

NASA CR  
150944

R-9808

ORBITAL MANEUVERING ENGINE  
FEED SYSTEM COUPLED STABILITY INVESTIGATION  
COMPUTER USER'S MANUAL

NAS9-14315

1 September 1975

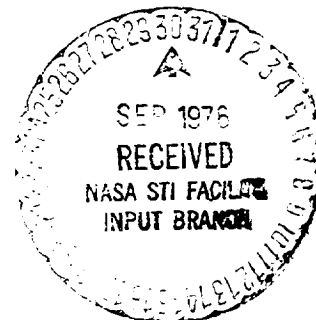


Rockwell International

(NASA-CR-150944) ORBITAL MANEUVERING ENGINE  
FEED SYSTEM COUPLED STABILITY INVESTIGATION,  
COMPUTER USER'S MANUAL (Rockwell  
International Corp., Canoga Park) 382 p HC  
\$10.75 CSCL 21H G3/20

N76-31281

Unclas  
02472







**Rockwell International**

**Rocketdyne Division**  
6633 Canoga Avenue  
Canoga Park, California 91304

R-9808

ORBITAL MANEUVERING ENGINE  
FEED SYSTEM COUPLED STABILITY INVESTIGATION  
COMPUTER USER'S MANUAL

NAS9-14315

1 September 1975

PREPARED FOR

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LYNDON B. JOHNSON SPACE CENTER  
Houston, Texas 77058

**PREPARED BY**

M. D. Schuman  
K. W. Fertig  
J. K. Hunting  
D. R. Kahn

**APPROVED BY**

*for* *D. R. Kahn*  
J. A. Nestlerode  
Principal Engineer  
OME/Feed System Coupled  
Stability Program

*F. M. Kirby*  
*for* R. D. Paster  
Program Manager  
SS/APS Programs



## FOREWORD

This document was prepared by Rocketdyne Division, Rockwell International Corporation, in accordance with Article I and Line Item Nos. 2, 3, and 4 of the Data Requirements List of Contract NAS9-14315 with the National Aeronautics and Space Administration. The contract period of performance was 6 August 1974 to 1 September 1975. The NASA/JSC Technical Monitor was Mr. F. D. Freeburn. The Rocketdyne Program Manager was Mr. R. H. Hesel for the first three months; he was replaced by Mr. R. D. Paster for the remainder of the program. Mr. J. A. Nestlerode served as the Principal Engineer, assisted by Dr. D. R. Kahn.

Several technical people at Rocketdyne performed work or served as consultants regarding specific areas of the various program tasks: Mssrs. J. K. Hunting, R. L. Nelson, and L. E. Sack with respect to the feed system hydrodynamics, Mr. F. R. Linow with respect to combustion dynamics, Mr. M. D. Schuman with respect to combustion dynamics, chamber dynamics, engineering model formulation, and computer programming, and Mr. K. W. Fertig with respect to numerical analysis, computer programming, and checkout.

## ABSTRACT

This report is an operating manual for the Feed System Coupled Stability Model. It is submitted as partial fulfillment of an 11-month program designed to develop, verify, and document a digital computer model that can be used to analyze and predict engine/feed system coupled instabilities in pressure-fed storable propellant propulsion systems over a frequency range of 10 to 1000 Hz.

The first section describes the analytical approach to modeling the feed system hydrodynamics, combustion dynamics, chamber dynamics, and overall engineering model structure, and presents the governing equations in each of the technical areas. This is followed by the Program User's Guide, which is a complete description of the structure and operation of the computerized model. Last, appendixes provide an alphabetized FORTRAN symbol table, detailed program logic diagrams, computer code listings, and sample case input and output data listings.



## CONTENTS

Introduction . . . . .	1
Mathematical Formulation of Model . . . . .	3
Introduction . . . . .	3
Feed System Dynamics . . . . .	5
Combustion Dynamics . . . . .	22
Chamber Dynamics . . . . .	35
Engineering Model . . . . .	51
Program User's Guide . . . . .	59
FSCSM Main Program . . . . .	59
SØLVW . . . . .	62
CHAMDY . . . . .	65
XIMAGF . . . . .	65
CHMCØN . . . . .	68
CØMMAT . . . . .	68
ZERØ . . . . .	68
Steady . . . . .	68
COMBDY . . . . .	68
Area . . . . .	71
LØFAC . . . . .	71
Head . . . . .	71
HYDRDY . . . . .	71
FRESP . . . . .	76
CØGAEL . . . . .	80
TDPLØT . . . . .	80
NØZADM, (RKTDIF, RKTZ, RKZDIF, TADAMS, ZADAMS) . . . . .	82
FSCSM Computer Model Input . . . . .	84
Hydrodynamic Input . . . . .	102
Program Output . . . . .	110
Program Operation . . . . .	112
References . . . . .	121
<u>Appendix A</u> . . . . .	A-1
Fortran Symbol Table . . . . .	A-1

<u>Appendix B</u> . . . . .	B-1
Program Flow Charts . . . . .	B-1
<u>Appendix C</u> . . . . .	C-1
Computer Code Listings . . . . .	C-1
<u>Appendix D</u> . . . . .	D-1
Sample Case Input . . . . .	D-1
<u>Appendix E</u> . . . . .	E-1
Sample Case Output . . . . .	E-1



## ILLUSTRATIONS

1. Differential Pressures Developed Across the Incremental Length of a Fluid Element . . . . .	5
2. Generalized Line Segment . . . . .	7
3. Schematic of the Injector as a Lumped Compressible Volume . . . . .	10
4. Generalized OME Feed System Schematic . . . . .	15
5. Logic Diagram for FSCM Main Program . . . . .	60
6. Logic Diagram for Subroutine SØLVW . . . . .	63
7. Logic Diagram for Subroutine CHAMDY . . . . .	66
8. Logic Diagram for Subroutine XIMAFG . . . . .	67
9. Logic Diagram for Subroutine STEADY . . . . .	69
10. Logic Diagram for Subprogram CØMBDY . . . . .	70
11. Logic Diagram for Subprogram HYDRDY . . . . .	72
12. Logic Diagram for Subroutine FRESP . . . . .	77
13. Logic Diagram for Subroutine NØZADM . . . . .	83
14. Percent Unburned Propellant as a Function of Distance from Injector Face . . . . .	95
15. Chamber and Nozzle Geometry . . . . .	98
16. FSCSM Program Overlay Structure . . . . .	114

## TABLES

1. Hydrodynamic Equations . . . . .	16
2. FSCSM Input Data . . . . .	85
3. Name List/Hyd/Data Input Names . . . . .	106



REVISION PAGE

**PRECEDING PAGE BLANK NOT FILMED**

R-9808/ix



## INTRODUCTION

Historically, during the development of pressure-fed propulsion systems, feed system/engine coupled instabilities have been frequently encountered. Resolution of these problems usually included increasing injector pressure drop to decouple the feed system from the combustor, the result being substantial system weight penalties. A dynamic computer model would be a useful tool in obviating coupled stability problems during the development of the Space Shuttle Orbit Maneuvering System (SS/OMS). A model could be used both as a system design tool to optimize component location and pressure profile (minimize system weight) and a system development tool to define test programs for assessing stability margins of the OMS.

This document is an operating manual for the Feed System Coupled Stability Model (FSCSM) and is submitted as partial fulfillment of an 11-month program conducted by Rocketdyne to develop and verify an engineering digital computer model for the NASA/JSC which can be used to analyze feed system/engine coupled instabilities in pressure-fed, storable propellant, propulsion systems over a frequency range of 10 to 1000 Hz (frequencies lower than the chamber transverse frequencies). The model is sufficiently general so that it may be readily applicable to present and future engine and propulsion programs. For scaling purposes, the baseline configuration chosen is the OMS engine. The model has been written for use on the NASA/JSC Univac 1110, EXEC-8 computer system, and provides NASA a tool which can be used to:

1. Conduct preliminary design tradeoff for feasibility studies prior to propulsion concept selection.
2. Guide the design of propulsion systems to ensure stability at all operating ranges and with minimum penalties.
3. Guide testing programs by predicting the least stable operating regimes thereby reducing the number of stability tests required.
4. Provide stability verification in the event system changes are made and hot-fire verification is impractical.
5. Diagnose problems on existing systems and evaluate potential solutions.

The work performed in completing the requirements of the program's technical effort is described in a separate companion document, entitled OME Feed System-Coupled Stability Model, Final Report (Ref. 1). This includes the mathematical formulation of the model, development of the model into an overall engineering structure, and verification of the model's operation and capabilities by comparing the model's theoretical predictions with experimental data from an OMS engine and test rig with known feed system/engine chugging history.

The present document contains a detailed description of the structure and operation of the FSCSM. In the first section, the mathematical formulation of the model is reviewed. The analytical approach to modeling the feed system hydrodynamics, combustion dynamics, chamber dynamics, and overall engineering structure is described and the equations utilized by the model in each of the technical areas are presented. The reader may consult Ref. 1 for more details pertaining to the derivation of the equations.

The Program User's Guide section contains the instructions necessary to operate the computer model and interpret the results. First, the structure and logic of the main program and all subroutines are described, followed by a description of the input data required to run FSCSM. The input is divided into four major sections: (1) main control, (2) nozzle admittance control, (3) hydrodynamics control, and (4) combustion dynamics control. The format, content, and description of each input data card is clearly specified for each control section. The output of the FSCSM computer program is then discussed in terms of each tabular page of printout. Finally, additional details on program operation are presented, including program size, overlay structure, computer time, and program input/output data set file information. Appendixes provide an alphabetized FORTRAN symbol table, detailed program logic diagrams, computer code listings, and sample case input and output data listings.

## MATHEMATICAL FORMULATION OF MODEL

### INTRODUCTION

During certain periods of a rocket engine's operation, conditions within the combustion chamber and feed system are time variant, i.e., the operation is not steady with respect to time. Prime interest of this computer model is focused on abnormal transient operation during unstable combustion, i.e., pressure oscillations in a combustion device which are driven by the feed system and sustained by the combustion process. Start and stop transients are not considered.

The deviations from steady-state combustion which occur during unstable burning depend upon the kind of instability experienced. Liquid rocket instabilities are classified according to their dominant time-varying processes. They may be divided initially into two categories, depending upon whether the instability oscillation wave length is long or short compared with the chamber dimensions.

If the instability wave length is considerably longer than the chamber length and diameter, pressure disturbances propagate rapidly through the combustion space compared with rates of change due to the instability. As a result, wave motion in the chamber may be neglected and chamber pressure can be considered to vary only with time but not to vary spatially (i.e.,  $P_c$  is a lumped parameter). These instabilities depend upon a fluid mechanical coupling between the propellant feed system(s) dynamics (fluctuating injection rates), the propellant combustion rates (delay times), and the combustion gas exhaust rates (pressure relaxation). Such instabilities can be further subdivided into various categories depending on the extent of wave motion in the feed system.

The breakpoint at which chamber wave motion becomes important is not abrupt. In reality, chamber wave motion is always present and, in effect, lumped chamber instabilities are really "zero order mode" limits of more general wave motion instabilities. In practice, it is found that the chamber gases

can be considered to act as a lump until the frequency of oscillation exceeds roughly one-fourth of the frequency of the lowest chamber acoustic resonance mode. At and above such frequencies wave motion becomes important and cannot be neglected in analysis. Chamber wave motion instabilities are characterized by the wave-length of the oscillatory motion being comparable to the chamber dimensions. As with lumped chamber instabilities, the driving energy comes from oscillatory spray combustion. With wave motion instabilities, however, in addition to the effects of injection rate fluctuations, there is the combustion response of burning propellant sprays as they are disturbed by passage of a pressure wave through them. Wave motion may increase local burning rates by any of several mechanisms: (1) a pressure effect on the drop vapor gas phase burning rates; (2) enhanced mixing between gases and between sprays and gases; and (3) increased spray gasification rates. Increased spray gasification may be due to transient increases in convective flow velocities, to increased temperature or concentration gradients, and/or spray droplet shattering. The instability amplitude depends upon the magnitude of the response, and vice versa; typically, the interacting processes are driven to a limit represented by abrupt, essentially complete consumption of the propellant sprays. This direct response can be so great that injection rate fluctuations may be of secondary importance. As a result this class of instability can also be further subdivided as to the importance of feed system coupling. In the absence of feed system coupling, the instability is referred to as "classical acoustic instability." Only longitudinal chamber modes with feed system coupled instabilities are considered in this program.



## FEED SYSTEM DYNAMICS

### Development of the Waterhammer Equations

Consider the differential control volume of a fluid element in a duct shown in Fig. 1.

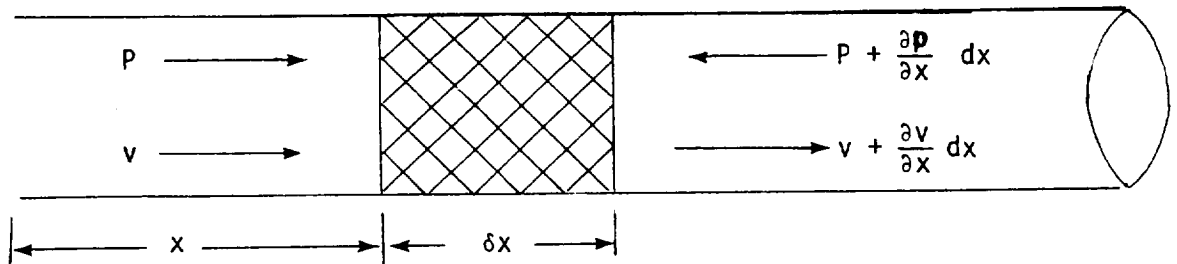


Fig. 1. Differential Pressures Developed Across the Incremental Length of a Fluid Element

Fluid compressibility and Newton's second law leads to the following pair of differential equations:

$$\frac{\partial p}{\partial t} = -\beta \frac{\partial v}{\partial x} = -c^2 \rho \frac{\partial v}{\partial x} \quad (1)$$

$$\frac{\partial p}{\partial x} = -\rho \frac{\partial v}{\partial t} = -\frac{\beta}{c^2} \frac{\partial v}{\partial t} \quad (2)$$

where

$p$  = fluid pressure

$v$  = fluid velocity

$\beta$  = fluid bulk modulus

$\rho$  = fluid density

$c$  = acoustic velocity =  $(\beta/\rho)^{1/2}$

There are several ways in which to solve these equations. The solution method presented here follows that of Ezekiel (Ref. 2). The general form of the solution that satisfies either of equations (1) and (2) is

$$p = F_1 \left( t + \frac{x}{c} \right) + F_2 \left( t - \frac{x}{c} \right) \quad (3)$$

where  $F_1$  and  $F_2$  are arbitrary functions.

Taking the partial derivative of p with respect to x and t separately and substituting the results in equations ( 1) and ( 2) gives:

$$\frac{\partial v}{\partial x} = - \frac{1}{\beta} \frac{\partial p}{\partial t} = - \frac{1}{\beta} \left[ F_1' \left( t + \frac{x}{c} \right) + F_2' \left( t - \frac{x}{c} \right) \right] \quad ( 4 )$$

$$\frac{\partial v}{\partial t} = - \frac{1}{\rho} \frac{\partial p}{\partial x} = - \frac{1}{\rho c} \left[ F_1' \left( t + \frac{x}{c} \right) - F_2' \left( t - \frac{x}{c} \right) \right] \quad ( 5 )$$

where

$$F' (\xi) = \frac{\partial F (\xi)}{\partial \xi} .$$

The expression for v is obtained from either equation ( 4) or ( 5):

$$zv = - F_1 \left( t + \frac{x}{c} \right) + F_2 \left( t - \frac{x}{c} \right) \quad ( 6 )$$

where

$$z \equiv (\rho\beta)^{\frac{1}{2}} . \quad ( 7 )$$

Letting the subscript o denote x=0, the upstream position, and the subscript L denote x=L, the downstream position, and defining  $\tau = L/c$  as the signal propagation time between the two positions, equations ( 3) and ( 6) become

$$p_o = F_1(t) + F_2(t) \quad ( 8 )$$

$$p_L = F_1(t+\tau) + F_2(t-\tau) \quad ( 9 )$$

$$zv_o = - F_1(t) + F_2(t) \quad ( 10 )$$

$$zv_L = - F_1(t+\tau) + F_2(t-\tau) \quad ( 11 )$$

Combining Eqs. ( 8) and ( 10), and Eqs. ( 9) and ( 11) separately, yields four additional relations:

$$p_o + zv_o = 2 F_2(t) \quad ( 12 )$$

$$p_o - zv_o = 2 F_1(t) \quad ( 13 )$$

$$p_L + z v_L = 2 F_2(t-\tau) \quad (14)$$

$$p_L - z v_L = 2 F_1(t+\tau). \quad (15)$$

Eliminating the functions  $F_1$  and  $F_2$  gives the final result as:

$$\left[ p_o + z v_o \right]_{(t-\tau)} = p_L + z v_L \quad (16)$$

$$\left[ p_L - z v_L \right]_{(t-\tau)} = p_o - z v_o \quad (17)$$

Consider now Fig. 2, which depicts a generalized line segment forming a portion of a feed system with many such segments.

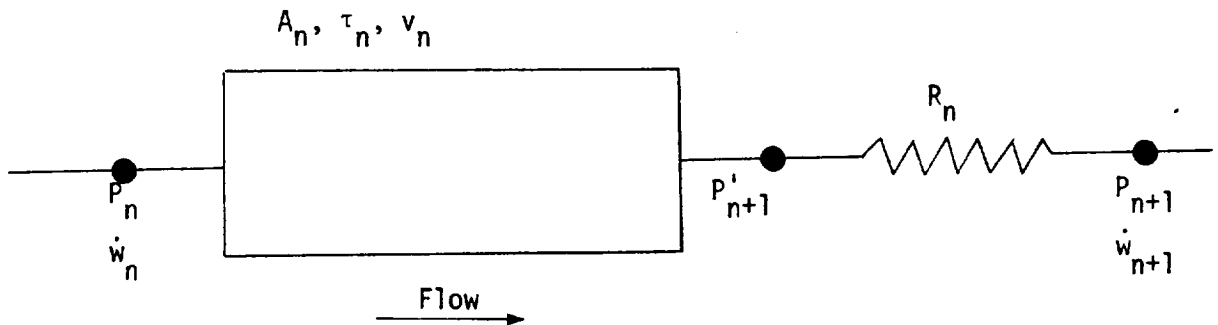


Figure 2. Generalized Line Segment

The equations which describe the pressure and flows as functions of time and of each other for the generalized line segment are obtained from Eqs. (16) and (17):

$$p_n - \left( \frac{v_n}{A_n g} \right) \dot{w}_n = \left[ p'_{n+1} - \left( \frac{v_n}{A_n g} \right) \dot{w}_{n+1} \right]_{(t-\tau_n)} \quad (18)$$

$$p'_{n+1} + \left( \frac{v_n}{A_n g} \right) \dot{w}_{n+1} = \left[ p_n + \left( \frac{v_n}{A_n g} \right) \dot{w}_n \right]_{(t-\tau_n)} \quad (19)$$

The expression,  $t - \tau_n$ , indicates values at  $\tau_n$  seconds before, and

$$p'_{n+1} = p_{n+1} + R_n |\dot{w}_n| \dot{w}_n \quad (20)$$

$$\dot{w}_n = \rho_n A_n v_n \quad (21)$$

Equations (18) and (19) are solutions of the wave equation, and equation (20) is the flow through a nonlinear fluid resistance. Letting

$$\alpha_n = \frac{v_n}{A_n g} \quad (22)$$

these equations can be combined to give:

$$p_n - \alpha_n \dot{w}_n = \left[ p_{n+1} + R_n |\dot{w}_{n+1}| (\dot{w}_{n+1} - \alpha_n) \right]_{(t-\tau_n)} \quad (23)$$

$$p_n + R_{n-1} |\dot{w}_n| \dot{w}_n + \alpha_{n-1} \dot{w}_n = \left[ p_{n-1} + \alpha_{n-1} \dot{w}_{n-1} \right]_{(t-\tau_{n-1})} \quad (24)$$

Eliminating  $p_n$  and rearranging into quadratic form results in

$$R_{n-1} \dot{w}_n^2 + (\alpha_{n-1} + \alpha_n) \dot{w}_n - \left[ p_{n-1} + \alpha_{n-1} \dot{w}_{n-1} \right]_{(t-\tau_{n-1})} + \left[ p_{n+1} + R_n |\dot{w}_{n+1}| (\dot{w}_{n+1} - \alpha_n) \right]_{(t-\tau_{n-1})} = 0 \quad (25)$$

which can be solved for the appropriate solution using the quadratic formula. The tank end parameters are obtained using a solution of Eq. (23) only. The injector end solution is obtained using the quadratic formula for equation (25).

The linear model incorporated in the Hydrodynamics subprogram utilizes the same basic equations, (23) and (24), but in the following linearized form:

$$(\delta p_n) - \alpha_n (\delta \dot{w}_n) = \left[ (\delta p'_{n+1}) - \alpha_n (\delta \dot{w}_{n+1}) \right]_{(t-\tau_n)} \quad (26)$$

$$(\delta p'_{n+1}) + \alpha_n (\delta \dot{w}_{n+1}) = \left[ (\delta p_n) + \alpha_n (\delta \dot{w}_n) \right]_{(t-\tau_n)} , \quad (27)$$

where

$$(\delta p'_{n+1}) = (\delta p_{n+1}) + 2R_n \bar{w}_{n+1} (\delta \dot{w}_{n+1}) . \quad (28)$$

These equations are then combined, resulting in

$$\alpha_n (\delta \dot{w}_n) - (\delta p_n) + \left[ (\delta p_{n+1}) + (\bar{R}_n - \alpha_n) (\delta \dot{w}_{n+1}) \right]_{(t-\tau_n)} = 0 \quad (29)$$

$$(\bar{R}_n + \alpha_n) (\delta \dot{w}_{n+1}) + (\delta p_{n+1}) - \left[ (\delta p_n) + \alpha_n (\delta \dot{w}_n) \right]_{(t-\tau_n)} = 0 , \quad (30)$$

where

$$\bar{R} = 2 R_N \bar{w}_{n+1} \quad (31)$$

At the tank end, the term  $\delta p_n$  is zero for constant tank pressure. At the injector end,  $\delta p_{n+1}$  is the independent variable.

### Injector Dynamics

The injector dynamics are included by treating the injector as a lumped compressible volume as shown in the figure below.

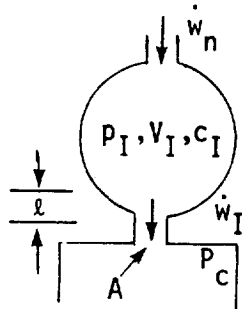


Figure 3. Schematic of the Injector as a Lumped Compressible Volume

The pressure in the injector manifold,  $p_I$ , is related to the entering flow,  $\dot{w}_n$ , from the upstream pipe segment and the injector flow,  $\dot{w}_I$ , as follows:

$$\frac{dp_I}{dt} = \frac{c_I^2}{V_I g} (\dot{w}_n - \dot{w}_I) \quad (32)$$

where  $V_I$  is the injector volume and  $c_I$  is the fluid sonic velocity.

