | 10 | 1 |
| -0.87449425 | 1.55209923 | -0.44789970 | 22 | 11 | 1 |
| -0.87449425 | 1.54999924 | -0.20619971 | 22 | 12 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| -0.87449425 | 1.54579926 | 0.04579914 | 22 | 13 | 1 |
| -0.87449425 | 1.53129864 | 0.33129978 | 22 | 14 | 1 |
| -0.87449425 | 1.48329926 | 0.63130009 | 22 | 15 | 1 |
| -0.87449425 | 1.26249886 | 0.80829805 | 22 | 16 | 1 |
| -0.87449425 | 0.99590038 | 0.87079906 | 22 | 17 | 1 |
| -0.87449425 | 0.75000101 | 0.89999962 | 22 | 18 | 1 |
| -0.87449425 | 0.51040184 | 0.92709923 | 22 | 19 | 1 |
| -0.87449425 | 0.24580103 | 0.94379902 | 22 | 20 | 1 |
| -0.87449425 | 0.0 | 0.94789791 | 22 | 21 | 1 |
| -1.91649055 | 0.0 | -1.71869755 | 23 | 1 | 1 |
| -1.91649055 | 0.20830166 | -1.71869755 | 23 | 2 | 1 |
| -1.91649055 | 0.42710131 | -1.71669769 | 23 | 3 | 1 |
| -1.91649055 | 0.65830034 | -1.71249866 | 23 | 4 | 1 |
| -1.91649055 | 0.92710066 | -1.68959713 | 23 | 5 | 1 |
| -1.91649055 | 1.17289925 | -1.58749866 | 23 | 6 | 1 |
| -1.91649055 | 1.31669807 | -1.37079906 | 23 | 7 | 1 |
| -1.91649055 | 1.38539886 | -1.11669827 | 23 | 8 | 1 |
| -1.91649055 | 1.41249943 | -0.87499994 | 23 | 9 | 1 |
| -1.91649055 | 1.41669941 | -0.64579958 | 23 | 10 | 1 |
| -1.91649055 | 1.41669941 | -0.41660023 | 23 | 11 | 1 |
| -1.91649055 | 1.41669941 | -0.19579983 | 23 | 12 | 1 |
| -1.91649055 | 1.41669941 | 0.03540039 | 23 | 13 | 1 |
| -1.91649055 | 1.41669941 | 0.30420023 | 23 | 14 | 1 |
| -1.91649055 | 1.35829926 | 0.56880003 | 23 | 15 | 1 |
| -1.91649055 | 1.15829945 | 0.73329931 | 23 | 16 | 1 |
| -1.91649055 | 0.92500049 | 0.80420017 | 23 | 17 | 1 |
| -1.91649055 | 0.69790053 | 0.83960056 | 23 | 18 | 1 |
| -1.91649055 | 0.47080117 | 0.86040020 | 23 | 19 | 1 |
| -1.91649055 | 0.22920233 | 0.87499905 | 23 | 20 | 1 |
| -1.91649055 | 0.0 | 0.88539886 | 23 | 21 | 1 |
| -3.20848370 | 0.0 | -1.57289791 | 24 | 1 | 1 |
| -3.20848370 | 0.19750107 | -1.57289791 | 24 | 2 | 1 |
| -3.20848370 | 0.38750029 | -1.56869888 | 24 | 3 | 1 |
| -3.20848370 | 0.59580070 | -1.56249905 | 24 | 4 | 1 |
| -3.20848370 | 0.84380066 | -1.54579830 | 24 | 5 | 1 |
| -3.20848370 | 1.07289886 | -1.45419884 | 24 | 6 | 1 |
| -3.20848370 | 1.19579837 | -1.25209713 | 24 | 7 | 1 |
| -3.20848370 | 1.23539925 | -1.01039791 | 24 | 8 | 1 |
| -3.20848370 | 1.24169922 | -0.79160017 | 24 | 9 | 1 |
| -3.20848370 | 1.25000000 | -0.58539951 | 24 | 10 | 1 |
| -3.20848370 | 1.25000000 | -0.38539970 | 24 | 11 | 1 |
| -3.20848370 | 1.25000000 | -0.18959993 | 24 | 12 | 1 |