The injector flow is controlled by the differential pressure across the injector as well as by the resistance and inertia of the injector orifices. Thus,

$$p_I - p_C = R_I \dot{w}_I^2 + \frac{\ell}{Ag} \frac{d}{dt} \dot{w}_I, \quad (33)$$

where  $p_C$  is the thrust chamber pressure,  $R_I$  is the injector hydraulic resistance and  $\ell/Ag$  is the equivalent inertance of all the injector orifices combined, i.e.,

$$\frac{1}{\ell/Ag} = g \sum_{i=1}^n \frac{1}{\ell_i/A_i} \quad (34)$$

In the preceding equation,  $l_i$  and  $A_i$  are the length and area, respectively, of an individual injector orifice.

An additional factor which can have a significant effect on the response of the feed system to chamber pressure oscillations is injector face flexibility. This effect can be expressed as a change in injector volume proportional to a change in injector pressure drop:

$$\frac{d V_I}{dt} = K \left( \frac{d p_I}{dt} - \frac{d p_c}{dt} \right) \quad (35)$$

Also,

$$\frac{dp}{dt} = \frac{c^2}{g} \frac{d}{dt} \left( \frac{w}{V} \right), \quad (36)$$

which can be rewritten as

$$\frac{dp}{dt} = \frac{c^2}{Vg} \dot{w} - \frac{c^2 \rho}{Vg} \dot{v}. \quad (37)$$

Combining Eqs. (35) and (37) gives

$$\frac{d p_I}{dt} = \frac{c_I^2}{V_I g} (\dot{w}_n - \dot{w}_I) - \frac{c_I^2 \rho_I}{V_I g} \left[ K \left( \frac{d p_I}{dt} - \frac{d p_c}{dt} \right) \right], \quad (38)$$

which can be rewritten as

$$\left( 1 + \frac{c_I^2 \rho_I K}{V_I g} \right) \frac{d p_I}{dt} = \frac{c_I^2}{V_I g} (\dot{w}_n - \dot{w}_I) + \frac{c_I^2 \rho_I K}{V_I g} \frac{d p_c}{dt} \quad (39)$$

This expression reduces to Eq. (32) if no injector flexibility exists ( $K = 0$ ).

### Two-Phase Flow Acoustic Velocity

In the waterhammer equations the acoustic velocity,  $c$ , of the fluid appears in two places; (1) directly in the constant relating flow to pressure, and (2) indirectly in the time delay value,  $\tau$ , which equals  $\ell/c$  seconds, where  $\ell$  is the pipe segment length. The acoustic velocity of a fluid is a property of that fluid. However, its effective value can be reduced by the elastic walls of the flow passage or by the entrainment of gas and vapor in the liquid (two phase flow). Gas in the liquid can appear from two sources. One source is direct entrainment from mixing of gas and liquid in the propellant tank, while the other can result from the evolution of dissolved gas as the pressure drops along the feed system.

Given the steady-state pressure at each point in the feed system and data on the solubility of the pressurant gas in the propellant as a function of pressure and temperature, the amount of gas in the fluid can be determined for each feed system segment. Then, knowing the amount of gas in the liquid, the effective acoustic velocity of the mixture may be calculated.

Assuming isentropic compression of the gas, the change in volume of the gas is

$$dV_g = - \frac{V_g}{Kp} dp, \quad (40)$$

and for the liquid

$$dV_\ell = \frac{V_\ell}{\beta} dp \quad (41)$$

Defining a constant,  $\alpha \equiv \frac{V_g}{V_\ell}$

the following relation is obtained:

$$\frac{dV_t}{V_t} = \frac{- dp}{\left[ \frac{1 + \alpha}{\frac{1}{\beta} + \frac{\alpha}{Kp}} \right]} \quad (42)$$



The bracketed term is the compressibility of the mixture. The density of the mixture can be shown to be

$$\rho_m = \frac{\alpha \rho_g + \rho_l}{(1 + \alpha)} \quad (43)$$

The acoustic velocity of a liquid in an elastic pipe is

$$c = \sqrt{\frac{1}{\frac{\rho}{g} \left( \frac{1}{\beta} + \frac{Dc_f}{eE} \right)}} \quad (44)$$

Using the above expressions for density and compressibility, the acoustic velocity, can be written as

$$c = \left[ \frac{1}{\frac{\rho_m}{1+\alpha} \left( \frac{\alpha}{\rho_l c_l^2} + \frac{1}{\rho_g c_g^2} + \frac{1+\alpha}{g} \frac{Dc_f}{Ee} \right)} \right]^{\frac{1}{2}} \quad (45)$$

This expression can be used to define the acoustic velocity of a feed system segment with two phase flow. For an all liquid system,  $\alpha = 0$  and the same equation can be used.

In the Hydrodynamics subprogram the effect of the wall compressibility term,  $\frac{Dc_f}{Ee}$ , on the fluid acoustic velocity is handled automatically (assuming input value of  $\frac{Dc_f}{Ee}$  are provided for each feed system segment). However, the program does not compute the effects of two-phase flow. If such flow occurs in the feed system being modeled, an effective fluid acoustic velocity must be pre-calculated for each affected segment. Equation (45) above, with the  $\frac{Dc_f}{Ee}$  term set equal to zero can be used for this calculation.

#### Simulation of Branch Lines

In the Hydrodynamics subprogram, branched lines are handled by assuming that each branch has zero internal volume and that its flows are incompressible. Thus, the pressures at the end of all segments which meet at a branch are set equal. The continuity of flow is then used to provide the additional equations in combination with the waterhammer equations to solve for the overall feed system dynamic response.

### Generalized Feed System Model

A schematic of the generalized feed system which is modeled by the hydrodynamics subprogram is shown in Fig. 4. The configuration chosen is based on design and operating mode data for the OMS, PBK, and RCS feed systems obtained from McDonnell Douglas/St. Louis. The system is comprised of 30 individual line segments, each denoted in Fig. 4, as the lines between the black dots. A continuous parameter representation of each line segment is obtained through the use of separate sets of waterhammer equations. Each line segment can have a different line length, area, wall compliance, fluid acoustic velocity and resistance, and hence can model a wide variety of feed system components by merely choosing the appropriate values from these parameters. Also included in the generalized model are lumped parameter descriptions of two injectors (designated "Ø" and "F" on Fig. 4). Parameters for the injectors are volume, resistance, inertance, fluid acoustic velocity and face flexibility.

The system of 57 equations describing the generalized Fig. 4 feed system is listed in Table 1. The equations are shown in the linearized, LaPlace transformed format required by the frequency response subroutine.

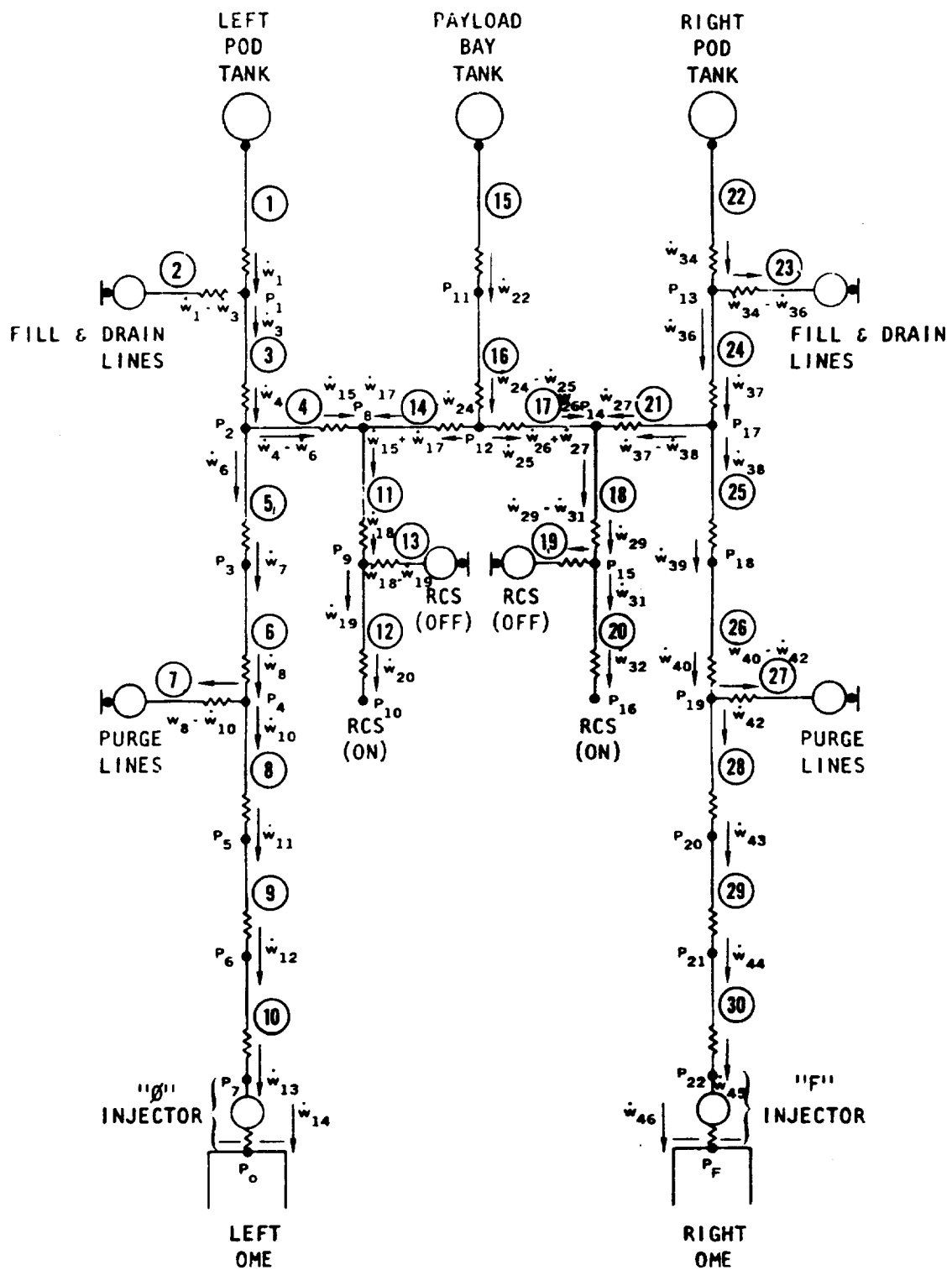


Figure 4. Generalized OME Feed System Schematic

TABLE 1. HYDRODYNAMIC EQUATIONS

$$P_1 + (R_1 + a_1) \dot{w}_1 + [P_1 + (R_1 - a_1) \dot{w}_1] e^{-2T_1 s} = 0 \quad 1-1$$

$$P_1 - a_3 \dot{w}_3 - [P_2 + (R_3 - a_3) \dot{w}_4] e^{-T_3 s} = 0 \quad 1-2$$

$$P_2 + (R_3 + a_3) \dot{w}_4 - [P_1 + a_3 \dot{w}_3] e^{-T_3 s} = 0 \quad 1-3$$

$$P_1 - (R_2 + a_2) (\dot{w}_1 - \dot{w}_3) - [P_1 - (R_2 - a_2) (\dot{w}_1 - \dot{w}_3)] e^{-2T_2 s} = 0 \quad 1-4$$

$$P_2 - a_5 \dot{w}_6 - [P_3 + (R_5 - a_5) \dot{w}_7] e^{-T_5 s} = 0 \quad 1-5$$

$$P_2 - a_4 (\dot{w}_4 - \dot{w}_6) - [P_8 + (R_4 - a_4) \dot{w}_{15}] e^{-T_4 s} = 0 \quad 1-6$$

$$P_8 + (R_4 + a_4) \dot{w}_{15} - [P_2 + a_4 (\dot{w}_4 - \dot{w}_6)] e^{-T_4 s} = 0 \quad 1-7$$

$$P_3 + (R_5 + a_5) \dot{w}_7 - [P_2 + a_5 \dot{w}_6] e^{-T_5 s} = 0 \quad 1-8$$

$$P_3 - a_6 \dot{w}_7 - [P_4 + (R_6 - a_6) \dot{w}_8] e^{-T_6 s} = 0 \quad 1-9$$

$$P_4 - (R_7 + a_7) (\dot{w}_8 - \dot{w}_{10}) - [P_4 - (R_7 - a_7) (\dot{w}_8 - \dot{w}_{10})] e^{-2T_7 s} = 0 \quad 1-10$$

$$P_4 - a_8 \dot{w}_{10} - [P_5 + (R_8 - a_8) \dot{w}_{11}] e^{-T_8 s} = 0 \quad 1-11$$

$$P_5 (R_8 + a_8) \dot{w}_{11} - [P_4 + a_8 \dot{w}_{10}] e^{-T_8 s} = 0 \quad 1-12$$

$$P_5 - a_9 \dot{w}_{11} - [P_6 + (R_9 - a_9) \dot{w}_{12}] e^{-T_9 s} = 0 \quad 1-13$$

TABLE 1. (Continued)

$$P_6 + (R_9 + a_9) \dot{w}_{12} - [P_5 + a_9 \dot{w}_{11}] e^{-T_9 s} = 0 \quad 1-14$$

$$P_6 - a_{10} \dot{w}_{12} - [P_7 + (R_{10} - a_{10}) \dot{w}_{13}] e^{-T_{10} s} = 0 \quad 1-15$$

$$P_7 + (R_{10} + a_{10}) \dot{w}_{13} - [P_6 + a_{10} \dot{w}_{12}] e^{-T_{10} s} = 0 \quad 1-16$$

$$P_7 - Z_0 s \dot{w}_{14} - R_0 \dot{w}_{14} = P_0 \quad 1-17$$

$$sP_7 - \left(\frac{c^2}{gV}\right)_0 \dot{w}_{13} + \left(\frac{c^2}{gV}\right)_0 \dot{w}_{14} + \left(\frac{c^2}{gV}\right)_0 K_0 sP_7 = \left(\frac{c^2}{gV}\right)_0 K_0 sP_0 \quad 1-18$$

$$P_8 + a_{14} \dot{w}_{17} - [P_{12} - (R_{14} - a_{14}) \dot{w}_{24}] e^{-T_{14} s} = 0 \quad 1-19$$

$$P_{12} - (R_{14} + a_{14}) \dot{w}_{24} - [P_8 - a_{14} \dot{w}_{17}] e^{-T_{14} s} = 0 \quad 1-20$$

$$P_8 - a_{11} (\dot{w}_{15} + \dot{w}_{17}) - [P_9 + (R_{11} - a_{11}) \dot{w}_{18}] e^{-T_{11} s} = 0 \quad 1-21$$

$$P_9 + (R_{11} + a_{11}) \dot{w}_{18} - [P_8 + a_{11} (\dot{w}_{15} + \dot{w}_{17})] e^{-T_{11} s} = 0 \quad 1-22$$

$$P_9 - (R_{13} + a_{13}) (\dot{w}_{18} - \dot{w}_{19}) - [P_9 - (R_{13} - a_{13}) (\dot{w}_{18} - \dot{w}_{19})] e^{-2T_{13} s} = 0 \quad 1-23$$

$$P_9 - a_{12} \dot{w}_{19} - [X + R_{12} - a_{12}) \dot{w}_{20}] e^{-T_{12} s} = 0 \quad 1-24$$

$$P_{11} + (R_{15} + a_{15}) \dot{w}_{22} + [P_{11} + (R_{15} - a_{15}) \dot{w}_{22}] e^{-2T_{15} s} = 0 \quad 1-25$$

$$P_{11} - a_{16} \dot{w}_{22} - [P_{12} + (R_{16} - a_{16}) (\dot{w}_{24} + \dot{w}_{25})] e^{-T_{16} s} = 0 \quad 1-26$$

TABLE 1. (Continued)

$$P_{12} + (R_{16} + a_{16}) (\dot{w}_{24} + \dot{w}_{25}) - [P_{11} + a_{16} \dot{w}_{22}] e^{-T_{16}^s} = 0 \quad 1-27$$

$$P_{13} + (R_{22} + a_{22}) \dot{w}_{34} + [P_{13} + (R_{22} - a_{22}) \dot{w}_{34}] e^{-2T_{22}^s} = 0 \quad 1-28$$

$$P_{13} - a_{24} \dot{w}_{36} - [P_{17} + (R_{24} - a_{24}) \dot{w}_{37}] e^{-T_{24}^s} = 0 \quad 1-29$$

$$P_{13} - (R_{23} + a_{23}) (\dot{w}_{34} - \dot{w}_{36}) - [P_{13} - (R_{23} - a_{23}) (\dot{w}_{34} - \dot{w}_{36})] e^{-2T_{23}^s} = 0 \quad 1-30$$

$$P_{12} - (R_{17} + a_{17}) \dot{w}_{25} - [P_{14} - a_{17} \dot{w}_{26}] e^{-T_{17}^s} = 0 \quad 1-31$$

$$P_{14} + a_{17} \dot{w}_{26} - [P_{12} - (R_{17} - a_{17}) \dot{w}_{25}] e^{-T_{17}^s} = 0 \quad 1-32$$

$$P_{17} - a_{21} (\dot{w}_{37} - \dot{w}_{38}) - [P_{14} + (R_{21} - a_{21}) \dot{w}_{27}] e^{-T_{21}^s} = 0 \quad 1-33$$

$$P_{14} + (R_{21} + a_{21}) \dot{w}_{27} - [P_{17} + a_{21} (\dot{w}_{37} - \dot{w}_{38})] e^{-T_{21}^s} = 0 \quad 1-34$$

$$P_{14} - a_{18} (\dot{w}_{26} + \dot{w}_{27}) - [P_{15} + (R_{18} - a_{18}) \dot{w}_{29}] e^{-T_{18}^s} = 0 \quad 1-35$$

$$P_{15} + (R_{18} + a_{18}) \dot{w}_{29} - [P_{14} + a_{18} (\dot{w}_{26} + \dot{w}_{27})] e^{-T_{18}^s} = 0 \quad 1-36$$

$$P_{15} - (R_{19} + a_{19}) (\dot{w}_{29} - \dot{w}_{31}) - [P_{15} - (R_{19} - a_{19}) (\dot{w}_{29} - \dot{w}_{31})] e^{-2T_{19}^s} = 0 \quad 1-37$$

$$P_{15} - a_{20} \dot{w}_{31} - [Y + (R_{20} - a_{20}) \dot{w}_{32}] e^{-T_{20}^s} = 0 \quad 1-38$$

$$P_{16} + (R_{20} + a_{20}) \dot{w}_{32} - [P_{15} + a_{20} \dot{w}_{31}] e^{-T_{20}^s} = 0 \quad 1-39$$

TABLE 1. (Continued)

$$P_{17} - a_{25} \dot{w}_{38} - [P_{18} + (R_{25} - a_{25}) \dot{w}_{39}] e^{-T_{25}s} = 0 \quad 1-40$$

$$P_{18} + (R_{25} + a_{25}) \dot{w}_{39} - [P_{17} + a_{25} \dot{w}_{38}] e^{-T_{25}s} = 0 \quad 1-41$$

$$P_{18} - a_{26} \dot{w}_{39} - [P_{19} + (R_{26} - a_{26}) \dot{w}_{40}] e^{-T_{26}s} = 0 \quad 1-42$$

$$P_{19} + (R_{26} + a_{26}) \dot{w}_{40} - [P_{18} + a_{26} \dot{w}_{39}] e^{-T_{26}s} = 0 \quad 1-43$$

$$P_{19} - (R_{27} + a_{27})(\dot{w}_{40} - \dot{w}_{42}) - [P_{19} - (R_{27} - a_{27})(\dot{w}_{40} - \dot{w}_{42})] e^{-2T_{27}s} = 0 \quad 1-44$$

$$P_{19} - a_{28} \dot{w}_{42} - [P_{20} + (R_{28} - a_{28}) \dot{w}_{43}] e^{-T_{28}s} = 0 \quad 1-45$$

$$P_{20} + (R_{28} + a_{28}) \dot{w}_{43} - [P_{19} + a_{28} \dot{w}_{42}] e^{-T_{28}s} = 0 \quad 1-46$$

$$P_{20} - a_{29} \dot{w}_{43} - [P_{21} + (R_{29} - a_{29}) \dot{w}_{44}] e^{-T_{29}s} = 0 \quad 1-47$$

$$P_{21} + (R_{29} + a_{29}) \dot{w}_{44} - [P_{20} + a_{29} \dot{w}_{43}] e^{-T_{29}s} = 0 \quad 1-48$$

$$P_{21} - a_{30} \dot{w}_{44} - [P_{22} + (R_{30} - a_{30}) \dot{w}_{45}] e^{-T_{30}s} = 0 \quad 1-49$$

$$P_{22} + (R_{30} + a_{30}) \dot{w}_{45} - [P_{21} + a_{30} \dot{w}_{44}] e^{-T_{30}s} = 0 \quad 1-50$$

$$P_{22} - Z_F s \dot{w}_{46} - R_F \dot{w}_{46} = P_F \quad 1-51$$

$$s P_{22} - \left(\frac{c^2}{gV}\right)_F (\dot{w}_{45} - \dot{w}_{46}) + \left(\frac{c^2}{gV}\right)_F K_F s P_{22} = \left(\frac{c^2}{gV}\right)_F K_F \overline{s P_F} \quad 1-52$$

TABLE 1. (Concluded)

$$P_4 + (R_6 + a_6) \dot{w}_8 - [P_3 + a_6 \dot{w}_7] e^{-T_6 s} = 0 \quad 1-53$$

$$P_{10} + (R_{12} + a_{12}) \dot{w}_{20} - [P_9 + a_{12} \dot{w}_{19}] e^{-T_{12} s} = 0 \quad 1-54$$

$$P_{17} + (R_{24} + a_{24}) \dot{w}_{37} - [P_{13} + a_{24} \dot{w}_{36}] e^{-T_{24} s} = 0 \quad 1-55$$

$$X = P_{10} \quad 1-56$$

$$Y = P_{16} \quad 1-57$$

NOTE:  $a_N = \frac{c_N}{A_{N9}}$



It should be noted that the hydrodynamics subprogram solves the complete system of 57 equations (describing the complete Fig. 4 feed system) each time it is called. Thus the frequency response of the entire system is calculated each time. It has been shown, however, that simpler feed systems, representing only a portion of the Fig. 4 schematic, can be modeled by merely assigning values to the parameters of the unneeded line segments which will exclude them from any effect on the system frequency response. This is accomplished automatically by the hydrodynamics subprogram via the assignment of very large resistances and very short lengths to all line segments for which no data is entered.

## COMBUSTION DYNAMICS

### Analytical Approach

In the past, the combustion response has been modeled with a simple time delay(s) (Ref. 3 through 9). This time delay represents the time required for the propellants to travel at their injected velocity from the point where they are injected to another point where they burn, and implies the burning is concentrated at a fixed plane some arbitrary distance from the injector face. The procedure outlined above is obviously an oversimplification of the burning process which is distributed in some fashion throughout the combustion chamber.

Steady-state combustion models (Ref. 10 and 11 for example) provide insight to determine the droplet burning distribution as well as additional information required to relate the distribution to a combustion response as a function of frequency. Combustion models are designed to march incrementally down the combustion chamber from a set of specified initial conditions. In so doing, the model calculates the rate at which the propellants are consumed as a function of the axial position in the combustion chamber (burning rate profile).

The analytical technique selected to describe the combustion dynamics is based on employing the mathematical expressions used in the steady-state combustion models (in particular the JANNAF DER program, Ref. 11). These mathematical expressions are expanded into time average and oscillatory components and are described in the following sections.

### Atomization Process

A very essential part of the combustion field initialization is the assignment of propellant spray droplet sizes and flowrates. Analytical descriptions of the atomization process are not available but empirical correlations that relate droplet diameter to injector geometry and flow conditions are available (Refs. 12, 13, and 14). For like-doublets, one empirical relationship is (Ref. 12).

$$D_d = 4.85 \times 10^4 v_j^{-0.75} (p_c/p_j)^{-0.52} d_j^{0.57} \quad (46)$$

where  $v_j$  is the liquid jet velocity and  $d_j$  is the liquid jet diameter at the atomization plane. (For steady-state analysis, the velocity is the injection velocity and the diameter is the orifice diameter.)

For purposes of the current analysis, the atomization process is described by:

$$D_d = K(d_j)_{x=x_{imp}}^a (v_j)_{x=x_{imp}}^b \quad (47)$$

where  $x_{imp}$  is the location of the atomization plane or the impingement point. Expanding Eq. 47 into time-averaged and oscillatory parts, yields the oscillatory droplet diameter

$$\frac{\tilde{D}_d}{D_d} = a \left( \frac{\tilde{d}_j}{d_j} \right)_{x=x_{imp}} + b \left( \frac{\tilde{v}_j}{v_j} \right)_{x=x_{imp}} \quad (48)$$

In order to evaluate the oscillatory droplet diameter, the oscillatory liquid jet diameter and velocity (and therefore the jet flowrate) are required at the atomization plane. Therefore, the dynamics of the fluid from the injector to the atomization plane is required and outlined in the following section.

#### Klystron Effect

The dynamics of the liquid propellant jet from the injector face to any location in the chamber are described by the continuity and momentum equations:

$$\frac{\partial}{\partial t} (A_j \rho_j) + \frac{\partial}{\partial x} (A_j \rho_j v_j) = 0 \quad (49)$$

$$\frac{\partial}{\partial t} (A_j \rho_j v_j) + \frac{\partial}{\partial x} (A_j \rho_j v_j^2) = -A_j \frac{\partial p}{\partial x} \quad (50)$$

Assuming

$$\rho_j = \text{constant} \quad (51)$$

$$\frac{\partial p}{\partial x} = 0 \quad (52)$$

$$\phi = \bar{\phi} + \tilde{\phi} \quad (\phi \text{ any variable}), \quad (53)$$

where

$$\bar{\phi} = f(x) \quad (\text{time average value}) \quad (54)$$

$$\tilde{\phi} = \phi' e^{-i\omega t} \quad (\text{oscillatory value}) \quad (55)$$

$$\phi' = g(x), \quad (56)$$

the preceding equations can be expanded into time average and oscillatory parts and integrated between the injector face and any location in the chamber to yield:

$$\left( \frac{\tilde{v}_j}{\bar{v}_j} \right) = e^{i\omega x / \bar{v}_j} \left( \frac{\tilde{m}_j}{\bar{m}_j} \right)_{inj} \quad (57)$$

$$\left( \frac{\tilde{A}_j}{\bar{A}_j} \right) = \frac{-i\omega x}{\bar{v}_j} e^{i\omega x / \bar{v}_j} \left( \frac{\tilde{m}_j}{\bar{m}_j} \right)_{inj} \quad (58)$$

$$\left( \frac{\tilde{m}_j}{\bar{m}_j} \right) = e^{i\omega x / \bar{v}_j} \left[ 1 - \frac{i\omega x}{\bar{v}_j} \right] \left( \frac{\tilde{m}_j}{\bar{m}_j} \right)_{inj} \quad (59)$$

where  $\omega$  is the angular frequency and the oscillatory injection rate,  $(\tilde{m}_j)_{inj}$  is determined by the feed system dynamics. Equation 59 is the oscillatory jet flowrate at  $x$  and is usually referred to as the Klystron effect (Ref. 15). The Klystron time delay,  $\tau_K$ , is therefore given by

$$\tau_{Kj} = \frac{x_{Kj}}{\bar{v}_j} \quad (60)$$

Considerable amplification of the injector face flow oscillations are possible when the Klystron effect is present and could explain the periodic burst of acoustic resonances called resurging and the steep-fronted waves seen in low and intermediate frequency instabilities.

### Droplet Vaporization

Theories of droplet combustion (Refs. 10, 16, 17) are available which may be used to evaluate the extent of coupling between droplet burning rate and local pressure and velocity fluctuations. In general, droplet burning is enhanced by increased turbulence levels or by periodic directional variations in velocity, because droplets are relatively heavy and resist following gas streamlines.

Calculation of the spray heating and vaporization is usually accomplished through specification of the corresponding individual droplet processes and summation over all the droplets that constitute the spray(s) being analyzed. The calculation of single droplet evaporation is usually based on a spherically symmetric model of simultaneous heat transfer and mass transfer across the gas side boundary, or film, separating the liquid droplet from the surrounding hot combustion gas. Forced convection and resultant nonspherical transfer processes are accounted for through empirical Nusselt number correlations for both heat and mass transfer.

For the fuel or oxidizer spray, the droplet continuity equation is

$$\frac{d}{dx} (A \rho_k v_k) = -A N_k \dot{m}_{\text{vap}_k} \quad (61)$$

and the vaporization rate is (Ref. 10)

$$\dot{m}_{\text{vap}_k} = \frac{\pi D_k k_{f_k} \text{Nu}_{H_k} Z_k}{C_{p_k} v_k} \quad (62)$$

where  $\rho_k$  is the spray density (mass of spray per unit chamber volume),  $N_k$  is the number of droplets per unit chamber volume, and

$$Z_k \equiv \frac{c_{p_{v_k}} \text{Nu}_{m_k} p \text{MW}_{v_k} D_{v_k}}{k_{f_k} \text{Nu}_{H_k} R T_{f_k}} \ln \left( \frac{p}{p-p_{v_k}} \right) \quad (63)$$

Noting that

$$\rho_k = N_k m_k = N_k \rho_{\ell_k} \frac{\pi D_k^3}{6} \quad (64)$$

the droplet number flowrate can be written as

$$\dot{N}_k = v_k A N_k = \frac{A v_k \rho_k}{m_k} \quad (65)$$

Therefore, Eq. 61 can be written as

$$\frac{d}{dx} (m_k \dot{N}_k) = - \frac{\dot{N}_k}{v_k} \dot{m}_{\text{vap}_k} \quad (66)$$

For steady-state combustion models, the preceding equation (along with Eq. 62) is numerically integrated allowing the droplet diameter,  $D_k$ , to vary along the length of the combustion and maintaining constant droplet number flowrate ( $\dot{N}_k$ ). Combs (Ref. 18) has shown that changing from a variable droplet diameter to a variable droplet number flowrate yields approximately the same results for steady-state vaporization. Therefore, in order to simplify the integration for stability analysis, the droplet diameter was held constant and the droplet number flowrate was assumed to vary.

Summing Eq. 61 over all fuel or oxidizer droplet size groups yields

$$\sum_k \frac{d}{dx} (A \rho_k v_k) = -A \sum_k \rho_k \frac{\dot{m}_{\text{vap}_k}}{m_k} = -A \sum_k \frac{\rho_k (6) Z_k k_{f_k} \text{Nu}_{H_k}}{\rho_{\ell_k} D_k^2 c_{p_{v_k}}} \quad (67)$$

which can be written as

$$\frac{d}{dx} (A \rho_s v_s) = - \frac{A \rho_s}{\tau_s} = - A \dot{m}_{vap_s} \quad , \quad (68)$$

where

$$\rho_s \equiv \sum_k \rho_k \quad (69)$$

$$v_s \equiv \frac{1}{\rho_s} \sum_k (\rho_k v_k) \quad (70)$$

$$\frac{1}{\tau_s} \equiv \frac{1}{\rho_s} \sum_k \frac{\rho_k (6) Z_k k_{f_k} Nu_{H_k}}{\rho_{l_k} D_k^2 c_{p_{v_k}}} \quad (71)$$

Letting  $Z_k, k_{f_k}, c_{p_{v_k}}$  be independent of  $k$  and assuming

$$\tau_s = f(t) \quad (72)$$

yields:

$$\tau_s = \frac{\rho_{l_s} c_{p_{v_s}} D_s^2}{(6) Z_s k_{f_s} Nu_{H_s}} \quad (73)$$

where

$$D_s^2 = (\dot{m}_s)_{inj} / \sum_k \left( \frac{\dot{m}_k}{D_k^2} \right)_{inj} \quad (74)$$

From Eq. 68

$$\frac{d(A \rho_s v_s)}{A \rho_s v_s} = - \frac{dx}{\tau_s v_s} \quad (75)$$

Integrating Eq. 75 between  $x_0$ , the start of vaporization plane, and any location  $x$  yields

$$A \rho_s v_s = (\dot{m}_s)_{x=x_0} \exp \left[ - \int_{x_0}^x \frac{dx}{\tau_s v_s} \right] \quad (76)$$

substituting Eq. 76 into Eq. 68 yields the fuel or oxidizer spray vaporization rate:

$$\dot{m}_{\text{vap}_s} = \frac{(\dot{m}_s)_{x=x_0}}{A \tau_s v_s} \exp \left[ - \int_{x_0}^x \frac{dx}{\tau_s v_s} \right] \quad (77)$$

Using perturbation techniques, the time average vaporization rate can be written as

$$\bar{\dot{m}}_{\text{vap}_s} = \frac{(\bar{\dot{m}}_s)_{x=x_0}}{A \bar{\tau}_s \bar{v}_s} \exp \left[ - \frac{(x-x_0)}{\bar{\tau}_s \bar{v}_s} \right] \quad (78)$$

and the oscillatory vaporization rate can be written as

$$\begin{aligned} \tilde{\dot{m}}_{\text{vap}_s} = \bar{\dot{m}}_{\text{vap}_s} & \left\{ \frac{(\dot{m}_s)_{x=x_{K_s}}}{(\dot{m}_s)_{x=x_0}} - \frac{\tilde{\tau}_s}{\bar{\tau}_s} - \frac{\tilde{v}_s}{\bar{v}_s} \right. \\ & \left. + \int_{x_0}^x \left( \frac{\tilde{\tau}_s}{\bar{\tau}_s} + \frac{\tilde{v}_s}{\bar{v}_s} \right) \frac{dx}{\bar{\tau}_s \bar{v}_s} \right\} \quad (79) \end{aligned}$$

Assuming

$$v_s \approx (v_s)_{x=x_{K_s}} \quad (80)$$

yields

$$\frac{\tilde{v}_s}{\bar{v}_s} = - \left( \frac{\tilde{v}_s}{\bar{v}_s} \right)_{x=x_{K_s}} \quad (81)$$



Letting  $\rho_{l_s}$ ,  $c_{p_{v_s}}$ , and  $k_{f_s}$  be constant, the oscillatory time delay can therefore be expressed as

$$\frac{\tilde{\tau}_s}{\bar{\tau}_s} = 2 \frac{\tilde{D}_s}{\bar{D}_s} - \frac{\tilde{Z}_s}{\bar{Z}_s} - \frac{\tilde{Nu}_{H_s}}{\bar{Nu}_{H_s}} + \left( \frac{\partial \tau_s}{\partial MR} \right) \frac{\tilde{MR}}{\bar{\tau}_s} \quad (82)$$

The oscillatory spray droplet diameter ( $D_s$ ) is given by Eq. 48 and the oscillatory flowrate is given by Eq. 59. The above formulation results in a linear oscillatory vaporization model similar to, but more realistic than Crocco's  $n-\tau$  model (Ref. 4). The formulation includes the effects of: (1) distributed energy release, (2) oscillations in the injection rate, (3) oscillations in droplet diameter, (4) oscillations in droplet temperature, (5) gas pressure and velocity oscillations, and (6) oscillations in the local mixture ratio.

Nusselt Number. It may be observed that one of the dominant terms in both the expressions for the average and oscillatory time delay is the Nusselt number. The Nusselt number, for longitudinal modes, is (Ref. 19).

$$Nu_{H_s} = 2.0 + 0.6 Pr_s^{1/3} \left[ \frac{\rho D_s}{\mu} |v - v_s| \right]^{1/2} \quad (83)$$

In order to evaluate the oscillatory Nusselt number, the oscillatory droplet spray velocity is required. The droplet spray velocity can be obtained from the drag equation.

$$m_s \frac{dv_s}{dt} = \frac{\pi}{8} \rho D_s^2 |v - v_s| (v - v_s) C_{D_s} \quad (84)$$

Letting

$$\tau_{\text{drag}_s} \equiv \frac{\rho_s D_s^2}{(18) \alpha_s \mu} \quad (85)$$

where

$$\alpha_s \equiv \frac{C_{D_s}}{24} \left( \frac{D_s \rho}{\mu} |v - v_s| \right) \quad , \quad (86)$$

the oscillatory droplet spray velocity can be written as

$$\tilde{v}_s = \left[ \frac{1 + i\omega \tau_{\text{drag}_s}}{1 + (\omega \tau_{\text{drag}_s})^2} \right] \tilde{v} = R_{v_s} \tilde{v} \quad (87)$$

Defining

$$F_\rho \equiv \left( \frac{\rho}{\bar{\rho}} \right)^{1/2}, \quad F_{v_s} = \left[ \frac{|v - v_s|}{c \Delta M_s} \right]^{1/2} \quad (88)$$

where

$$\Delta M_s \equiv \left[ \frac{|v - v_s|}{c} \right]_{\text{steady state}} \quad , \quad (89)$$

the Nusselt number can be written as

$$Nu_{H_s} = 2.0 + 0.6 Pr_s^{1/3} \left[ \frac{\bar{\rho} D_s}{\mu} c \Delta M_s \right]^{1/2} F_\rho F_{v_s} \left( \frac{D_s}{\bar{D}_s} \right)^{1/2} \quad (90)$$

Expanding the preceding equation into time average and oscillatory parts yields

$$\overline{Nu}_{H_s} = 2.0 + 0.6 Pr_s^{1/3} \left[ \frac{\overline{\rho D_s}}{\mu} c \Delta M_s \right]^{1/2} \overline{F}_\rho \overline{F}_{V_s} \quad (91)$$

$$\frac{\tilde{Nu}_{H_s}}{\overline{Nu}_{H_s}} = \left( \frac{\overline{Nu}_{H_s} - 2}{\overline{Nu}_{H_s}} \right) \left[ \frac{1}{2} \left( \frac{\tilde{D}_s}{\overline{D}_s} \right) + \frac{\tilde{F}_\rho}{\overline{F}_\rho} + \frac{\tilde{F}_{V_s}}{\overline{F}_{V_s}} \right] \quad (92)$$

Letting

$$\frac{\tilde{F}_{V_s}}{\overline{F}_{V_s}} = R_{F_{V_s}} \frac{\tilde{v}}{\overline{c}}, \text{ and} \quad (93)$$

$$\frac{\tilde{F}_\rho}{\overline{F}_\rho} = R_{F_\rho} \frac{\tilde{\rho}}{\overline{\rho}}, \quad (94)$$

the oscillatory Nusselt number is

$$\frac{\tilde{Nu}_{H_s}}{\overline{Nu}_{H_s}} = \frac{\overline{Nu}_{H_s} - 2}{\overline{Nu}_{H_s}} \left[ \frac{1}{2} \left( \frac{\tilde{D}_s}{\overline{D}_s} \right) + R_{F_\rho} \left( \frac{\tilde{\rho}}{\overline{\rho}} \right) + R_{F_{V_s}} \left( \frac{\tilde{v}}{\overline{c}} \right) \right] \quad (95)$$

For small perturbations in the pressure, the linear response factors and the time average values for  $F_\rho$  and  $F_v$  are

$$\bar{F}_\rho = 1, \bar{F}_{v_s} = 1 \quad (96)$$

$$R_{F_\rho} = \frac{1}{2}, R_{F_{v_s}} = \frac{1}{2} \left( \frac{1 - R_{v_s}}{\Delta M_s} \right) \quad (97)$$

Calculations have been made which indicate that, for large droplet diameters, the average and oscillatory Nusselt numbers are quite sensitive to pressure and velocity oscillations. Therefore, the Nusselt number can have a significant effect on engine stability.

Droplet Heat Transfer Blockage Term. The oscillatory combustion time delay given by Eq. 82 requires the evaluation of the heat transfer blockage term ( $Z_s$ ) which is related to the combustion gas and liquid vapor properties by Eq. 63. Because the vapor pressure ( $P_{v_s}$ ) at the droplet surface is related to the droplet temperature, the blockage term also depends on the oscillatory droplet surface temperature inside the droplet which is given by:

$$\frac{\partial}{\partial t} (\rho_l c_{v_l} T_l) = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 k_{eff_l} \frac{\partial T_l}{\partial r} \right) \quad (98)$$

Therefore, the oscillatory heat transfer rate to the droplet can be related to the oscillatory droplet surface temperature by

$$\tilde{Q}_s = R_{T_s} \tilde{T}_s \quad (99)$$

The droplet heating rate can also be written as (Ref. 10)

$$Q_s = Z_s k_{f_s} Nu_{H_s} \left[ \frac{(T - T_s)}{(e^{Z_s} - 1)} - \frac{\Delta H_{vad_s}}{c_{p_{v_s}}} \right] (\pi D_s) \quad (100)$$

Assuming that

$$\left( \frac{d\bar{T}_s}{dt} \right) = 0 \quad (\text{droplet at "wet bulb" temperature}) \quad (101)$$

and

$$\frac{\tilde{p}_v}{\bar{p}_v} = R_{p_v} \frac{\tilde{p}}{\bar{p}}, \quad (102)$$

etc. for the other variables, the response factor for the heat transfer blockage term can be related to droplet and gas properties and flow conditions.

Examination of the response factor for the heat transfer blockage term indicates that this term is not important at low frequencies (also see Ref. 20 and 21). Therefore, the oscillatory heat blockage term has not been included in the computer program. Detailed equations for this term are presented in Appendix B of Ref. 1.

### Generalized Vaporization Rate Expression

In order to maintain generality in representing the combustor dynamics, the spray vaporization rates (fuel and oxidizer) were written as:

$$\begin{aligned} \dot{m}_{\text{vap}_s} = \bar{m}_{\text{vap}_s} \left\{ c_{1s} \left( \frac{\tilde{p}}{\bar{p}} \right)_{x=0} + c_{2s} \left( \frac{\tilde{p}}{\bar{p}} \right) + c_{3s} \left( \frac{\tilde{p}}{\bar{p}} \right)_{x=0} \right. \\ + c_{4s} \left( \frac{\tilde{p}}{\bar{p}} \right) + c_{5s} (\tilde{MR})_{x=0} + c_{6s} (\tilde{MR}) \\ + c_{7s} \left( \frac{\tilde{v}}{\bar{c}} \right)_{x=0} + c_{8s} \left( \frac{\tilde{v}}{\bar{c}} \right) + \int_{x_0}^x \left[ c_{9s} \left( \frac{\tilde{p}}{\bar{p}} \right)_{x=0} \right. \\ + c_{10s} \left( \frac{\tilde{p}}{\bar{p}} \right) + c_{11s} \left( \frac{\tilde{p}}{\bar{p}} \right)_{x=0} + c_{12s} \left( \frac{\tilde{p}}{\bar{p}} \right) + c_{13s} (\tilde{MR})_{x=0} \\ \left. + c_{14s} (\tilde{MR}) + c_{15s} \left( \frac{\tilde{v}}{\bar{c}} \right)_{x=0} + c_{16s} \left( \frac{\tilde{v}}{\bar{c}} \right) \right] \frac{dx}{\bar{\tau}_s \bar{v}_s} \left. \right\} \quad (103) \end{aligned}$$

Combining the expressions of the preceding sections with this generalized vaporization rate expression yielded the combustion coefficients:

$$c_{1s} = R_{m_s} - R_{v_s} - 2 R_{D_s} + \left( \frac{\bar{Nu}_s - 2}{\bar{Nu}_s} \right) \frac{R_{D_s}}{2} \quad (104)$$

$$c_{4s} = R_{F_s} \left( \frac{\bar{Nu}_s - 2}{\bar{Nu}_s} \right) \quad (105)$$

$$c_{6s} = - \left( \frac{\partial \tau_s}{\partial MR} \right) \frac{1}{\tau_s} \quad (106)$$

$$c_{8s} = R_{FVs} \left( \frac{\overline{Nu}_s - 2}{\overline{Nu}_s} \right) \quad (107)$$

$$c_{9s} = 2 R_{Ds} - \left( \frac{\overline{Nu}_s - 2}{\overline{Nu}_s} \right) \frac{R_{Ds}}{2} + R_{Vs} \quad (108)$$

$$c_{12s} = -c_{4s} \quad (109)$$

$$c_{14s} = -c_{6s} \quad (110)$$

$$c_{16s} = -c_{8s} \quad (111)$$

$$c_{2s} = c_{3s} = c_{5s} = c_{7s} = c_{10s} = c_{11s} = c_{13s} = c_{15s} = 0 \quad (112)$$

where the subscript  $s$  denotes the fuel or oxidizer and

$$R_{ms} = G_{inj_s} \left( 1 - \frac{i\omega x_{ks}}{\bar{v}_s} \right) \quad (113)$$

$$R_{Us} = G_{inj_s} \quad (114)$$

$$R_{Ds} = \left[ b_s - a_s \frac{i\omega x_{imp_s}}{\bar{v}_s} \right] G_{inj_s} \quad (115)$$

In the above expressions  $G_{inj_s}$  is the oscillatory injection rate divided by the oscillatory pressure at the injector face and is calculated by the hydrodynamics subprogram.

The main function of the combustion dynamics subprogram is the calculation of the combustion coefficients. The general spray vaporization rate expressions are used in the chamber dynamic subprogram which is discussed in the following section.

## CHAMBER DYNAMICS

### Analytical Approach

Two methods of approach were considered for solving the chamber dynamics. The first method used a linear lump chamber coefficient. This method is valid only at low frequencies (less than 500 Hz) and results in a set of nonlinear algebraic equations to be solved.