C-3

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | |
|---|---|---|---|---|---|
| -3.20848370 | 1.25000000 | 0.01039386 | 24 | 13 | 1 |
| -3.20848370 | 1.23959923 | 0.23960018 | 24 | 14 | 1 |
| -3.20848370 | 1.16249943 | 0.46040052 | 24 | 15 | 1 |
| -3.20848370 | 1.03129864 | 0.63539386 | 24 | 16 | 1 |
| -3.20848370 | 0.84790009 | 0.73540020 | 24 | 17 | 1 |
| -3.20848370 | 0.64170063 | 0.77089977 | 24 | 18 | 1 |
| -3.20848370 | 0.43330061 | 0.78960037 | 24 | 19 | 1 |
| -3.20848370 | 0.21250153 | 0.79789925 | 24 | 20 | 1 |
| -3.20848370 | 0.0 | 0.80210018 | 24 | 21 | 1 |
| -4.62448120 | 0.0 | -1.40619659 | 25 | 1 | 1 |
| -4.62448120 | 0.17500043 | -1.40619659 | 25 | 2 | 1 |
| -4.62448120 | 0.35000074 | -1.39999771 | 25 | 3 | 1 |
| -4.62448120 | 0.53750068 | -1.39169788 | 25 | 4 | 1 |
| -4.62448120 | 0.75000012 | -1.36459827 | 25 | 5 | 1 |
| -4.62448120 | 0.92709994 | -1.25829792 | 25 | 6 | 1 |
| -4.62448120 | 1.02499866 | -1.07919788 | 25 | 7 | 1 |
| -4.62448120 | 1.06250000 | -0.87079990 | 25 | 8 | 1 |
| -4.62448120 | 1.07289886 | -0.68539989 | 25 | 9 | 1 |
| -4.62448120 | 1.08329868 | -0.50419986 | 25 | 10 | 1 |
| -4.62448120 | 1.08329868 | -0.33329964 | 25 | 11 | 1 |
| -4.62448120 | 1.08329868 | -0.16660029 | 25 | 12 | 1 |
| -4.62448120 | 1.08329868 | 0.01039881 | 25 | 13 | 1 |
| -4.62448120 | 1.07499886 | 0.20839965 | 25 | 14 | 1 |
| -4.62448120 | 1.02499866 | 0.41040033 | 25 | 15 | 1 |
| -4.62448120 | 0.92080009 | 0.57920074 | 25 | 16 | 1 |
| -4.62448120 | 0.77080017 | 0.68750095 | 25 | 17 | 1 |
| -4.62448120 | 0.58330071 | 0.72080040 | 25 | 18 | 1 |
| -4.62448120 | 0.39580059 | 0.73750019 | 25 | 19 | 1 |
| -4.62448120 | 0.19170028 | 0.73960018 | 25 | 20 | 1 |
| -4.62448120 | 0.0 | 0.73959923 | 25 | 21 | 1 |
| -6.10449219 | 0.0 | -1.23959923 | 26 | 1 | 1 |
| -6.10449219 | 0.15000027 | -1.23329926 | 26 | 2 | 1 |
| -6.10449219 | 0.30420035 | -1.22289848 | 26 | 3 | 1 |
| -6.10449219 | 0.46670038 | -1.20209789 | 26 | 4 | 1 |
| -6.10449219 | 0.63330013 | -1.16249943 | 26 | 5 | 1 |
| -6.10449219 | 0.78129995 | -1.07079792 | 26 | 6 | 1 |
| -6.10449219 | 0.83540245 | -0.91039938 | 26 | 7 | 1 |
| -6.10449219 | 0.87500000 | -0.73540014 | 26 | 8 | 1 |
| -6.10449219 | 0.88539973 | -0.57919979 | 26 | 9 | 1 |
| -6.10449219 | 0.89169979 | -0.43329984 | 26 | 10 | 1 |
| -6.10449219 | 0.89579976 | -0.29160017 | 26 | 11 | 1 |
| -6.10449219 | 0.89579976 | -0.14999956 | 26 | 12 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| -6.10449219 | 0.89579976 | -0.00619990 | 26 | 13 | 1 |
| -6.10449219 | 0.88750017 | 0.15629959 | 26 | 14 | 1 |
| -6.10449219 | 0.85830081 | 0.33129978 | 26 | 15 | 1 |
| -6.10449219 | 0.78130150 | 0.48750114 | 26 | 16 | 1 |
| -6.10449219 | 0.65630150 | 0.57499981 | 26 | 17 | 1 |
| -6.10449219 | 0.51040018 | 0.62500095 | 26 | 18 | 1 |
| -6.10449219 | 0.34580040 | 0.64170074 | 26 | 19 | 1 |
| -6.10449219 | 0.16670001 | 0.65000153 | 26 | 20 | 1 |
| -6.10449219 | 0.0 | 0.65630054 | 26 | 21 | 1 |
| -7.60449219 | 0.0 | -1.05209923 | 27 | 1 | 1 |
| -7.60449219 | 0.13329995 | -1.04160023 | 27 | 2 | 1 |
| -7.60449219 | 0.26460022 | -1.01869869 | 27 | 3 | 1 |
| -7.60449219 | 0.38540006 | -0.97290134 | 27 | 4 | 1 |
| -7.60449219 | 0.50000095 | -0.90620041 | 27 | 5 | 1 |
| -7.60449219 | 0.59380150 | -0.81250095 | 27 | 6 | 1 |
| -7.60449219 | 0.65830135 | -0.69999945 | 27 | 7 | 1 |
| -7.60449219 | 0.69169998 | -0.57079977 | 27 | 8 | 1 |
| -7.60449219 | 0.70420170 | -0.44789964 | 27 | 9 | 1 |
| -7.60449219 | 0.70830154 | -0.33329952 | 27 | 10 | 1 |
| -7.60449219 | 0.70830154 | -0.21869940 | 27 | 11 | 1 |
| -7.60449219 | 0.70830154 | -0.11040026 | 27 | 12 | 1 |