The second method employed a first-order perturbation model to define the chamber frequency and growth coefficient along with the oscillatory pressure distribution in the chamber. This method is valid for all frequencies of interest in the present program (10 to 1000 Hz). For the oscillatory variables, solutions of the form  $\phi = \phi' e^{-i\omega t}$ , where  $\omega$  is the complex frequency, were assumed. These forms yielded a set of nonlinear differential equations which were numerically integrated between the injector face and the nozzle inlet plane. Using iteration techniques and the requisite boundary conditions at the injector and nozzle inlet plane, the chamber frequency and growth coefficient are obtained.

Consideration of the degree of complexity in solving the governing equations by each of the above methods as well as the range of validity of each approach resulted in choosing the first-order perturbation models as the best method for describing the chamber dynamics. In the following paragraphs, the derivation and solution to the first-order perturbation model stability equations are presented.

### First-Order Perturbation Model

In this section, chamber model equations are stated without showing their detail derivations. Complete derivation of the basic equations is presented in Ref.

1. Assumptions used in the derivation of the basic equations are: (1) ideal gas flow is a valid state equation; (2) dilute sprays occupy a negligible fraction of chamber volume; (3) the spray can be represented by a finite number of droplets; each droplet group contains a large number of locally identical drops; and, each size group constitutes a separate liquid phase and exchange terms between liquid phases are not included; (4) drag contributes only kinetic energy to the spray energy equation; (5) secondary "shear" breakup of

drops is not included; (6) negligible coupling between diffusion and thermal gradients; and (7) no body forces.

The following equations can be formulated for the gas phase:

#### Gas Continuity

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = \sum_n \sum_j (N_j^n \dot{m}_{\text{vap}j}^n) \quad (116)$$

#### Gas Momentum

$$\begin{aligned} \frac{\partial}{\partial t} (\rho \vec{u}) + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = & -\nabla p + \nabla \cdot \underline{\underline{\tau}} \\ & - \sum_n \sum_j \left[ N_j^n (\vec{F}_j^n - \dot{m}_{\text{vap}j}^n \vec{u}_j^n) \right] \end{aligned} \quad (117)$$

#### Equation of State

$$p = \rho RT \quad (118)$$

#### Shear Stress

$$\underline{\underline{\tau}} = -\mu_{\text{eff}} \left[ \nabla \vec{u} + (\nabla \vec{u})^t - \frac{2}{3} (\nabla \cdot \vec{u}) \underline{\underline{I}} \right] \quad (119)$$

#### Gas Energy

$$\begin{aligned} \frac{\partial}{\partial t} \left[ \rho \left( h + \frac{u^2}{2} \right) \right] + \nabla \cdot \left[ \rho \vec{u} \left( h + \frac{u^2}{2} \right) \right] \\ = -\nabla \cdot \vec{q} + \nabla \cdot (\vec{u} \underline{\underline{\tau}}) + \frac{\partial p}{\partial t} \\ + \sum_n \sum_j \left\{ N_j^n \left[ \dot{m}_{\text{vap}j}^n \left( h_j + \frac{(u_j^n)^2}{2} \right) \right. \right. \\ \left. \left. - Q_j^n - \vec{u}_j^n \cdot \vec{F}_j^n \right] \right\} \end{aligned} \quad (120)$$



Gas Mixture Ratio

$$\begin{aligned}
 & \frac{\partial}{\partial t} (\rho MR) + \nabla \cdot (\rho \vec{u} MR) \\
 & - \rho \mathcal{D}_{eff} \left[ \nabla^2 MR - \frac{2|\nabla MR|^2}{MR + 1} \right] \\
 & - (\nabla MR) \cdot \nabla (\rho \mathcal{D}_{eff}) = \\
 & (2 MR + 1) \left[ \sum_n^{ox} \sum_j N_j^n \dot{m}_{vap,j}^n \right] \\
 & - (MR)^2 \left[ \sum_n^{fu} \sum_j N_j^n \dot{m}_{vap,j}^n \right] \quad (121)
 \end{aligned}$$

Heat Transfer Rate

$$\vec{q} = -k_{eff} \nabla T - \sum_i (\rho \mathcal{D}_{eff}) h_i \nabla y_i \quad (122)$$

Drag Force

$$\vec{F}_j^n = \frac{\pi}{8} \left\{ \rho (D_j^n)^2 |\vec{u} - \vec{u}_j^n| (\vec{u} - \vec{u}_j^n) C_{D,j}^n \right\} \quad (123)$$

Assuming

- (1) Diffuser, thermal and viscous gradients are negligible,
- (2) Droplet drag forces and heat transfer to the droplets are negligible,
- (3) Droplet velocities are approximately equal to the gas velocity,

and letting

$$h = \left( \frac{c_p}{R} \right)_\phi RT + (h_{ref})_\phi + \left( \frac{\partial h}{\partial MR} \right)_\phi (MR - MR_\phi), \quad (124)$$

$$\left( \frac{c_p}{R} \right)_\phi = \frac{\gamma_\phi}{(\gamma_\phi - 1)}, \quad (125)$$

$$R = R_\phi + \left( \frac{\partial R}{\partial MR} \right)_\phi (MR - MR_\phi), \quad (126)$$

$$\dot{m}_{\text{vap}_{\text{ox}}} \equiv \sum_n \sum_j^{\text{ox}} N_j^n \dot{m}_{\text{vap}_j}^n \quad (127)$$

$$\dot{m}_{\text{vap}_{\text{fu}}} \equiv \sum_n \sum_j^{\text{fu}} N_j^n \dot{m}_{\text{vap}_j}^n \quad (128)$$

where the subscript  $\phi$  denotes that the properties are evaluated based on the overall injection mixture ratio during steady-state operation, the preceding equations can be simplified for longitudinal modes to

Gas Continuity

$$A \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (A \rho v) = A (\dot{m}_{\text{vap}_{\text{ox}}} + \dot{m}_{\text{vap}_{\text{fu}}}) \quad (129)$$

Gas Momentum

$$\rho \frac{\partial v}{\partial t} + \rho v \frac{\partial v}{\partial x} + \frac{\partial p}{\partial x} = 0 \quad (130)$$

Equation of State

$$p = \rho T \left[ R_\phi + \left( \frac{\partial R}{\partial MR} \right)_\phi (MR - MR_\phi) \right] \quad (131)$$

Gas Energy

$$A \frac{\partial p}{\partial t} + Av \frac{\partial p}{\partial x} + \gamma_\phi p \frac{\partial}{\partial x} (Av) =$$

$$(\gamma_\phi - 1) A \left\{ \dot{m}_{\text{vap}_{\text{ox}}} \left[ \Delta h_{\text{ox}} - \left( \frac{\partial h}{\partial MR} \right)_\phi (2MR + 1) \right] \right.$$

$$\left. + \dot{m}_{\text{vap}_{\text{fu}}} \left[ \Delta h_{\text{fu}} + \left( \frac{\partial h}{\partial MR} \right)_\phi (MR)^2 \right] \right\} \quad (132)$$

Gas Mixture Ratio

$$\rho \frac{\partial MR}{\partial t} + \rho v \frac{\partial MR}{\partial x} = (MR + 1) \left[ \dot{m}_{\text{vap}_{\text{ox}}} - (MR) \dot{m}_{\text{vap}_{\text{fu}}} \right] \quad (133)$$

Because of the complexity in solving nonlinear partial differential equations, perturbation techniques were used to simplify the governing dynamic equations. Assuming

$$\phi = \bar{\phi} + \tilde{\phi} \quad (\phi \text{ any variable}), \quad (134)$$

where

$$\bar{\phi} = f(x) \quad (135)$$

and

$$\tilde{\phi} = g(x, t), \quad (136)$$

each perturbation quantity was taken to be of order  $(\epsilon)$ , where  $(\epsilon)$  is a small ordering parameter that is a measure of the wave amplitude. The perturbation expressions for each of the independent variables were substituted back into the nonlinear partial differential equations, where all terms of the order  $(\epsilon^2)$  or higher were neglected. The resulting time-averaged equations were solved for the time-averaged variables and the oscillatory equations were solved by assuming solutions of the form

$$\tilde{\phi} = \phi' e^{-i\omega t} \quad (137)$$

where  $\phi' = f(x)$  and  $\omega$  is the complex frequency. The resulting equations form a system of ordinary differential equations in terms of the variables  $\phi'$  and can be numerically integrated by employing boundary conditions and iteration techniques.

Following this approach the perturbation equations were expressed as:

$$\rho \equiv \bar{\rho} \left[ 1 + \rho' e^{-i\omega t} \right] \quad (138)$$

$$v \equiv \bar{v} + c_{\phi} v' e^{-i\omega t} \quad (139)$$

$$T = \bar{T} \left[ 1 + T' e^{-i\omega t} \right] \quad (140)$$

$$p = \bar{p} \left[ 1 + p' e^{-i\omega t} \right] \quad (141)$$

$$MR = \overline{MR} + MR' e^{-i\omega t} \quad (142)$$

$$\dot{m}_{\text{vap}_{\text{ox}}} = \overline{\dot{m}_{\text{vap}_{\text{ox}}}} + \dot{m}'_{\text{vap}_{\text{ox}}} e^{-i\omega t} \quad (143)$$

$$\dot{m}_{\text{vap}_{\text{fu}}} = \overline{\dot{m}_{\text{vap}_{\text{fu}}}} + \dot{m}'_{\text{vap}_{\text{fu}}} e^{-i\omega t} \quad (144)$$

The time-averaged equations were determined to be:

(a) Gas Continuity

$$\frac{d}{dx} (A \bar{\rho} \bar{v}) = A (\overline{\dot{m}_{\text{vap}_{\text{ox}}}} + \overline{\dot{m}_{\text{vap}_{\text{fu}}}}) \quad (145)$$

(b) Gas Momentum

$$\bar{\rho} \bar{v} \frac{d\bar{v}}{dx} + \frac{d\bar{p}}{dx} = 0 \quad (146)$$

(c) Equation of State

$$\bar{p} = \bar{\rho} R_{\phi} T \left[ 1 + \frac{1}{R_{\phi}} \left( \frac{\partial R}{\partial MR} \right)_{\phi} (\overline{MR} - MR_{\phi}) \right] \quad (147)$$

(d) Gas Energy

$$A \bar{v} \frac{d\bar{p}}{dx} + \gamma_{\phi} \bar{p} \frac{d}{dx} (A \bar{v}) =$$

$$(\gamma_{\phi} - 1) A \left\{ \overline{\dot{m}_{\text{vap}_{\text{ox}}}} \left[ \Delta h_{\text{ox}} - \left( \frac{\partial h}{\partial MR} \right)_{\phi} (2 \overline{MR} + 1) \right] \right.$$

$$\left. + \overline{\dot{m}_{\text{vap}_{\text{fu}}}} \left[ \Delta h_{\text{fu}} + \left( \frac{\partial h}{\partial MR} \right)_{\phi} (\overline{MR})^2 \right] \right\} \quad (148)$$

(e) Gas Mixture Ratio

$$\bar{\rho} \bar{v} \frac{d\overline{MR}}{dx} = (\overline{MR} + 1) \left[ \overline{\dot{m}_{\text{vap}_{\text{ox}}}} - \overline{MR} \overline{\dot{m}_{\text{vap}_{\text{fu}}}} \right] \quad (149)$$

and the oscillatory equations were determined to be:

(a) Gas Continuity

$$\begin{aligned} \rho' \left( \frac{-i\omega}{c_\phi} \right) + \frac{dv'}{dx} + \frac{v'}{A\bar{\rho}} \frac{d(A\bar{\rho})}{dx} \\ + \left( \frac{\bar{v}}{c_\phi} \right) \frac{d\rho'}{dx} + \frac{\rho'}{A\bar{\rho} c_\phi} \frac{d}{dx} (A\bar{\rho} \bar{v}) = \\ \frac{(\dot{m}'_{\text{vap}_{\text{ox}}} + \dot{m}'_{\text{vap}_{\text{fu}}})}{\bar{\rho} c_\phi} \end{aligned} \quad (150)$$

(b) Gas Momentum

$$\begin{aligned} v' \left( \frac{-i\omega}{c_\phi} \right) + \frac{v'}{c_\phi} \frac{d\bar{v}}{dx} + \frac{\bar{v}}{c_\phi} \frac{dv'}{dx} - \frac{\rho'}{\bar{\rho} c_\phi^2} \frac{d\bar{p}}{dx} \\ + \frac{\rho'}{\bar{\rho} c_\phi^2} \frac{d\bar{p}}{dx} + \frac{\bar{p}}{\bar{\rho} c_\phi^2} \frac{d\rho'}{dx} = 0 \end{aligned} \quad (151)$$

(c) Equation of State

$$p' = \rho' + T' + \frac{\bar{\rho} \bar{T}}{\bar{p}} \left( \frac{\partial R}{\partial MR} \right)_\phi MR' \quad (152)$$

(d) Gas Energy

$$\begin{aligned} p' \left( \frac{-i\omega}{c_\phi} \right) + \left( \frac{\bar{v}}{c_\phi} \right) \left[ \frac{dp'}{dx} + \frac{p'}{\bar{p}} \frac{d\bar{p}}{dx} \right] \\ + \frac{v'}{\bar{p}} \frac{d\bar{p}}{dx} + \frac{\gamma_\phi}{\bar{p}} \left[ \frac{dv'}{dx} + \frac{v'}{A} \frac{dA}{dx} \right] \\ + \frac{\gamma_\phi p'}{A c_\phi} \frac{d}{dx} (A \bar{v}) = \frac{(\gamma_\phi - 1)}{\bar{p} c_\phi} \left\{ \dot{m}'_{\text{vap}_{\text{ox}}} \left[ \Delta h_{\text{ox}} \right] \right. \end{aligned}$$

$$\begin{aligned}
& - \left( \frac{\partial h}{\partial MR} \right)_{\phi} (2 \overline{MR} + 1) \left] + \dot{m}'_{\text{vap fu}} \left[ \Delta h_{\text{fu}} \right. \right. \\
& \left. \left. + \left( \frac{\partial h}{\partial MR} \right)_{\phi} (\overline{MR})^2 \right] - 2 \overline{m}'_{\text{vap ox}} \left( \frac{\partial h}{\partial MR} \right)_{\phi} MR' \right. \\
& \left. + 2 \overline{m}'_{\text{vap fu}} \overline{MR} \left( \frac{\partial h}{\partial MR} \right)_{\phi} MR' \right\} \quad (153)
\end{aligned}$$

(e) Gas Mixture Ratio

$$\begin{aligned}
MR' \left( \frac{-i\omega}{c_{\phi}} \right) + \left( \frac{\overline{v}}{c_{\phi}} \right) \frac{d MR'}{dx} + \left[ \left( \frac{\overline{v}}{c_{\phi}} \right) \rho' + v' \right] \frac{d MR}{dx} \\
= \frac{(\overline{MR} + 1)}{\overline{\rho} c_{\phi}} \left[ \dot{m}'_{\text{vap ox}} - \overline{MR} \dot{m}'_{\text{vap fu}} \right] \\
+ \frac{1}{\overline{\rho} c_{\phi}} \left[ \overline{m}'_{\text{vap ox}} - (2 \overline{MR} + 1) \overline{m}'_{\text{vap fu}} \right] (MR') \quad (154)
\end{aligned}$$

In the computer model, the preceding set of ordinary differential equations are numerically integrated between the injector face and the nozzle inlet plane. The method of calculating the complex frequency for the perturbation model, based on nozzle admittances calculated from upstream and downstream variables, is discussed in the Engineering Model section.

### Steady-State Solution

The boundary conditions for the steady-state differential equations are

$$x = x_0 \quad (155)$$

$$\overline{p}_{x_0} = p_c \quad (156)$$

$$\overline{v}_{x_0} = \overline{v}_{x=0} \quad (157)$$

$$\overline{MR}_{x_0} = MR_{x=0} \quad (\text{if } \overline{v}_{x=0} \neq 0) \quad (158)$$

$$(\overline{A\overline{v}})_{x_0} = (\overline{m})_{x=0} \quad (\text{if } \overline{v}_{x=0} \neq 0) \quad (159)$$

Assuming small Mach numbers, i.e.,  $M^2 \ll 1$ , the steady-state differential equations can be integrated between the start plane for vaporization ( $x_0$ ) and any location ( $x$ ) to yield

$$\bar{p} = \text{constant} = p_c \quad (160)$$

$$\bar{MR} = \frac{\left(\frac{\bar{MR}}{1+\bar{MR}}\right)_{x=0} (A\bar{\rho}\bar{v})_{x=0} + (\bar{m}_{ox})_{inj} (1 - \phi_{ox})}{\left(\frac{1}{1+\bar{MR}}\right)_{x=0} (A\bar{\rho}\bar{v})_{x=0} + (\bar{m}_{fu})_{inj} (1 - \phi_{fu})} \quad (161)$$

$$\begin{aligned} \bar{v} = & \frac{(A\bar{v})_{x=0}}{A} + \frac{(\gamma_\phi - 1)}{\gamma_\phi \bar{p} A} \left\{ (\bar{m}_{ox})_{inj} (1 - \phi_{ox}) \Delta h_{ox} \right. \\ & + (\bar{m}_{fu})_{inj} (1 - \phi_{fu}) \Delta h_{fu} + (A\bar{\rho}\bar{v}\bar{MR})_{x=0} \left(\frac{\partial h}{\partial \bar{MR}}\right)_\phi \\ & \left. - \bar{MR} \left(\frac{\partial h}{\partial \bar{MR}}\right) \left[ (A\bar{\rho}\bar{v})_{x=0} + (\bar{m}_{ox})_{inj} (1 - \phi_{ox}) \right. \right. \\ & \left. \left. + (\bar{m}_{fu})_{inj} (1 - \phi_{fu}) \right] \right\} \quad (162) \end{aligned}$$

$$\bar{p} = \frac{1}{A\bar{v}} \left\{ (A\bar{\rho}\bar{v})_{x=0} + (\bar{m}_{ox})_{inj} (1 - \phi_{ox}) \right. \\ \left. + (\bar{m}_{fu})_{inj} (1 - \phi_{fu}) \right\} \quad (163)$$

$$\bar{T} = \frac{\bar{p}}{\bar{p} R_\phi \left[ 1 + \frac{1}{R_\phi} \left(\frac{\partial R}{\partial \bar{MR}}\right)_\phi (\bar{MR} - \bar{MR}_\phi) \right]} \quad (164)$$

where

$$\phi_s = e^{-(x-x_0)/\tau_s \bar{v}_s} \quad (165)$$

If the gaseous injection velocity is equal to zero ( $\bar{v}_{x=0} = 0$ ), the steady-state mixture ratio and density at  $x = x_0$  are determined by

$$\overline{MR}_{x_0} = \left( \frac{\bar{m}_{ox}}{\bar{m}_{fu}} \right)_{inj} \left( \frac{\bar{\tau}_{fu} \bar{v}_{fu}}{\bar{\tau}_{ox} \bar{v}_{ox}} \right) \quad (166)$$

$$\begin{aligned} (\bar{\rho})_{x_0} \frac{(\gamma_\phi - 1)}{\gamma_\phi \bar{p}} \left\{ \overline{MR}_{x_0} \left[ \Delta h_{ox} - \overline{MR}_{x_0} \left( \frac{\partial h}{\partial MR} \right)_\phi \right] \right. \\ \left. + \Delta h_{fu} - \overline{MR}_{x_0} \left( \frac{\partial h}{\partial MR} \right)_\phi \right\} = \overline{MR}_{x_0} + 1 \end{aligned} \quad (167)$$

These equations were developed by taking the limit as  $x \rightarrow x_0$  from a downstream distance.

### Oscillatory Solution

The boundary conditions for the oscillatory differential equations are

$$\textcircled{a} \quad x = 0 \quad (168)$$

$$p' = \Delta p \quad (169)$$

$$v' = (v')_{x=0} \quad (170)$$

From these boundary conditions and the oscillatory differential equations the oscillatory conditions at the start plane for vaporization ( $x_0$ ) can be determined and are:

$$p'_{x_0} = \left[ \Delta p \cos \left( n \frac{\omega x_0}{c_\phi} \right) + i \gamma_\phi n (v')_{x=0} \sin \left( n \frac{\omega x_0}{c_\phi} \right) \right] e^{-\frac{i\omega}{c_\phi} n^2 \frac{\bar{v}_{x=0}}{c_\phi} x_0} \quad (171)$$



$$v'_{x_0} = \left[ \frac{i}{\gamma \phi} \frac{\Delta p}{\eta} \sin \left( n \frac{\omega x_0}{c \phi} \right) + (v')_{x=0} \cos \left( n \frac{\omega x_0}{c \phi} \right) \right] e^{-\frac{i\omega}{c \phi} n^2 \frac{\bar{v}_{x=0}}{c \phi} x_0} \quad (172)$$

$$p'_{x_0} = \frac{P'_{x_0}}{\gamma \phi} \quad (173)$$

$$MR'_{x_0} = (MR')_{x=0} e^{\left( \frac{i\omega}{c \phi} \right) \left( \frac{c \phi}{\bar{v}_{x=0}} \right) x_0} \quad (174)$$

$$T'_{x_0} = p'_{x_0} - \rho'_{x_0} - \frac{1}{\bar{R}_{x_0}} \left( \frac{\partial R}{\partial MR} \right)_{\phi} MR'_{x_0} \quad (175)$$

where

$$n \equiv \sqrt{\frac{\bar{\rho}_{x_0} c \phi^2}{\gamma \phi \bar{p}}} \quad (176)$$

If the gaseous injection velocity is equal to zero ( $\bar{v}_{x=0} = 0$ ), the oscillatory mixture ratio at  $x_0$  is determined by

$$\begin{aligned} MR'_{x_0} &= \left[ \frac{(\bar{m}_{fu})_{inj}}{A \bar{\rho}_{x_0} c \phi} (1 + \overline{MR}_{x_0}) - \frac{i\omega}{c \phi} (\bar{\tau}_{fu} \bar{v}_{fu}) \right] \\ &= \frac{(\overline{MR}_{x_0} + 1)}{A \bar{\rho}_{x_0} c \phi} (\bar{m}_{fu})_{inj} \overline{MR}_{x_0} \left( \frac{\dot{m}'_{vap_{ox}}}{\bar{m}_{vap_{ox}}} \frac{\dot{m}'_{vap_{fu}}}{\bar{m}_{vap_{fu}}} \right) \\ &= (v'_{x_0}) \overline{MR}_{x_0} \frac{(\overline{MR}_{x_0} + 1)}{(\overline{MR}_{x_0} + 2)} \left( 1 - \frac{\overline{MR}}{\overline{MR}_{inj}} \right) \quad (177) \end{aligned}$$

This equation was developed by taking the limit of the mixture ratio equation as  $x \rightarrow x_0$  from a downstream distance.