| -7.60449219 | 0.70830154 | 0.00629991 | 27 | 13 | 1 |
| -7.60449219 | 0.69790077 | 0.13329899 | 27 | 14 | 1 |
| -7.60449219 | 0.65630150 | 0.25829893 | 27 | 15 | 1 |
| -7.60449219 | 0.60000134 | 0.37920094 | 27 | 16 | 1 |
| -7.60449219 | 0.51670170 | 0.46670151 | 27 | 17 | 1 |
| -7.60449219 | 0.41670007 | 0.53540230 | 27 | 18 | 1 |
| -7.60449219 | 0.29790002 | 0.58130074 | 27 | 19 | 1 |
| -7.60449219 | 0.14579999 | 0.60830021 | 27 | 20 | 1 |
| -7.60449219 | 0.0 | 0.61460114 | 27 | 21 | 1 |
| -9.33349609 | 0.0 | -0.86460114 | 28 | 1 | 1 |
| -9.33349609 | 0.10830015 | -0.85830116 | 28 | 2 | 1 |
| -9.33349609 | 0.20829993 | -0.80620098 | 28 | 3 | 1 |
| -9.33349609 | 0.29170001 | -0.74790096 | 28 | 4 | 1 |
| -9.33349609 | 0.35420132 | -0.66660213 | 28 | 5 | 1 |
| -9.33349609 | 0.40630054 | -0.58329964 | 28 | 6 | 1 |
| -9.33349609 | 0.41370117 | -0.49579948 | 28 | 7 | 1 |
| -9.33349609 | 0.46250153 | -0.41249925 | 28 | 8 | 1 |
| -9.33349609 | 0.47500134 | -0.33329958 | 28 | 9 | 1 |
| -9.33349609 | 0.48960114 | -0.25209993 | 28 | 10 | 1 |
| -9.33349609 | 0.48960114 | -0.17710012 | 28 | 11 | 1 |
| -9.33349609 | 0.48960114 | -0.10209942 | 28 | 12 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

```
 -9.33349609   0.48960114  -0.32289963   28   13    1
 -9.33349609   0.48330116   0.06669986   28   14    1
 -9.33349609   0.46880150   0.16250134   28   15    1
 -9.33349609   0.43330193   0.25630093   28   16    1
 -9.33349609   0.38750172   0.33340073   28   17    1
 -9.33349609   0.31869972   0.39790358   28   18    1
 -9.33349609   0.23330003   0.45209885   28   19    1
 -9.33349609   0.11670005   0.50210094   28   20    1
 -9.33349609   0.0          0.51039982   28   21    1
-12.20849609   0.0          0.09379959   29    1    1
-12.20849609   0.0          0.09379959   29    2    1
-12.20849609   0.0          0.09379959   29    3    1
-12.20849609   0.0          0.09379959   29    4    1
-12.20849609   0.0          0.09379959   29    5    1
-12.20849609   0.0          0.09379959   29    6    1
-12.20849609   0.0          0.09379959   29    7    1
-12.20849609   0.0          0.09379959   29    8    1
-12.20849609   0.0          0.09379959   29    9    1
-12.20849609   0.0          0.09379959   29   10    1
-12.20849609   0.0          0.09379959   29   11    1
-12.20849609   0.0          0.09379959   29   12    1
-12.20849609   0.0          0.09379959   29   13    1
-12.20849609   0.0          0.09379959   29   14    1
-12.20849609   0.0          0.09379959   29   15    1
-12.20849609   0.0          0.09379959   29   16    1
-12.20849609   0.0          0.09379959   29   17    1
-12.20849609   0.0          0.09379959   29   18    1
-12.20849609   0.0          0.09379959   29   19    1
-12.20849609   0.0          0.09379959   29   20    1
-12.20849609   0.0          0.09379959   29   21    1
                                                      0      END OF BODY DATA
          0.0          0.0         0.0     0    0              0.0     0     0
                   0.0                       0.0
    0         0         0         0         0
OPERATOR - PLEASE USE PLOTTING FORMS 0120 _
X Z OUT      45.       10.       30.                       6.5   ORT                 0
X Z OUT       0.        0.        0.                       6.5   ORT                 1
N             1         2
N             4         1
N             5         3
END
X Z OUT      45.       10.       30.                       6.5   ORT                 0
X Z OUT       0.        0.        0.                       6.5   ORT                 1
    0         0         0         0         0    0    0    0    0      TERMINATION CARD
```