The ordinary differential equations describing the oscillatory solution are solved using a second order implicit finite difference method. This method has the advantage of being simple to implement and modify, as well as being unconditionally stable for systems of equations which do not have exponentially growing solutions. The method as applied to the first order system

$$Y' = AY + g \quad (178)$$

where  $Y$  and  $g$  are  $n \times 1$  vectors and  $A$  is an  $n \times n$  matrix is as follows:

$$y_{i+1} = y_i + \frac{\Delta x}{2} A_{i+\frac{1}{2}} (y_i + y_{i+1}) + g_{i+\frac{1}{2}} \quad (179)$$

Here, the subscript  $i$  refers to the  $i$ 'th mesh point in the finite difference scheme, e.g.,  $x_i = x_0 + i\Delta x$ . The  $y_i$  approximate the  $Y$  vector at  $x_i$ . That is  $y_i \sim Y_i = Y(x_i)$ . The subscript  $i+\frac{1}{2}$  refers to evaluation at  $x_i + \Delta x/2$ ; e.g.,  $A_{i+\frac{1}{2}} = A(x_i + \frac{\Delta x}{2})$ .

That the above method leads to a second order approximation (error is proportional to  $\Delta x^3$ ) can be shown as follows:

Solving for  $y_{i+1}$  yields

$$y_{i+1} = (I - \frac{\Delta x}{2} A_{i+\frac{1}{2}})^{-1} (I + \frac{\Delta x}{2} A_{i+\frac{1}{2}}) y_i + \Delta x (I - \frac{\Delta x}{2} A_{i+\frac{1}{2}})^{-1} g_{i+\frac{1}{2}} \quad (180)$$

Without loss of generality, assume  $i = 0$ .

From the two expansions

$$Y_1 = Y_{\frac{1}{2}} + \frac{\Delta x}{2} Y'_{\frac{1}{2}} + \frac{\Delta x^2}{8} Y''_{\frac{1}{2}} + o(\Delta x^3) \quad (181)$$

$$Y_0 = Y_{\frac{1}{2}} - \frac{\Delta x}{2} Y'_{\frac{1}{2}} + \frac{\Delta x^2}{8} Y''_{\frac{1}{2}} + o(\Delta x^3) \quad (182)$$

the following are obtained

$$Y_1 = Y_0 + \Delta x Y'_{\frac{1}{2}} + o(\Delta x^3) \quad (183)$$

and

$$Y_{\frac{1}{2}} = (Y_0 + Y_1)/2 + o(\Delta x^2) \quad (184)$$

Let  $y_0 = Y_0$ ; it is necessary to show  $y_1 = Y_1 + o(\Delta x^3)$  in order to demonstrate second-order accuracy.

From the differential equation

$$g_{1/2} = Y'_{1/2} - A_{1/2} Y_{1/2} \quad (185)$$

Substituting (185) into (180) and noting  $y_0 = Y_0$ ,

$$\begin{aligned} y_1 &= \left(I - \frac{\Delta x}{2} A_{1/2}\right)^{-1} \left(I + \frac{\Delta x}{2} A_{1/2}\right) Y_0 + \Delta x \left(I - \frac{\Delta x}{2} A_{1/2}\right)^{-1} (Y'_{1/2} - A_{1/2} Y_{1/2}) \\ &= \left(I - \frac{\Delta x}{2} A_{1/2}\right)^{-1} \left\{ Y_0 + \Delta x Y'_{1/2} + A_{1/2} \Delta x \left(\frac{1}{2} Y_0 - Y_{1/2}\right) \right\} \end{aligned} \quad (186)$$

Using (183) and (184), gives the result

$$\begin{aligned} y_1 &= Y_1 + \left(I - \frac{\Delta x}{2} A_{1/2}\right)^{-1} o(\Delta x^3) \\ &= Y_1 + o(\Delta x^3) \end{aligned} \quad (187)$$

Consider now the stability of the finite difference formula (179) for systems which do not have exponentially increasing solutions; that is, the real part of each of the eigenvalues of  $A$  is negative. To prove that they are stable for this situation, define the error  $\epsilon_i = Y_i - y_i$  and consider the two equations given by (179) and

$$\begin{aligned} Y_{i+1} &= \left(I - \frac{\Delta x}{2} A_{i+1/2}\right)^{-1} \left(I + \frac{\Delta x}{2} A_{i+1/2}\right) Y_i + \\ &\quad \Delta x \left(I - \frac{\Delta x}{2} A_{i+1/2}\right)^{-1} g_{i+1/2} + o(\Delta x^3), \end{aligned} \quad (188)$$

the latter resulting from (187). Subtracting (179) from (188)

$$\epsilon_{i+1} = \left(I - \frac{\Delta x}{2} A_{i+1/2}\right)^{-1} \left(I + \frac{\Delta x}{2} A_{i+1/2}\right) \epsilon_i + o(\Delta x^3) \quad (189)$$

Let  $B = \frac{\Delta X}{2} A_{i+1/2}$ . The method is stable if and only if the matrix  $(I-B)^{-1}(I+B)$  has a spectral radius less than one, for this would produce (Ref. 22)

$$\lim_{n \rightarrow \infty} \left[ (I-B)^{-1}(I+B) \right]^n = 0 \quad (190)$$

Since the eigenvalues of  $(I-B)^{-1}(I+B)$  are just equal to  $(1+\beta)/(1-\beta)$ , where  $\beta$  is an eigenvalue of  $B$ , the spectral radius of  $(I-B)^{-1}(I+B)$  is just

$$\max_{\beta} \left| \frac{1+\beta}{1-\beta} \right| \quad (191)$$

For this to be less than one,

$$|1+\beta| < |1-\beta| \quad (192)$$

for all  $\beta$ . This implies

$$1+\beta + \bar{\beta} + \beta\bar{\beta} < 1-\beta-\bar{\beta} + \beta\bar{\beta} \quad (193)$$

or

$$\beta + \bar{\beta} < -(\beta + \bar{\beta}) \quad (194)$$

$$\text{Real}(\beta) < 0 \quad (195)$$

Since  $\beta = \frac{\Delta X}{2\alpha}$ , where  $\alpha$  is an eigenvalue of  $A$ , the method will be stable if all the eigenvalues of  $A$  have real parts less than zero, that is, the solutions to (178) are not exponentially increasing.

### Nozzle Admittance

The nozzle admittance based on downstream conditions is calculated based on the following analysis.

The gas flowrate of the nozzle inlet plane is

$$\dot{m} = \frac{\rho A_t g}{c^*} = A \rho v \quad (196)$$

where the characteristic velocity is

$$c^* = \frac{\sqrt{g \gamma R T}}{\gamma \left[ \frac{2}{\gamma+1} \right]^{(\gamma+1)/2(\gamma-1)}} \quad (197)$$

For short nozzles, the oscillatory mass flowrate can be written as

$$\frac{\dot{m}'}{\dot{m}} = \frac{\rho'}{\rho} + \frac{v'}{v} = \frac{\rho'}{\rho} - \frac{1}{2} \frac{T'}{T} - \left( \frac{\partial \bar{c}^*}{\partial MR} \right) \frac{MR'}{c^*} \quad (198)$$

Assuming

$$\frac{\rho'}{\rho} = \frac{1}{\gamma} \frac{p'}{p}, \quad \frac{T'}{T} = \left( \frac{\gamma-1}{\gamma} \right) \frac{p'}{p}, \quad (199)$$

the nozzle admittance for a short nozzle can be written as

$$A_{N_s} = \bar{\rho} \bar{c} \bar{v} \left[ \frac{(\gamma-1)}{2\gamma \bar{p}} - \left( \frac{\partial \bar{c}^*}{\partial MR} \right) \frac{MR'}{c^* \bar{p}'} \right] \quad (200)$$

Assuming

$$A_{ND} = A_{Ns} \left( \frac{A_{ND}}{A_{Ns}} \right) \quad MR = \text{constant} \quad (201)$$

the nozzle admittance based on downstream conditions becomes

$$A_{ND} = \left[ 1 - \left( \frac{\partial \bar{c}^*}{\partial MR} \right) \frac{MR' \bar{p}}{\bar{c}^* p'} \left( \frac{2\gamma}{\gamma-1} \right) \right]_{x=\ell} A_{NMR} = \text{constant} \quad (202)$$

where  $A_{NMR} = \text{constant}$  is calculated using the admittance program developed by Bell (Ref. 23).

The nozzle admittance based on upstream conditions is

$$A_{NU} = \gamma \phi \left( \frac{v'}{p'} \right)_{x=\ell} \quad (203)$$

where  $(v'/p')_{x=\ell}$  is calculated from the oscillatory solution. For solutions to the chamber dynamic equations, the nozzle admittance based on upstream and downstream conditions must be equal. The method of calculating the complex frequency which satisfies this condition is discussed in the Engineering Model section which follows.

## ENGINEERING MODEL

### Analytical Approach

The overall model structure had the greatest variety of factors influencing its nature. Some of these factors were related to the overall confidence in the success of the effort. Factors relating to cost included the solution time and numerical stability, which bears on the number of runs which will be required for a solution. Still other factors were related to the JSC Univac 1110 capabilities. The remaining factors concerning the overall model structure reflect on its conversion cost applicability to the resolution of propulsion system problems. Its accuracy has direct bearing on the design margins involved. The type of impact and the obtainability of characterization parameters could not limit the accuracy and useability. The type and useability of the output was also given due consideration as well as the degree of generalization such that the model can be applied to a range of systems.

The structure of the Engineering Program was based on a trade-off of setup time, storage capabilities, and solution time. General input data to the program includes geometric factors, engine operating conditions and propellant properties. An equilibrium gas properties program similar to NASA ODE computer program, and the DER combustion model program are executed external to the stability program. The control program then executes the nozzle admittance and hydrodynamics programs to calculate the admittance and oscillatory injector flowrate as a function of frequency and stores the results on tapes. Steady-state distributed combustion parameters calculated from the DER Model are inputs to the combustion dynamics subprogram which are iterated with the chamber dynamic subprogram until the nozzle and injection admittance conditions are satisfied. The solution method for obtaining solutions for the complex frequency is outlined in the following section.

### Determination of Complex Frequency

The complex frequency,  $\omega$ , is determined such that the boundary condition in the nozzle is satisfied. Specifically, the admittance is required to be continuous across the interface between the combustion zone and the zone immediately downstream of the combustion zone. In the downstream zone, the nozzle admittance,  $A_{N_D}$ , is computed from a nozzle admittance program.

In the upstream combustion zone, the nozzle admittance,  $A_{N_U}$ , is computed from the oscillatory flow parameters determined by the chamber dynamics.

The complex frequency must be such that

$$A_{N_U} = A_{N_D} \quad (204)$$

Let  $\omega = x + iy$  and  $F = A_{N_U} - A_{N_D} = u + iv$ . The numerical problem is to find  $x$  and  $y$  such that

$$u(x,y) = 0 \quad (205)$$

$$v(x,y) = 0 \quad (206)$$

Several methods were considered for solving this system of equations. Because  $F$  is not an analytic function of  $\omega$ , the complex form of the Newton-Raphson method may not always work. On the other hand, one could use the two-dimensional form of Newton-Raphson (Ref. 24), but since the derivatives of  $u$  and  $v$  with respect to  $x$  and  $y$  must be computed numerically, the two-dimensional Newton-Raphson method will require three functional evaluations of  $F$  at each  $\omega$ , i.e.,  $(x,y)$ ,  $(x + \Delta x,y)$ , and  $(x,y + \Delta y)$ . Alternatively, a far more efficient method is to use the two-dimensional form of the secant method (Ref. 24) since this does not require the evaluation of any derivatives. Specifically, this method approximates the  $u$  and  $v$  surfaces with linear functions  $u_L$  and  $v_L$  (planes) based on three previous guesses for  $\omega$ ,  $(x_1,y_1)$ ,  $(x_2,y_2)$ ,  $(x_3,y_3)$ . The next guess for  $\omega$ ,  $(x_4,y_4)$ , is determined from the equations  $u_L(x_4,y_4) = v_L(x_4,y_4) = 0$ . The new value of  $\omega$  then replaces one of the previous three values, normally the one with the largest error as measured by the absolute value of  $F(x_j,y_j)$ , and the iteration



is continued until convergence is reached. The actual equations for the above process take the following form. Let  $w_j = x_j + iy_j$ ,  $u_j = u(x_j, y_j)$ , and  $v_j = v(x_j, y_j)$  for  $j = 1, 2, 3$ .

1. Determine  $\pi_j$ ,  $j = 1, 2, 3$ , such that

$$\pi_1 + \pi_2 + \pi_3 = 1 \quad (207)$$

$$\pi_1 u_3 + \pi_2 u_2 + \pi_3 u_1 = 0 \quad (208)$$

$$\pi_1 v_3 + \pi_2 v_2 + \pi_3 v_1 = 0 \quad (209)$$

2. Compute  $\omega_4 = \pi_1 \omega_3 + \pi_2 \omega_2 + \pi_3 \omega_1$ . (210)

3. Compute  $u_4$  and  $v_4$  based on  $x_4$  and  $y_4$ .

4. Test for convergence, i.e., require that  $\min |\omega_j - \omega_4| / |\omega_4| < \epsilon_1$  and  $|F_4| < \epsilon_2$ .

If the process has not converged, continue with steps 5, 6, and 7.

5. Determine the  $j$  between 1 and 3 such that  $u_j^2 + v_j^2$  is maximum.

6. Replace  $\omega_j$ ,  $u_j$ , and  $v_j$  by  $\omega_4$ ,  $u_4$ , and  $v_4$ , respectively.

7. Go to 1.

Operationally, steps 4 and 5 may be altered to replace the  $\omega$ 's cyclically, i.e.,  $\omega_i \rightarrow \omega_{i-1}$ ,  $u_i \rightarrow u_{i-1}$ ,  $v_i \rightarrow v_{i-1}$ . In fact the computer program as written alternates between these two procedures every three iterations in order to avoid any possible cycling that may occur.

The above algorithm has been found to be very efficient when the first three guesses are relatively near an actual solution. The difficult problem was to develop a searching algorithm which determines the regions in the  $\omega$  plane where solutions exist.

One possible procedure would be to utilize the fact that the surface  $u^2 + v^2$  has an absolute minimum at each solution. Using any reasonable value of  $\omega$  as a first guess, one might be tempted to employ a gradient, or modified gradient, method to march along the surface until one comes near a relative minimum. Unfortunately, this procedure fails because the surface  $u^2 + v^2$  has many relative minima which are not actual solutions. The reason for the large number of relative minima (and maxima) for this surface is undoubtedly due to the coupling between the combustion processes and the feed system oscillation in conjunction with the very rapid change of the feed system response as a function of frequency. The searching algorithm must be able to discriminate between those relative minima that are not solutions and those that are. Such a procedure was developed for this program. It takes advantage of the fact that a large portion of the computations required are only a function of the real part of  $\omega$ , i.e., they use  $x$  as an independent variable and do not depend upon  $y$ . Thus,  $y$  may be changed without having to redo many of the calculations within the program.

Intuitively, the idea is to increment  $x$  through a range of values, while determining  $y$  at each  $x$  according to the criterion mentioned below, until it is determined that a solution has been crossed. This determination employs the use of a test function which changes sign when a root is crossed in the same manner that a single equation in one unknown changes sign as it goes through a zero. The task of developing a defining criterion for  $y$  and a test function for  $x$  would be easy if, for example,  $v$  were a strong function of  $y$  and  $u$  were a strong function of  $x$ . Then, each  $x, y$  could be chosen such that  $v(x,y) = 0$  and, as  $x$  is incremented, a solution would be crossed when  $u[x,y(x)]$  changes sign. Unfortunately, neither  $u$  nor  $v$  behaves this way.

To develop functions that do behave this way, the following procedure was developed. First, for each  $x$ , choose  $y$  such that the absolute value of  $F$  is minimized. This can be done in several ways. The program uses a method that always guarantees finding a value if one exists. Essentially, the absolute

value of  $F$  squared and its gradients are computed. The value of  $y$  is altered in the direction indicated by the gradient until either the gradient changes sign or is so close to zero that convergence has been reached. Once the gradient changes sign, Muller's method (Ref. 25) is used to converge on the root. This is essentially a bisection method followed by inverse parabolic interpolation. For this searching process, it is not necessary to make the convergence criteria very tight, since only rough estimates are eventually needed in order to start the two-dimensional secant method described earlier.

Now that a criterion for  $y$  has been established, it is only necessary to find a test function that will change sign when a solution is crossed while incrementing  $x$ . Such a function is given by

$$uu_x + vv_x \tag{211}$$

This function acts as a very good test function because it represents the coordinate direction in the  $u,v$  plane along which the vector  $(u,v)$  changes most with  $x$ . When this coordinate changes sign as one goes from, say,  $x_1$  to  $x_2$  with  $y_1$  and  $y_2$  chosen so that the length of the vector  $(u,v)$  is minimized, then it is very likely that a solution has been crossed. Exceptions to this rule occur when one is near relative minima of the surface  $u^2 + v^2$  that are not zero. To see this, consider the actual equations that are being solved. In order that the vector  $(u,v)$  is minimum for each  $y$ , it is necessary that  $\partial(u^2 + v^2)/\partial y = 0$ . That is  $uu_y + vv_y = 0$ . Combining this with the above equation, we see we are finding an  $x$  and  $y$  such that the matrix equation

$$\begin{pmatrix} u_x & v_x \\ u_y & v_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{212}$$

is satisfied. The matrix is just the transpose of Jacobian of  $u$  and  $v$  with respect to  $x$  and  $y$ .

This equation can be satisfied if either  $u$  and  $v$  are zero, or the Jacobian is singular. The Jacobian is necessarily singular at all relative minima of the surface  $u^2 + v^2$  except those at  $u = v = 0$ . In order to differentiate between those solutions to (212) that are due to singularities of the Jacobian and those that are due to  $u$  and  $v$  vanishing, we employ two different tests. First of all, when a singularity point is crossed, the determinant of the Jacobian should change sign. If this occurs, then the program rejects this point as a possible solution. Sometimes, however, the determinant does not change sign because either the convergence criterion used in the searching algorithm is too loose or because the singularity has a double root. In either case, the procedure is to test the condition number\* of the transpose of the Jacobian matrix in the region near the suspected solution. If the condition number does not exceed a given input limit (e.g., around 80), then the point in question is usually a solution.

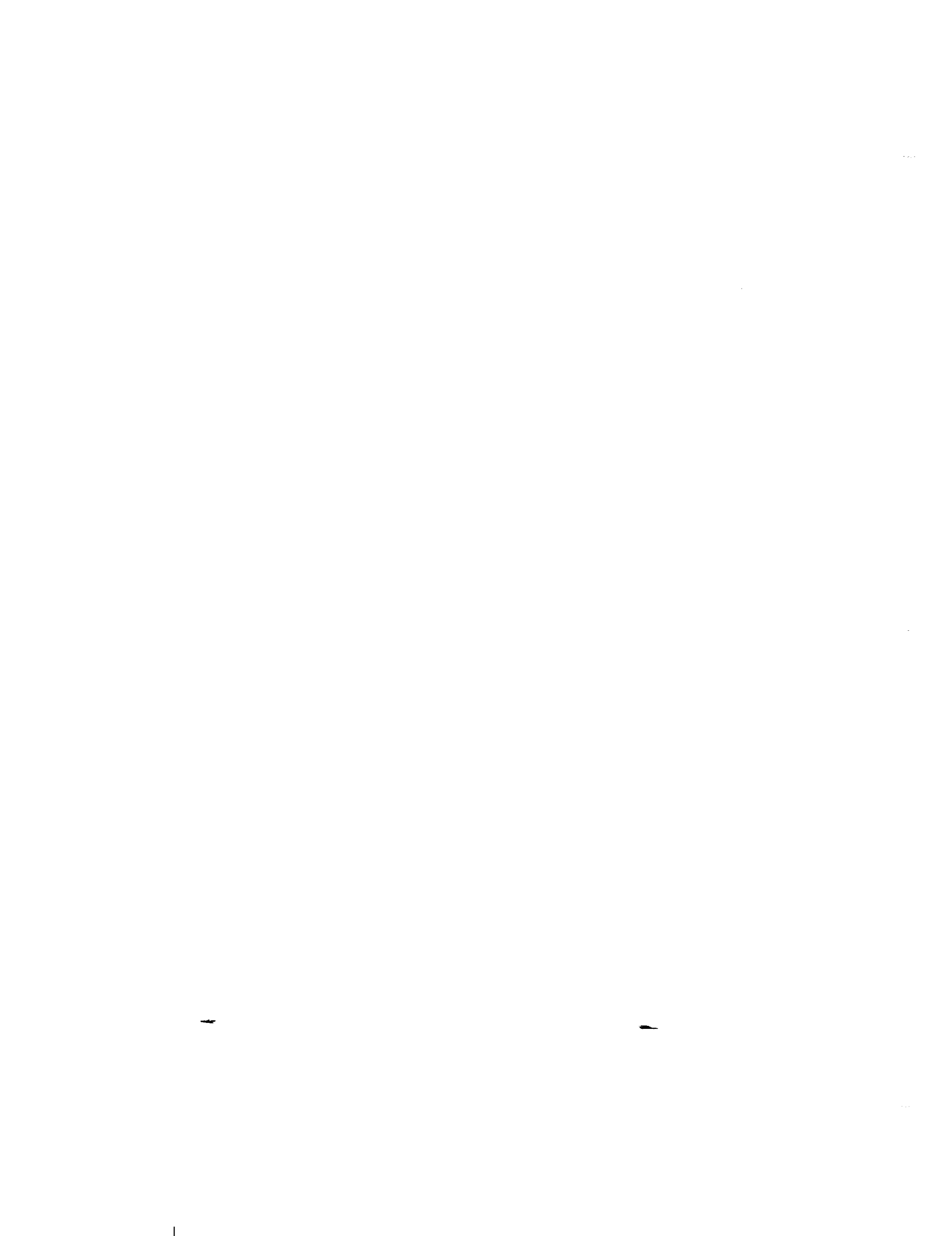
Once it is determined that a potential solution has been crossed between  $x_1$  and  $x_2$ , for example, the procedure is to (a) determine  $x_3$  based on the method of false position using the test function given in (211), (b) determine  $y_3$  to minimize  $|F|$ , and (c) use  $(x_1, y_1)$ ,  $(x_2, y_2)$ , and  $(x_3, y_3)$  as the required first three guesses for the two-dimensional secant method.

The above procedure has been found to be most satisfactory for the conditions tested in this program. The search algorithm described above has several salient features. First, as mentioned earlier, the search method takes advantage of the fact that many of the computations are not a function of the imaginary part of  $\omega$ , namely  $y$ : This allows the minimization of  $|F|$  with respect to  $y$  to proceed with high efficiency. Secondly, and more importantly,

---

\*The condition number of a matrix,  $A$ , is a measure of how sensitive a solution to the system  $Ab=c$  is to perturbations in  $c$ . It is equal to the square root ratio of the absolute value of the largest eigenvalue of  $A'A$  to the smallest eigenvalue of  $A'A$ . For singular matrices, the condition number is infinite. For matrices that are nearly singular, the condition number will be quite large.

the procedure has been automated to the extent that the user only has to specify a frequency range and a maximum number of roots desired in that range. The algorithm will start at the lower end of the frequency range and will increment through it until either the maximum number of roots are found or the upper end of the frequency range is reached. This is a very powerful property since it does not require the user to have a clear knowledge of the location of any of the roots in the  $\omega$  plane.



## PROGRAM USER'S GUIDE

### FSCSM MAIN PROGRAM

The main program for the FSCSM computer model performs much of the input/output activity of the model and controls the sequencing through the various major subprogram blocks of the model. After writing the header page (Fig. 5), the main program reads in and writes out the data described in Table 2 of the input section of this report under the heading Main Control Section Input. The program then computes the area profiles through a call to subroutine AREA. It then begins its main do loop controlling the number of solutions to the nozzle admittance boundary equation that are desired. During the iterations for each solution to be found, the main program proceeds with successive calls to subroutines NØZADM, HYDRDY, CØMBDY, and SØLVW in order to compute the downstream nozzle admittance factor, the feed system response parameters, the combustion dynamics coefficients, and performance calculations necessary to solve the nozzle admittance boundary equations respectively. Subroutine SØLVW also causes a call to subroutine CHAMDY which computes the oscillatory profiles. Further, during the first iteration in the do loop for the main program for the first solution, the main program also calls subroutine STEADY in order to obtain the steady-state profiles.

The variable ISCNT is the FORTRAN variable set by the main program and altered in SØLVW which controls the type of iteration being performed. When ISCNT equals one or four, the search algorithm described on pages 54-56 is called out. This is the initial condition at the beginning of each set of iterations to solve the nozzle admittance boundary equation. When ISCNT equals five, the two-dimensional secant method is being performed in SØLVW.

**PRECEDING PAGE BLANK NOT FILMED**

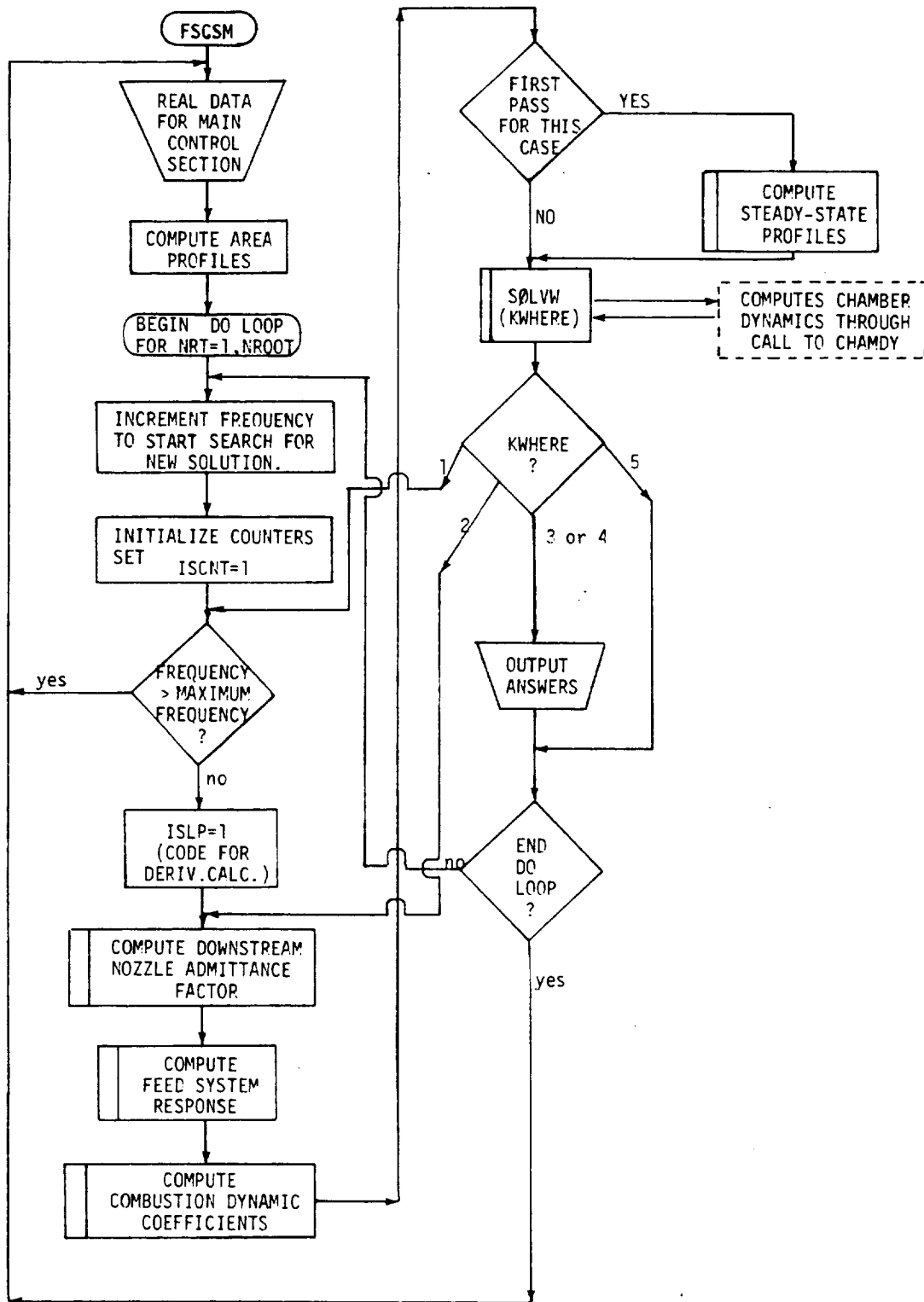


Figure 5. Logic Diagram for FSCSM Main Program

R-9808/60

ORIGINAL PAGE IS  
OF POOR QUALITY



The variable KWHERE, which is set by subroutine SØLVW, controls the logical flow in the main program subsequent to a call to SØLVW. When ISCNT equals one or four and KWHERE equals one, the control is returned to that portion of the main program which starts the calculation for the next value of  $\omega$  being tried by the search algorithm. When KWHERE equals two and ISCNT equals one or four, control is passed to that portion which will perform calculations for a perturbed  $\omega$  in order to compute the derivative of  $A_{N_U} - A_{N_D}$  with respect to  $\omega_R$ . When ISCNT equals 5 and convergence has not been reached, the normal exit from SØLVW also sets KWHERE equal to two. For this case however, no derivatives are calculated. The program will just proceed with successive calls to NØZADM, HYDRDY, and CØMBDY in order to compute the downstream nozzle admittance factor, the feed system response, and the combustion dynamic coefficients, respectively. It then proceeds to SØLVW in order to obtain an updated estimate of  $\omega$  using the two-dimensional secant method.

When KWHERE equals three or four, convergence on a solution to the nozzle admittance boundary equation has been attained. For this case, control is passed to that portion of the main program which prints the final results for that solution.

When KWHERE equals five, control is passed to the end of the main do loop in the main program. The output portion is bypassed. This only occurs when an error was detected by subroutine SØLVW.

## SØLVW

This subroutine performs many of the calculations and controls most of the logical flow required to match the downstream boundary condition on the nozzle admittance. The FORTRAN variable ISCNT controls the logical flow within subroutine SØLVW (Fig. 6). If ISCNT equals one, then that portion of the subroutine used for searching the  $\omega$ -plane for possible solutions to the nozzle admittance boundary equation is used. Two calls to SØLVW are used for this purpose. During the first call, the imaginary part of  $\omega$  is determined so that the absolute value of the upstream nozzle admittance minus the downstream nozzle admittance is minimized. The second call is made in order to complete the computation of the Jacobian of this difference with respect to the real portion of  $\omega$ . (The derivatives of the difference with respect to the imaginary part of  $\omega$  are computed during the first call.) When ISCNT equals one, tests are also made to determine if a solution is nearby. The actual test function and the logic employed is described on pages 54-56.

Once it is determined that a possible solution is bracketed by two successive frequencies, the variable ISCNT is set equal to four. Subroutine SØLVW performs the same calculations for this value of the variable ISCNT as it does when ISCNT equals one. The only difference occurs at the end of the second call to SØLVW. At that point, checks are made to ensure that the potential solution is in fact an actual solution and not due to a singularity in the Jacobian. If the error passes certain criteria and at least three iterations have been performed with ISCNT equal to four, then ISCNT is set equal to five for subsequent calls to SØLVW. Between each iteration for ISCNT equal to four, as well as for the first iteration for ISCNT equal to four, the real part of the frequency is modified using the method of false position or the bisection method, depending upon the value of the iteration counter, KSCNT4.

When ISCNT equals five, no derivatives are computed. For this situation, subroutine SØLVW only computes the difference between the upstream and downstream nozzle admittances based on the current value of  $\omega$ . It then checks to determine whether convergence has been obtained. If not, the

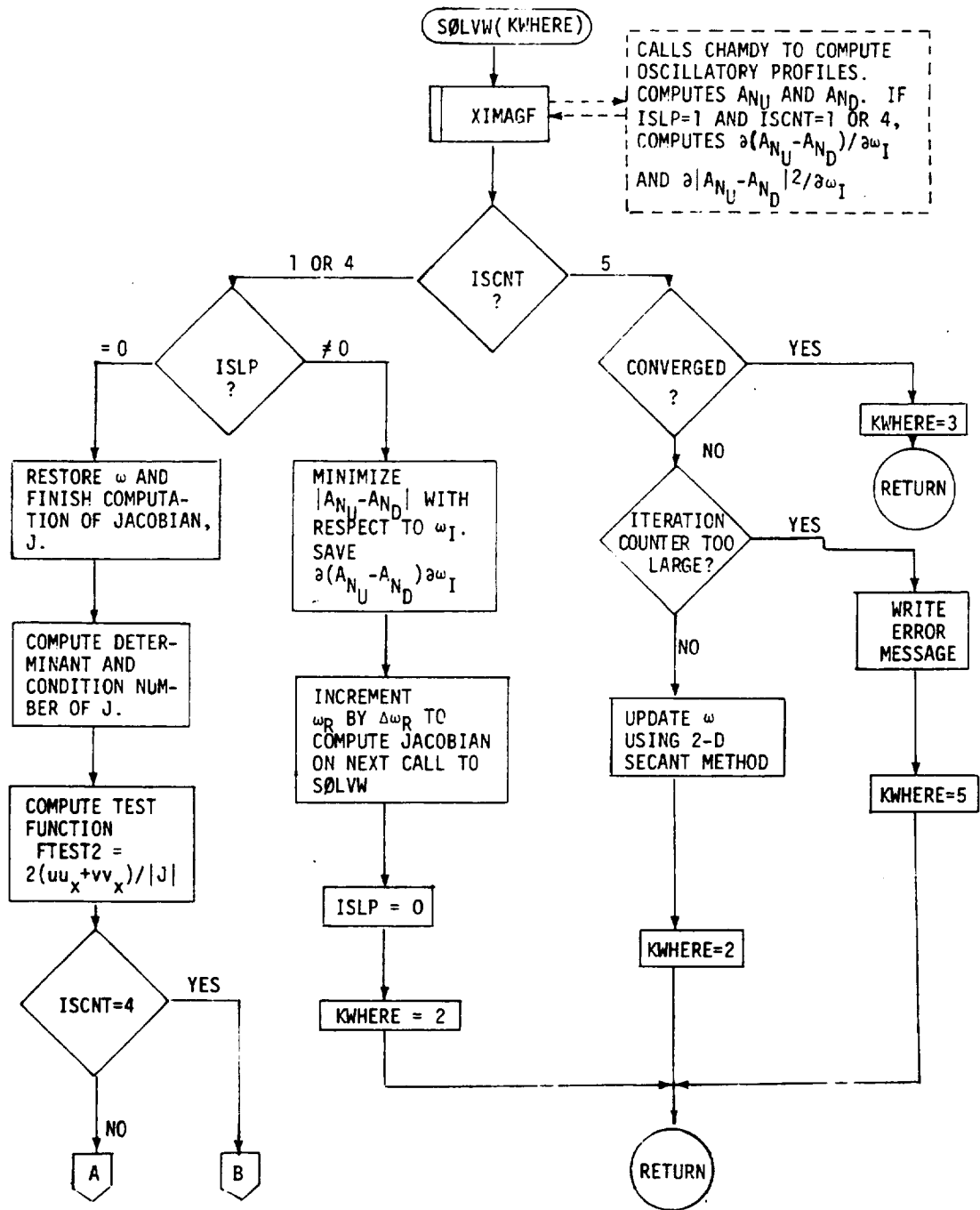


Figure 6. Logic Diagram for Subroutine SØLVW

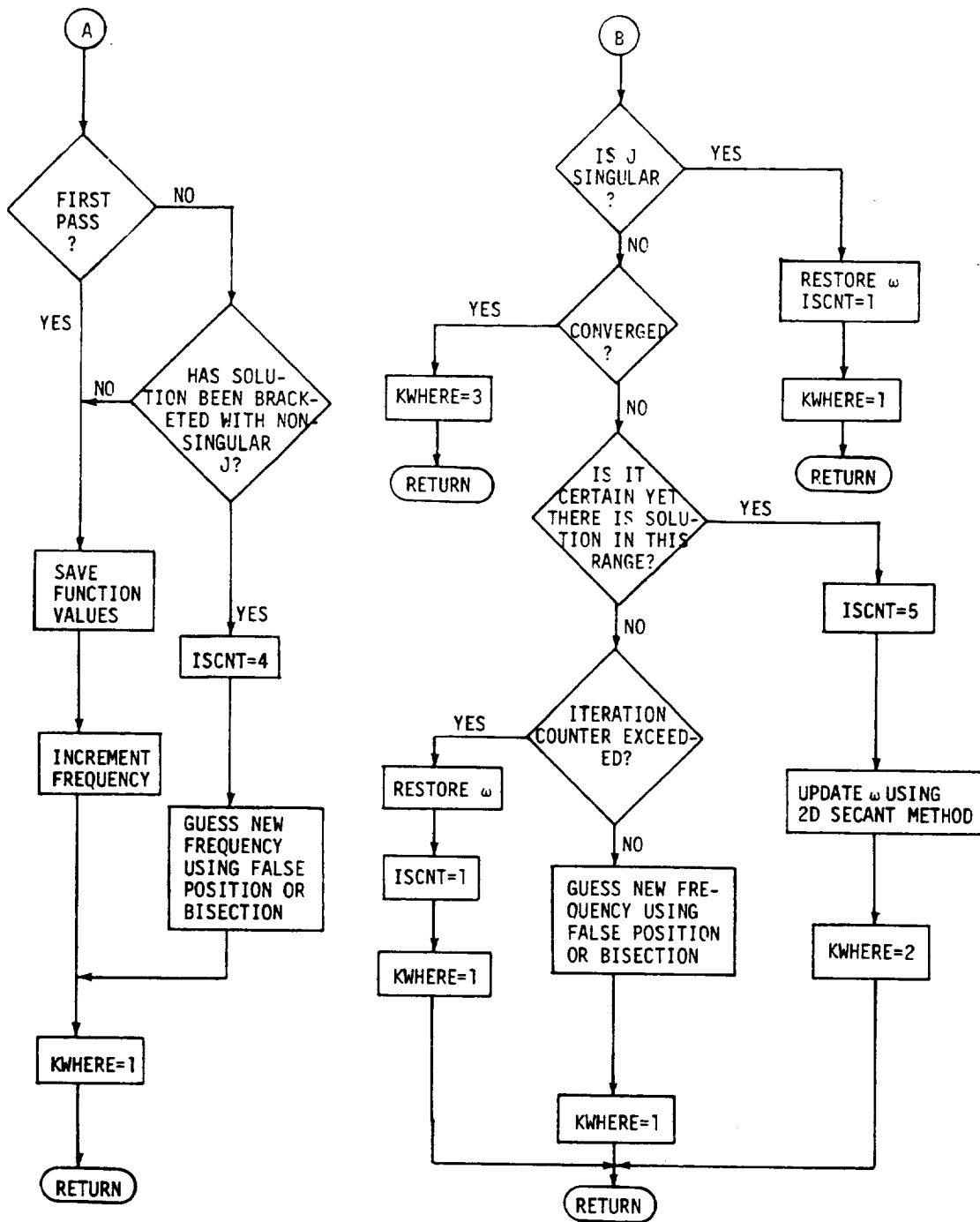


Figure 6. (Concluded)

value of  $\omega$  is updated according to the two-dimensional secant method described on page 53.

The variable which controls the flow in the main program subsequent to a call to SØLVW is KWHERE and is described in the section of this report dealing with the main program.

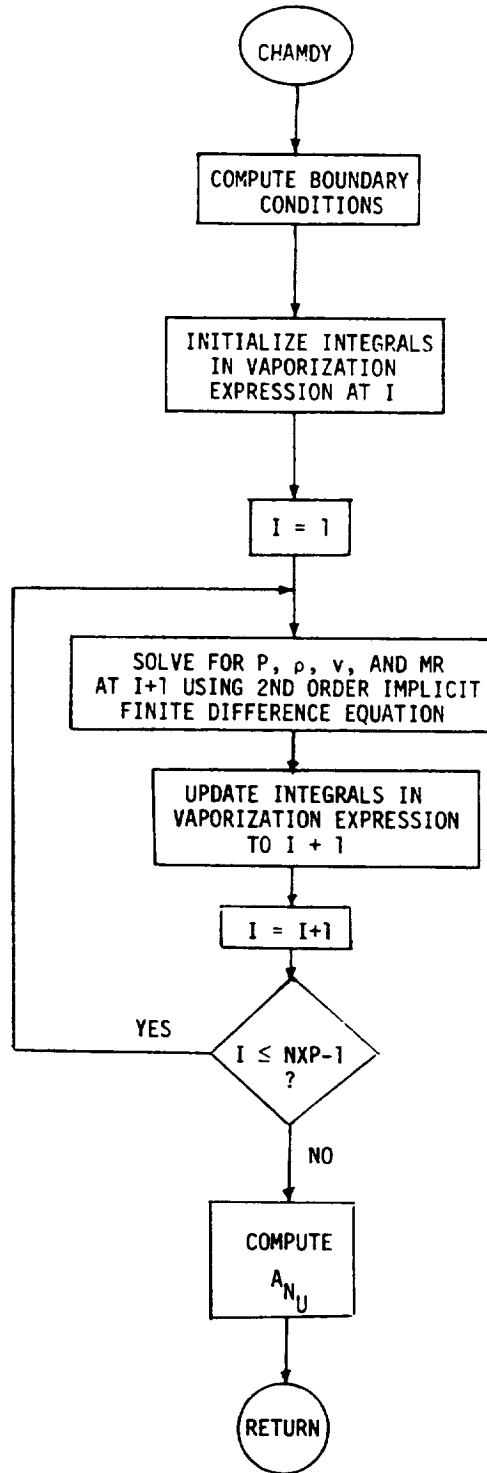
#### CHAMDY

This subprogram is called by XIMAGF in order to compute the oscillatory pressure, temperature, velocity, mixture ratio, and density profiles. From these quantities (Fig. 7), it determines the upstream nozzle admittance. It solves for the oscillatory profiles by solving the linearized set of differential equations presented on pages 41 and 42. This is done using a second order implicit finite difference scheme. Those integrals appearing in the vaporization expression which cannot be integrated analytically are numerically integrated using the trapezoidal rule.

Because the differential equations represent a linear initial value problem, the finite difference equations are also linear and one can "march off" the solution from the initial plane. Since the four differential equations are coupled, replacing them at each axial position by their finite difference approximation results in a four by four system of complex linear equations. Because of the nature of the differential equations, the resulting matrix equations are essentially diagonally dominant and can therefore be solved very quickly using Gaussian elimination with the diagonal element used for pivoting.

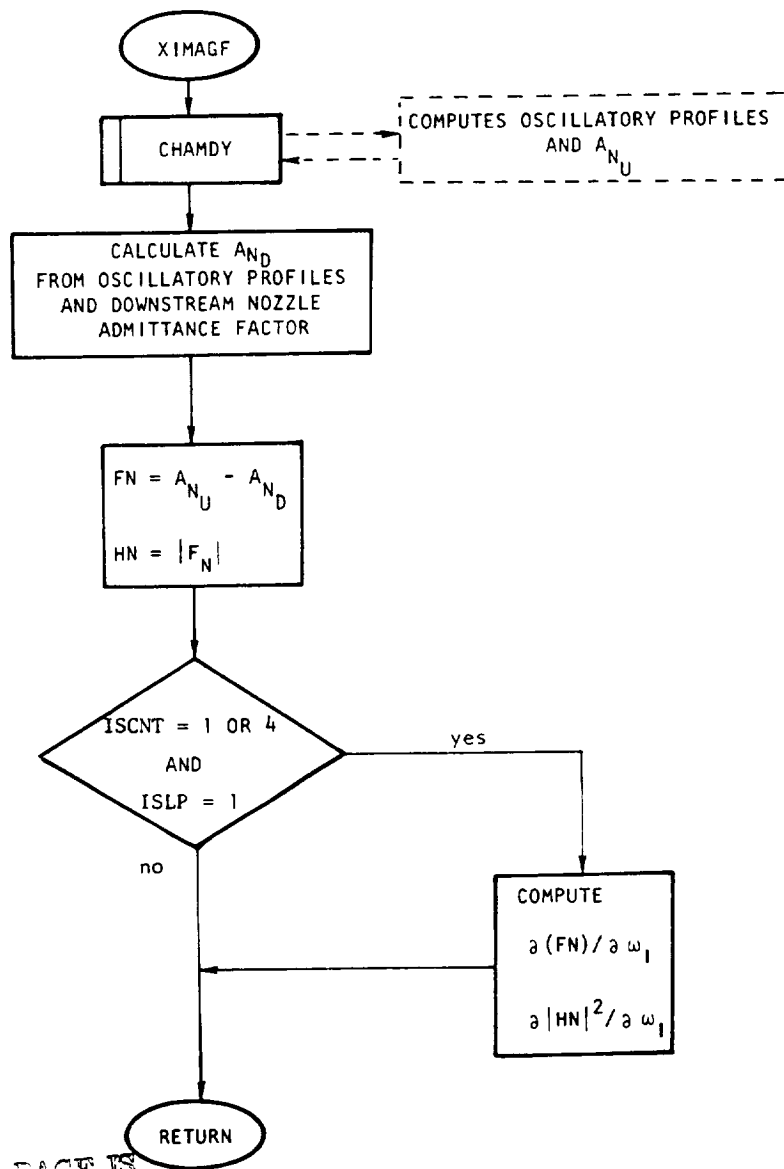
#### XIMAGF

This subroutine is called by SØLVW and ZERØ. Its main function is to compute the difference between the upstream and downstream nozzle admittances (Fig. 8). When the program is still performing the search algorithm, this routine computes the derivative of this difference with respect to the imaginary part of  $\omega$  as well as the derivative of the absolute value of this difference squared with respect to the imaginary part of  $\omega$ .



ORIGINAL PAGE IS  
OF POOR QUALITY

Figure 7. Logic Diagram for Subroutine CHAMDY



ORIGINAL PAGE IS  
OF POOR QUALITY

Figure 8. Logic Diagram for Subroutine XIMAGF

#### CHMCØN

This routine is called by CHAMDY and calculates certain parameters necessary for determining the coefficients used in CHAMDY.