# SAMPLE OUTPUT - GRIDPLOT

ORIGINAL PAGE IS OF POOR QUALITY

ADDITIONAL ARBITRARY STATION SPECIFICATIONS

| DEPENDENT STATION | M-STATIONS 1ST HALF | M-STATIONS 2ND HALF | N-STATIONS 1ST HALF | N-STATIONS 2ND HALF |
|---|---|---|---|---|
| 1 | 0 | 0 | 2 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 3 | 0 |
| 6 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 12 | | | 0 | 0 |
| 13 | | | 0 | 0 |
| 14 | | | 0 | 0 |
| 15 | | | 0 | 0 |

NEW DATA...

MMAX = 41, NMAX = 35 ----> 1360 QUADRILATERALS

| XI | YI | ZI | MI | NI | NB |
|---|---|---|---|---|---|
| 12.20020440 | 0.0 | -0.42710010 | 1 | 1 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 2 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 3 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 4 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 5 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 6 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 7 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 8 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 9 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 10 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 11 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 12 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 13 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 14 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 15 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 16 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 17 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 18 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 19 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 20 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 21 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 22 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 23 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 24 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 25 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 26 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 27 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 28 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 29 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 30 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 31 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 32 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 33 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 34 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 35 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 36 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 37 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 38 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 39 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 40 | 1 |
| 12.20020440 | 0.0 | -0.42710010 | 1 | 41 | 1 |
| . | | . | | | . |
| . | | . | | | . |
| . | | . | | | . |
| -9.22040440 | 0.11000000 | 0.00210094 | 34 | 39 | 1 |
| -9.22040440 | 0.00002219 | 0.00727768 | 34 | 40 | 1 |
| -9.22040440 | 0.0 | 0.01000077 | 34 | 41 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 1 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 2 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 3 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 4 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 5 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 6 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 7 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 8 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 9 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 10 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 11 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 12 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 13 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 14 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 15 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 16 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 17 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 18 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 19 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 20 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 21 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 22 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 23 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 24 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 25 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 26 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 27 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 28 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 29 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 30 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 31 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 32 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 33 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 34 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 35 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 36 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 37 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 38 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 39 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 40 | 1 |
| -12.20020440 | 0.0 | 0.09370004 | 35 | 41 | 1 |

ORIGINAL PAGE IS
OF POOR QUALITY

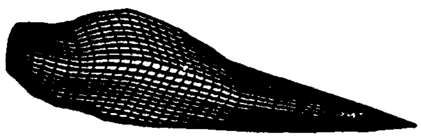
CONFIGURATION DESCRIPTION

NEW BODY WITH IMPROVED GRID

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.0 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | | | | | |
| | | | | | | | | | |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | | | | | | | | | |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | | | | | | | | | |
| | | | | | | | | | |
| 0.0 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 0.0 | | | | | | | | | |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | | | | | | | | | |
| . | | | | . | | | | | . |
| . | | | | . | | | | | . |
| . | | | | . | | | | | . |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | | | | | | | | | |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | | | | | | | | | |

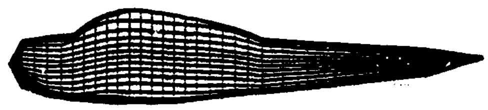
PLOT DATA

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| X Z OUT | 40.00000 | 10.00000 | 30.00000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | [illegible] | 0 |
| X Z OUT | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | [illegible] | 1 |



CESSNA 182 FUSELAGE (N=20, M=21) POINT MODIFIED

X Z OUT 40. 10. 30. 6.5 ORT 0



CESSNA 182 FUSELAGE (N=20, M=21) POINT MODIFIED

X Z OUT 0. 0. 0. 6.5 ORT 1

NEW BODY WITH IMPROVED GRID

X Z OUT 40. 10. 30. 6.5 ORT 0

NEW BODY WITH IMPROVED GRID

X Z OUT 0. 0. 0. 6.5 ORT 1

# REFERENCES


1. Fox, S. R.; and Smetana, F. O.: "A Study of Optimum Cowl Shapes and Flow Port Locations for Minimum Drag with Effective Engine Cooling--Volume I." NASA CR-159379, November 1980.

2. Hess, J. L.; and Smith, A. M. O.: "Calculation of Potential Flow About Arbitrary Bodies." Progress in Aeronautical Sciences, Vol. 8, pp. 1-138, Pergamon Press, 1967.

3. Smetana, F. O.; Summey, D. C.; Smith, N. S.; and Carden, R. K.: "Light Aircraft Lift, Drag, and Moment Prediction--A Review and Analysis." NASA CR-2523, May 1975, 480 pages.