#### CØMMAT

This routine solves the four by four complex system of linear equations discussed in the section describing subroutine CHAMDY. It uses Gaussian elimination with pivoting on the diagonal.

#### ZERØ

This routine is called by subroutine CHAMDY. Its function is to find the zero of a given functional when that zero is bracketed both above and below. The functional in this case is the derivative with respect to the imaginary part of  $\omega$  of the absolute value squared of the difference between the upstream and downstream nozzle admittances. Finding the zero of this functional is done in order to minimize the error in the difference between the nozzle admittances with respect to the imaginary part of  $\omega$ . The method used by subroutine ZERO is due to Muller (Ref. 25). It essentially involves a bisection step followed by inverse parabolic interpolation to determine the next guess.

#### STEADY

This routine (Fig. 9) is called by the main program to determine the time independent solution to the set of differential equations given on page 40. These equations have been analytically integrated on pages 42-44. This subroutine uses these latter equations to determine the steady state profiles. Also, several parameters which are a function of these steady state variables are computed and saved for subsequent use by the chamber dynamics subprogram, CHAMDY. If the FORTRAN variable IPRSTE is greater than zero, a printout of the steady state profiles will be given.

#### CØMBDY

This subprogram (Fig. 10) calculates the fuel and oxidizer combustion coupling coefficients required for the determination of the time oscillatory vaporization rates needed to solve the chamber dynamics. The equations for these parameters



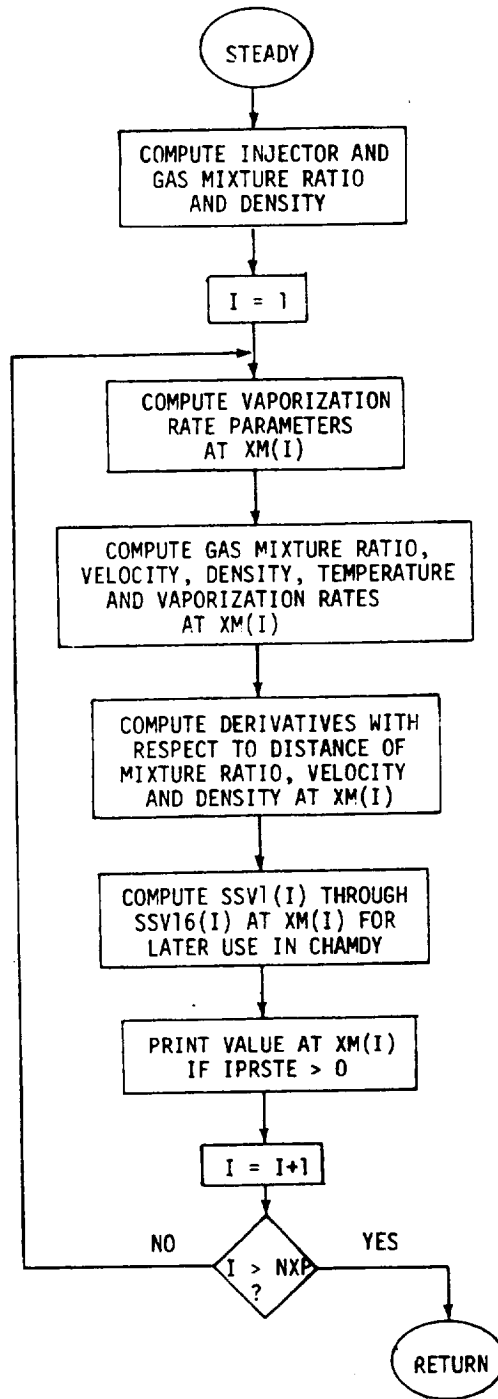


Figure 9. Logic Diagram for Subroutine STEADY

ORIGINAL PAGE IS  
OF POOR QUALITY

R-9808/69

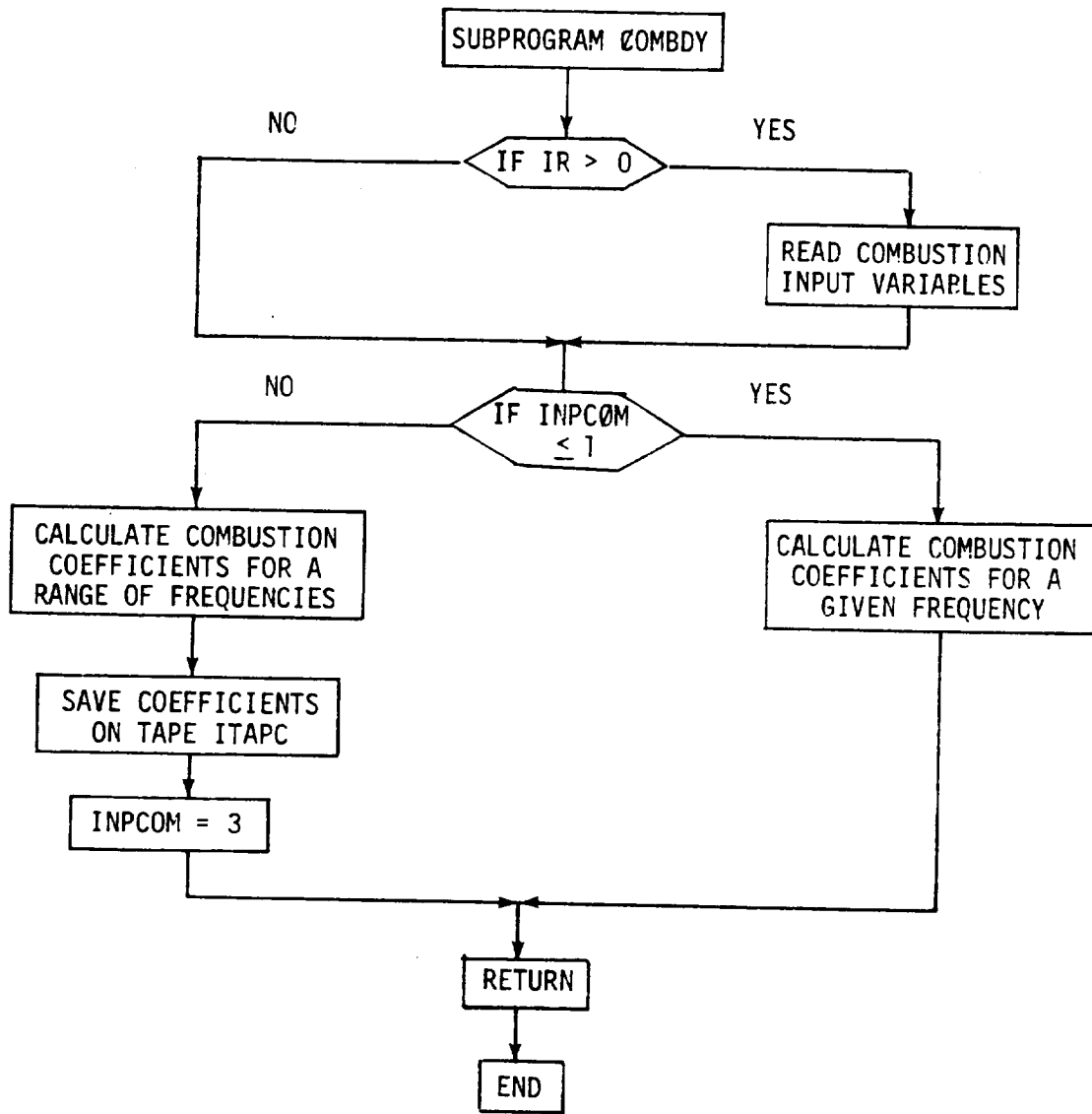


Figure 10. Logic Diagram for Subprogram CØMBDY

are given on pages 33-34. During the first pass into this program, the combustion dynamic input variables are read in from input device 5 and written out onto device 6. A discussion of these variables is given in the Model Input Section. The logical flow in CØMBDY is controlled by the FØRTRAN variable INPCØM. If this variable is less than or equal to one, the combustion coefficients are computed for the current frequency only. If INPCØM is greater than one, these coefficients are computed for the entire frequency table, FREQT (e.g., from 10 to 1000 Hz), and saved on tape/disk ITAPE for subsequent use by the main program.

#### AREA

This subroutine is called by the main program. It computes the area profiles and axial distance profiles necessary for solution of the steady-state and transient profiles.

#### LØCFAC

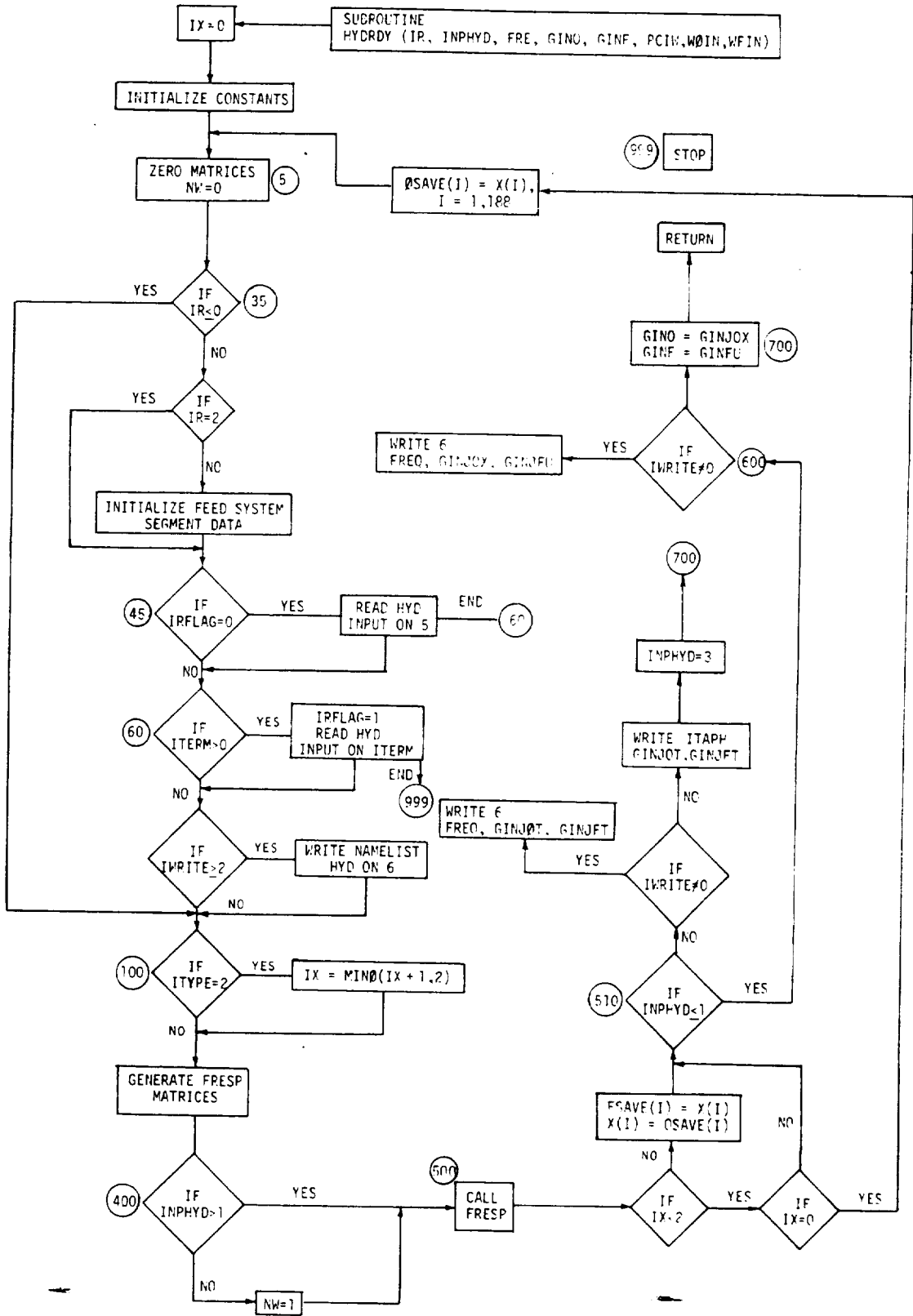
This routine is used to determine the subscript, I, within an ordered array, TX, such that the input argument, X, is in the interval TX(I), TX(I+1). This routine also returns the interpolating factor  $FX = (X - TX(I)) / (TX(I+1) - TX(I))$  for subsequent use in linear interpolation.

#### HEAD

This subroutine is called by the main program to print the heading page which gives the title of the program, by whom and where it was developed, and the program sponsor.

#### HYDRDY

Subroutine HYDRDY (Fig. 11) is called by the main program to calculate the frequency domain characteristics of the feed system. Functions performed by HYDRDY are (1) reading of input data describing the physical attributes



ORIGINAL PAGE IS  
OF POOR QUALITY

Figure 11. Logic Diagram for Subprogram HYDRDY

of the feed system components, (2) generation of a matrix of linear differential equations representing the complete feed system, (3) solution of the feed system equations to yield the amplitude and phase response of all feed system pressures and flowrates as a function of chamber pressure oscillation amplitude and frequency, and (4) generation of tabulated output of injector flowrate frequency response for use by the main program.

A basic assumption of subroutine HYDRDY is that the feed system being modeled can be represented by the generalized schematic of Fig. 4 (or by some portion of this schematic). This assumption is necessary because HYDRDY sets up and solves the complete set of simultaneous equations representing the Fig. 4 schematic. By assigning very high resistance and very short length attributes to any of the 30 numbered line segments of the generalized schematic, whose segment can effectively be excluded from having any effect on the frequency response characteristics of the rest of the system. With this approach a wide variety of feed systems can be modeled with no changes to the program other than the input data.

Figure 11 shows the functional block diagram for subroutine HYDRDY. When called, the subroutine initially zeroes the values of all of the elements of the coefficient and time delay matrices C and TD in labeled common block F. The values of various fixed input arguments required by the frequency response subroutine (FRESP) are then initialized.

Input argument IR directs the reading of subroutine HYDRDY input data. If IR is zero or less, the program assumes that all required data has previously been read and the data read function is bypassed. If IR = 1, the program assumes that no hydrodynamic data has been read and proceeds to initialize all hydrodynamic input variables to values which will exclude all 30 line segments and both injectors of the generalized Fig. 4 feed system schematic. Control is then passed to statement 45 for reading of input data. If

IR = 2, the assumption is made that most input data is already set up (such as from a previous case during the same program run). Control is passed directly to statement 45 for reading any changes to the input data.

Input data reading for HYDRDY is in the NAMELIST format (NAMELIST name HYD) and is normally in the form of card input on logical unit number 5. However, if the program is run in a timesharing environment, an option is provided for reading data from a timesharing terminal. This option is controlled by variables IRFLAG and ITERM. Both of these variables are stored in labeled common block/F/ and can be changed by input to NAMELIST HYD. Variable IRFLAG is tested in statement 45 and if non-zero specifies reading NAMELIST HYD data from unit 5. If timesharing terminal input is desired, variable ITERM is set (by input data read or block data initialization of labeled common block/F/) to the logical unit number to the timesharing terminal. If variable ITERM is non-zero, statement 46 sets IRFLAG to 1. Thus, once terminal data input has been specified, all subsequent data reads will default to the terminal. Card input can be respecified (for a subsequent data case) by entering IRFLAG = 0 in the terminal data input.

Input variables for subroutine HYDRDY are described in detail in the Hydrodynamic Input Section. The variables include the length (L), cross-sectional area (A), propellant sonic velocity (V), propellant density (RHOL), hydraulic resistance (R), and wall compliance (CW) for each of the 30 numbered waterhammer segments in the generalized Fig. 4 feed system schematic. Input variables for the left ("O") injector of Fig. 4 are resistance (RO), inertance (ZO), volume (VOL), propellant sonic velocity (VO), and injector deflection constant (KO). The corresponding input variables for the right ("F") injector are RF, ZF, VOLF, VF and KF. The designation of the two injectors as "O" and "F" is a notational convenience for cases in which the feed system being modeled has only one injector and sufficiently simple flow paths so that both oxidizer and fuel systems can simultaneously be laid out on the Fig. 4 schematic. Such cases have the advantage of reduced computer time because the

frequency response of both fuel and oxidizer feed systems is obtained with a single call to subroutine HYDRDY. Of the fuel and oxidizer feed systems overlap when laid out on the Fig. 4 schematic, subroutine HYDRDY must be called twice - once for each feed system.

When data input is complete, a value of 2 (or greater) for variable IWRITE specifies a printout of all input data on logical unit 6. IWRITE = 0 is the default and specifies no printout of input data.

Next, control is passed to the DO loop at statement 100 in which the input values for propellant density, propellant acoustic velocity and segment wall compliance for each of the 30 waterhammer segments of the Fig. 4 schematic are combined to yield an effective acoustic velocity for each segment. The subsequent statements, up to statement 400, combine the input variables as required to yield the constant coefficients of the 57 linear waterhammer and injector equations describing the complete Fig. 4 feed system. Simultaneous solution of these 57 equations, at each specified input frequency, yields the oscillatory amplitude and phase response of all pressures and flowrates in the feed system to inputs via chamber pressure oscillations at that frequency.

At statement 500 a call to subroutine FRESP is made to obtain the frequency response solution of the feed system equations. Initially, however, at statement 400 the value of input argument INPHYD is tested to determine the desired output from FRESP. If INPHYD is greater than 1, HYDRDY will call FRESP to calculate the feed system frequency response for each of the frequencies in array FREQT. The total number of frequencies is given by variable NFREQT and may range from 1 to 100. Both the variable, NFREQT, and the array, FREQT are in labeled common block/COMTAP/. If the value of INPHYD is less than or equal to 1 HYDRDY will call FRESP to calculate the feed system frequency response for the single frequency given by input argument, FRE.

Output data from subroutine HYDRDY consists of a pair of complex numbers for each specified input frequency. If INPHYD was specified as  $\leq 1$  then the output numbers GINJ0X and GINJFU are returned in labeled common block/F/ and also in the HYDRDY argument list as GIN0 and GINF. The real

and imaginary parts of complex number  $GINJ\phi X$  ( $GIN\phi$ ) represent the amplitude and phase angle respectively of  $\frac{\Delta W\phi}{W\phi IN} / \frac{\Delta PC}{PCIN}$  at frequency  $FREQ$  ( $FRE$ ). Similarly, the complex number  $GINJFU$  ( $GINF$ ) represents the amplitude and phase angle of  $\frac{\Delta W\phi}{WFIN} / \frac{\Delta PC}{PCIN}$  at frequency  $FREQ$  ( $FRE$ ),  $W\phi IN$ ,  $WFIN$  and  $PCIN$  are the input normal values for the oxidizer injector flowrate, fuel injector flowrate and chamber pressure, respectively, from the HYDRDY argument list.

If INPHYD was specified as >1, then rather than a single pair of complex numbers representing oscillatory injection flowrates, HYDRDY returns two arrays,  $GINJ\phi T$  and  $GINJFT$ , containing the oscillatory injection flowrate amplitude and phase data for each of the NFREQT frequencies in array FREQT. The output arrays  $GINJ\phi T$  and  $GINJFT$  are stored in labeled common block /COMTAP/ and are also written out on the output device whose logical unit number is designated by variable ITAPH in labeled common block /COMTAP/. The order of storage on the output device is  $GINJ\phi T(I)$ ,  $GINJFT(I)$ , for I values from 1 through NFREQT. After writing the  $GINJ\phi T$  and  $GINJFT$  arrays on the output device HYDRDY sets the value of variable INPHYD to 3. Also, before returning control to the main program, HYDRDY tests the value of variable IWRITE. If IWRITE is non-zero, each specified frequency and the corresponding value of  $GINJ\phi T$  and  $GINJFT$  are written out on logical unit 6. If only one frequency was specified ( $INPHYD \leq 1$ ), then only the single point values of  $FREQ$ ,  $GINJ\phi X$  and  $GINJFU$  are written out.

It should be noted that although output from a single call to HYDRDY contains values for both "oxidizer" and "fuel" oscillatory injection flowrates (at one or more frequencies), the output values actually refer to the " $\phi$ " and "F" injectors of the Fig. 4 schematic. Thus, unless both oxidizer and fuel feed systems can simultaneously be modeled with the Fig. 4 layout, it is necessary to call HYDRDY twice - once for the oxidizer feed system and once for the fuel feed system.

#### FRESP

Subroutine FRESP (Fig. 12) is used to obtain the frequency domain characteristics of the feed system indirectly from input data that describes the physical



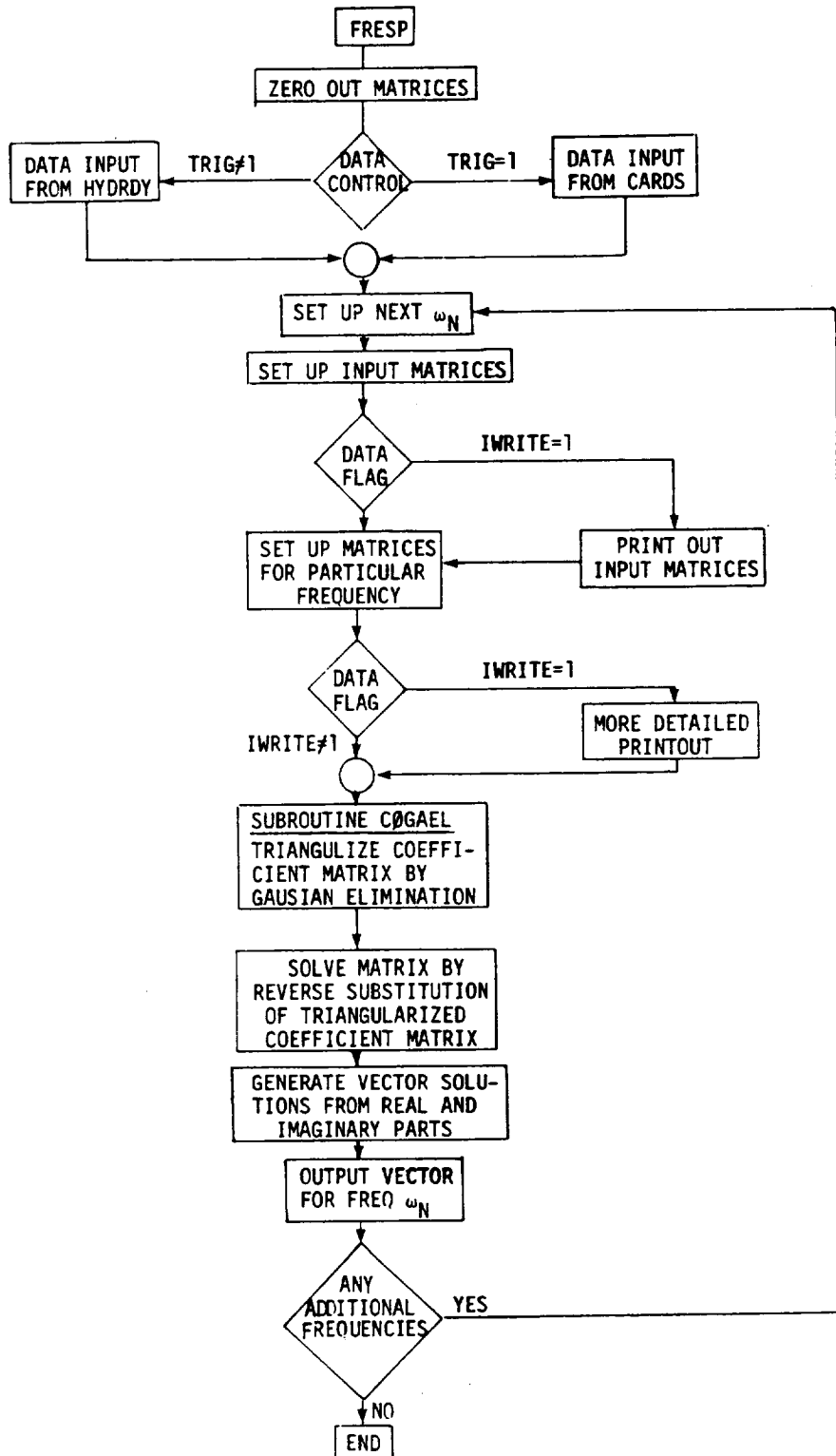


Figure 12. Logic Diagram for Subroutine FRESP

ORIGINAL PAGE IS  
OF POOR QUALITY

R-9808/77

characteristics of the feed system. (The actual input to FRESP is generated by the subroutine HYDRDY which orders the physical characteristics of the system into specific matrices of coefficients that FRESP can use as input data.)

FRESP merely solves for the variables  $X_i$  in the following relationship:

$$C \{X\} = a Y$$

where  $Y$  is a single input variable that represents a unit value of the injector end combustion chamber pressures. The matrix  $a$  then relates the specific pressure input to each applicable equation that contains combustion chamber pressure ( $a$  may contain both static and dynamic terms.) The matrix  $C$  is simply the coefficients of the linear differential equations that represent the physical system. The values of the coefficients for the  $a$  and  $C$  matrices are computed by the subroutine HYDRDY.

The FRESP matrices can be expressed as:

$$C_{ijk} S^{k-1} \cdot \{X_j\} = a_{ik} S^{k-1} \cdot Y$$

with the differential operator defined as  $S = J\omega$ , where  $J = \sqrt{-1}$  and  $\omega$  is the frequency. The matrices may be broken down to provide real matrices and imaginary matrices.

$$C_{ij1} - C_{ij3} \omega^2 + C_{ij5} \omega^4 - \dots + J C_{ij2} \omega - C_{ij4} \omega^3 + \dots \\ \cdot X_j = a_{i1} - a_{i3} \omega^2 + \dots + J a_{i2} \omega - a_{i4} \omega^3 + \dots \cdot Y \quad (213)$$

Since the time delay coefficients used in the differential equations are of the form  $e^{-\tau S} \cdot X$ , which is equivalent to  $e^{-\tau j \omega} \cdot X$ , and since  $e^{-jy} = \cos(y) + j \sin(y)$ , these terms may be added to the previously formed real and imaginary matrices to give:

$$\left\{ \left[ \begin{array}{l} C_{ij1} - C_{ij3} \omega^2 + \dots + \cos(\tau_{ij} \omega) \\ J [C_{ij2} \omega - C_{ij4} \omega^3 + \dots + \sin(\tau_{ij} \omega)] \end{array} \right] \right\} \cdot [X_j] = \\
 \left\{ \left[ \begin{array}{l} a_{i1} - a_{i3} \omega^2 + \dots - \cos(\tau_i \omega) \\ J [a_{i2} \omega - a_{i4} \omega^3 + \dots + \sin(\tau_i \omega)] \end{array} \right] \right\} Y \quad (214)$$

and solved for  $[X_i]$  :

$$[X_i] = \left\{ \left[ \begin{array}{l} C_{ij1} - C_{ij3} \omega^2 + \dots + \cos(\tau_{ij} \omega) \\ J [C_{ij2} \omega - C_{ij4} \omega^3 + \dots + \sin(\tau_{ij} \omega)] \end{array} \right] \right\} \cdot \\
 \left\{ \left[ \begin{array}{l} a_{i1} - a_{i3} \omega^2 + \dots + \cos(\tau_i \omega) \\ J [a_{i2} \omega - a_{i4} \omega^3 + \dots + \sin(\tau_i \omega)] \end{array} \right] \right\} \cdot \quad (215)$$

The matrices are multiplied and then solved for  $[X_i]$  in the subroutine C0GAEL which employs the standard Gaussian elimination procedure for solving linear equations. The  $[X_i]$  solution is still separated into real and imaginary components, and are simply combined to form a vector for each variable. The procedure is repeated for each frequency being considered.

### CØGAEL

This subroutine is called by the hydrodynamic frequency response subroutine, FRESP, to triangularize the complex matrix of feed system equations. Back substitution into the triangular system of equations is subsequently performed by subroutine FRESP to yield the real and imaginary portions of each feed system variable.

The conventional method of Gaussian elimination is employed by CØGAEL to triangularize the system of equations. The reduction process proceeds in column order from left to right. First the complex element with the largest absolute value in the current ("pivot") column at or below the diagonal is located. Then the rows are interchanged if required to move this maximum element (pivot element) to the diagonal. The row interchange serves to minimize the round-off errors from the subsequent reduction process. The pivot row (row containing the pivot element) is then divided by the pivot element yielding 1.0 from the pivot element. Finally, the elements in the pivot column below the diagonal are eliminated by subtracting the appropriate multiple of the pivot row from each row below it. The subtraction is not actually performed on the elements below the diagonal since these elements do not enter into the subsequent back substitution process performed by subroutine FRESP.

It should be noted that the above discussion refers to the complex matrix as if the elements were single numbers. The actual elements are stored as two numbers in each row, the real portion to the left of the constant term, and the imaginary portion on the right. This distinction does not alter the elimination process except that two separate numbers must be operated on at each step.

### TDPLØT

If the value of ICRT is greater than 0 this subroutine is called by the frequency response routine FRESP to generate CRT plots of the gain and phase of the output variables as a function of frequency.

The input arguments to TDPLØT are, W, an array of up to 101 frequencies; Y, an array of gain or phase values; NFP, the number of data points in arrays W and Y; TL, the lowest desired frequency grid line; XR, the highest desired frequency grid line; and LL, a flag indicating a gain plot if 1 or a phase plot if 2.

Initially TDPLØT scans the Y array for the maximum and minimum values and generates values for the Y axis grid scaling. The first value of Y is not included in this scan. This allows an initial very low value of frequency to be used to approximate the system DC frequency response without upsetting the plot frequency scaling. TDPLØT uses the standard graphics package routines for the SC-4020 to generate the plot grids and plot the data points. If the value of LL is 1 the CRT frame is advanced and a plot of Y(I) versus W(I) is made on the bottom half of the page. If LL is 2, the frame is not advanced and the plot is made on the top half of the page.

In addition to the plots, TDPLØT prints the numeric value of the first Y array element immediately above the corresponding plot. This element typically corresponds to a very low frequency value (default value of .001 cps in subroutine FRESP) which is well below the frequency range desired for the plot and approximates the DC value of the output variable.

TDPLØT does not generate any titles or identifying information on the plots.

NQZADM, (RKTDIF, RKTZ, RKZDIF, TADAMS, ZADAMS)

This routine and its support routines, is called by the main program to determine the nozzle admittance based on downstream conditions. The programs were developed and programmed by Georgia Institute of Technology and the user is referred to Ref. 23 for a complete description of these routines. The main nozzle admittance program was modified by Rocketdyne so that input data could be read if required and also the nozzle admittance saved on a tape unit ITAPN (Fig. 13).

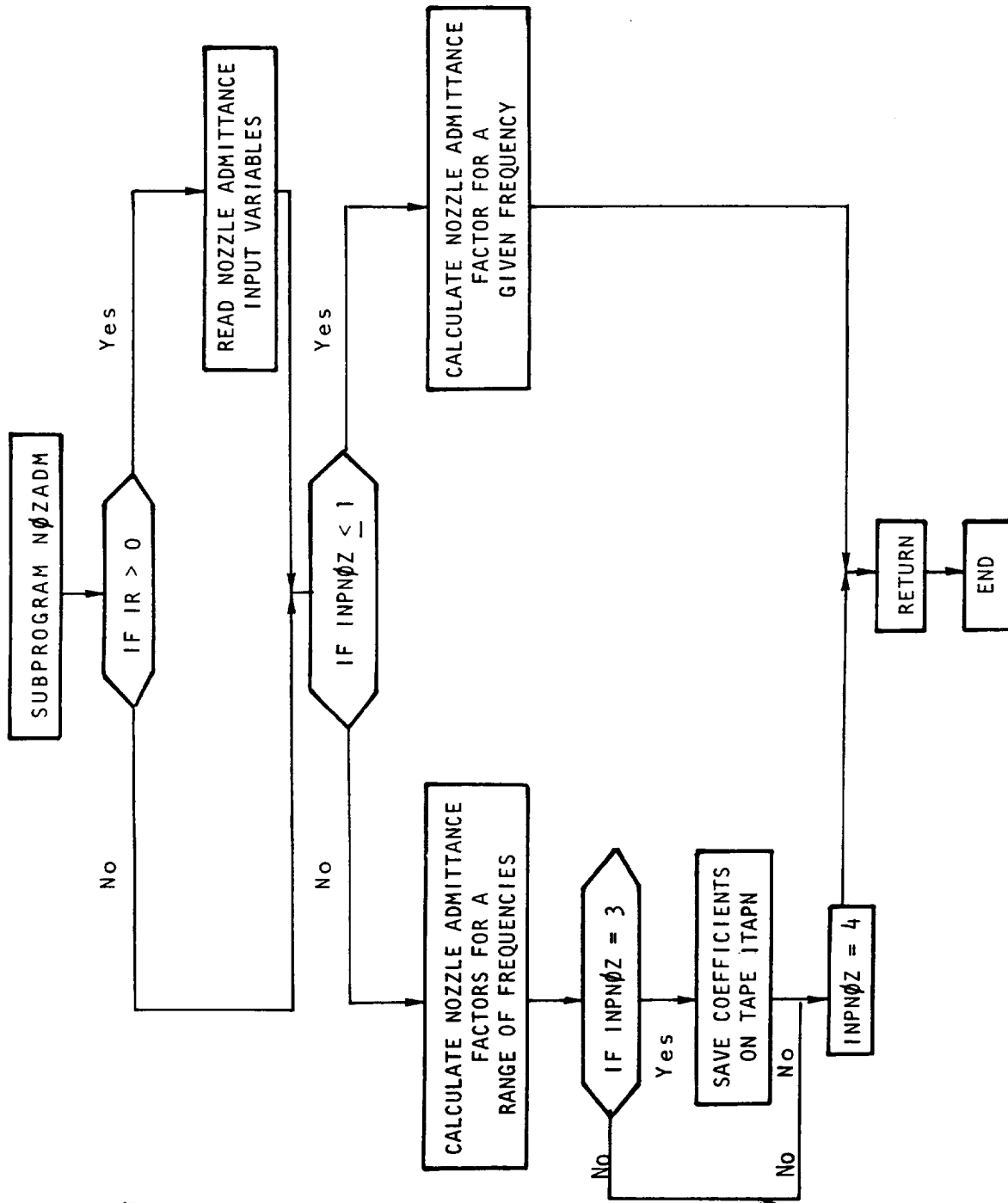


Figure 13. Logic Diagram for Subroutine NØZADM

## FSCSM COMPUTER MODEL INPUT

This section, and the Hydrodynamic Input section, describes the input necessary to run the FSCSM computer model. The input is broken up into four major control sections. These are the main control section input, the nozzle admittance control section input, the hydrodynamics control section input, and the combustion dynamics control section input. Table 2 lists all the variables that are to be input for each control section. This input is in the usual 80 character card form. Listed in Table 2 for each control section are each card's number and format, the variable names appearing on each card, and a brief description of each variable appearing in the list.

The main control section requires either eight or nine cards depending upon the input value of INPNØZ. (If INPNØZ is less than or equal to three, the last card in this section should be input). The first two cards, cards 1 and 2, should contain title and case identification information. These are read in alpha numeric format and printed at the top of almost every page of output to identify the case being considered.

The third card contains control information for various run options, file numbers for the auxiliary storage of datasets used by the program, print codes, and the number of mesh points to be used in the analysis. The control flags are INPHYD, INPCØM, and INPNØZ. These flags allow the user to save the results from the current case or use the results from a previous case for the current case. It is recommended that these datasets be set up as permanent files in order to use them for subsequent job submittals.

The first of these flags, INPHYD, controls the hydrodynamics feed system dataset. If it is input as one, no action will be taken with respect to saving or reading information on or from a dataset. The hydrodynamics coupling terms will be recomputed each time the frequency changes. If INPHYD is input as two, a table of hydrodynamic coupling terms will be generated for the frequency range specified by the input variables NFREQ, FREQMI, and FREQMX. This table will be saved on file ITAPH (also input on this card) and used to linearly interpolate in each time the frequency changes. If INPHYD is input as three, the program assumes a



TABLE 2. FSCSM INPUT DATA

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Main Program Input	THESE CARDS MUST ALWAYS BE INPUT		
	1,2 (18A4)	((TITLE(I,J), I=1,18), J=1,2)	Title information is input on the first two cards in the input data-set. These should be used to identify the case being run.
	3 (12I6)	INPHYD	Code used for hydrodynamic calculations. 1: Hydrodynamic coupling terms (HCT) will be computed each time the frequency changes. 2: A table of HCT's will be computed using frequency table, FREQT, and saved on file ITAPH. Each time the frequency changes in trying to satisfy the nozzle admittance boundary condition, the HCT will be interpolated for in that table. 3: Table of HCT's already resides on file ITAPH from an earlier run. When the frequency changes, the HCT will be interpolated for in that table.
		INPCØM	Code used for the combustion dynamics coefficients (CDC). It can take on values from one to three. It has the same meaning with respect to the CDC as INPHYD does with respect to the HCT. The tables are saved on file ITAPC.
		INPNØZ	Code used for the downstream nozzle admittance term (DNAT). 1: The DNAT is computed each time the frequency changes. 2: A table of DNT's versus frequency using frequency table, FREQT, is computed and saved internally. Each time the frequency changes in trying to satisfy the nozzle admittance boundary condition, the DNT will be interpolated for in that table.

TABLE 2. (Continued)

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Main Program Input (Cont.)			3: Same as 2, but the table is also saved on file ITAPN.
			4: A table of DNT's versus frequency already resides on file ITAPN (from a previous run). Interpolation is the same as 2 and 3.
		ITAPH	File used to save hydrodynamic coupling term table. Must be specified if INPHYD>2.
		ITAPC	File used to save combustion coefficients table. Must be specified if INPCOM>3.
		ITAPN	File used to save nozzle admittance terms table. Must be specified if INPNØZ>3.
		IPRHYD	Code used for hydrodynamics coupling term table printout. 0: Table will not be printed. 1: Table will be printed.
		IPRCOM	Code used for combustion coupling coefficients printout 0: Coefficients will not be printed. 1: Coefficients will be printed at each frequency found to satisfy the nozzle admittance boundary condition.
		IPRNØZ	Code used for the downstream nozzle admittance terms table printout. 0: Table will not be printed. 1: Table will be printed.
		IPRCHM	Code used for the oscillatory profiles printout. 0: Profiles will not be printed. 1: Profiles will be printed at each frequency found to satisfy the nozzle admittance boundary condition.

TABLE 2. (Continued)

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Main Program Input (Cont.)		IPRSTE	Code used for the steady-state profiles printout. 0: Profiles will not be printed. 1: Profiles will be printed.
		NXP	Number of axial positions to be used between the X0 plane (start of vaporization) and the nozzle inlet plane. (Both the X0 and nozzle inlet planes must be counted.)
	4 (6E12.8)	X0	Axial coordinate of start of vaporization plane, inches.
		XNØZ	Nozzle inlet plane, inches
		RINJ	Radius of combustion chamber at injector face, inches.
		GAMØ	Ratio of specific heats of combustion gas ( $C_p/C_v$ ) evaluated at overall mixture ratio, unitless.
		CØ	Sonic velocity evaluated at overall mixture ratio, ft/sec.
		DEL P	Oscillatory non-dimensional pressure amplitude at injector face, unitless.
	5 (6I6)	NRØØT	Absolute value of this variable is the number of frequency solutions that will be searched for starting at the frequency specified by the real part of ØMEGA and ending at FROMAX. This variable may be input as either positive or negative. (See input of ØMEGAR for explanation.)
		IWRT	Intermediate output dump code used to write the oscillatory profiles solved for in CHAMDY for each iteration. 0: Oscillatory profiles will not be printed out between iterations. 1: Oscillatory profiles will be printed for each iteration.

TABLE 2. (Continued)

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Main Program Input (Cont.)		IWSKP	Intermediate output dump code used in subroutine SØLVW. 0: No intermediate output will be printed from SØLVW during iterations. 1: Limited intermediate output will be printed by SØLVW 2: Extended intermediate output will be printed by SØLVW.
		KNTMX	Maximum number of iterations allowed to minimize the error in the nozzle admittance boundary equation with respect to the imaginary part of $\omega$ .
		KNTRMX	Maximum number of times the frequency will be allowed to be changed by DELFRQ during the searching algorithm between each solution.
		KNTSMX	Maximum number of iterations allowed for the convergence of the two-dimensional secant method used in SØLVW.
	6 (6E12.8)	ØMEGAR	Starting value for the real part of complex frequency of NROOT>0. This should be input in units of Hertz times $2\pi$ . If NROOT<0, this should be input in Hertz.
		ØMEGAI	Starting value for the imaginary part of complex frequency. This should be input as the growth coefficient if NROOT>0. It should be input as the decrement if NROOT<0.
		FRQMAX	Maximum frequency above which no solutions to the nozzle admittance boundary equation will be sought, Hertz.
		DELFRQ	Increment used to adjust the frequency during the searching portion of the algorithm to solve the nozzle admittance boundary equation, Hertz.

TABLE 2. (Continued)

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Main Program Input (Cont.)		DELMX	Maximum allowable change in the growth coefficient between two successive iterations in the portion of the program that minimizes the error in the nozzle boundary condition equation with respect to the growth coefficient, $\text{sec}^{-1}$ .
		CTEST	Upper bound on the condition number of the transpose of the Jacobian of the difference between the upstream and downstream nozzle admittances with respect to the complex frequency. If the condition number of that matrix exceeds CTEST for a given frequency, then it is assumed the Jacobian is singular near that frequency and hence a solution will not be sought at that point. Unitless.
	7 (4E12.8)	EPSF	Relative error criterion used during the search algorithm/or the portion of the program that minimizes the error, HN, in the nozzle admittance boundary equation with respect to the growth coefficient, $\omega_I$ . To obtain convergence, it is necessary that $\frac{\partial  HN ^2}{\partial \omega_I} /  N_D  < \text{EPSF}.$ Unitless.
		EPSX	Relative error criterion used during the search algorithm for the portion of the program that minimizes the error, HN, in the nozzle admittance boundary equation with respect to the growth coefficient, $\omega_I$ . To obtain convergence, it is necessary that $ \omega_{I_1} - \omega_{I_2}  /  \omega_{I_2}  < \text{EPSX}$ where the subscripts 1 and 2 refer to two successive iterations. Unitless.

TABLE 2. (Continued)

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Main Program Input (Cont.)		EPSFS	Tightened relative error criterion used to determine if convergence has been obtained while iterating to solve the nozzle admittance boundary equation. To obtain convergence, it is necessary that $\frac{ A_{N_U} - A_{N_D} }{A_{N_D}} < \text{EPSFS}$ Unitless.
		EPSXS	Tightened relative error criterion used to determine if convergence has been obtained while iterating to solve the nozzle admittance boundary equation. To obtain convergence, it is necessary that $ \omega_1 - \omega_2  /  \omega_2  < \text{EPSXS}$ where the subscripts 1 and 2 refer to successive iterations. Unitless.
		PC	Steady-state chamber pressure, psia.
	8 (3E12.8)	MBØXI	Oxidizer injection flowrate, lbm/sec
		MBFUI	Fuel injection flowrate, lbm/sec
INPUT THIS CARD ONLY IF INPNØZ<3			
	9 (I12,2E12.8)	NFREQT	Number of points in frequency table.
		FREOMI	Minimum frequency in frequency table, Hertz.
		FREQMX	Maximum frequency in frequency table, Hertz.

TABLE 2. (Continued)

CONTROL SECTION	CARD NO./ FORMAT	VARIABLE NAME	DESCRIPTION
Nozzle Admittance Program	1 (4E12.8)	INPUT THE CARD IN THIS CONTROL SECTION ONLY IF INPNØZ ≤ 3	
		RCCX	Ratio of the radius of curvature at the nozzle inlet to the chamber radius at nozzle inlet, unitless (see Fig. 15 ).
		RCTX	Ratio of the radius of curvature upstream of the throat to the chamber radius at nozzle inlet, unitless (see Fig. 15 ).
		ANGLEX	Nozzle convergence half angle, degrees, (see Fig. 15 ).
		CRR	Contraction ratio, cross-sectional area of chamber/throat area, unitless (see Fig. 15 )
Hydrodynamics Program	THIS CONTROL SECTION READS IN ITS INPUT DATA IN NAMELIST FORMAT. THE NAMELIST NAME IS /HYD/. INPUT THIS DATA ONLY IF INPHYD<2.		
	&HYD*	Input these characters starting in column two of the first card of the input.	
	NAMELIST Variables in any order	See Table 3 for a listing of the NAMELIST data input names. The accompanying text describes the meaning of these variables.	
	&END*	Character string denoting the end of the NAMELIST input block.	

\*For Univac 1110 systems, use \$HYD and \$END

TABLE 2. (Continued)

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Combustion Dynamics Program	INPUT THE CARDS IN THIS CONTROL SECTION ONLY IF $INPCOM < 2$		
	1 (6E12.8)	XKØX	Klystron constant for oxidizer jet, inches.
		TAUBØX	Steady-state oxidizer vaporization time delay, sec.
		VBØX	Steady-state oxidizer injection velocity, ft/sec.
		DELHØX	Pseudo energy term for oxidizer, Btu/lbm.
		TDRAGØ	Steady-state oxidizer drag time delay, sec.
		ADVØX	Velocity exponent for the oxidizer atomization process, unitless.
	2 (5E12.8)	ADDØX	Oxidizer liquid jet diameter exponent, unitless.
		DELVØX	Steady-state velocity difference between oxidizer droplets and gas stream normalized to the sonic velocity at the overall mixture ratio, unitless.
		NUBØX	Steady-state oxidizer Nusselt number used in vaporization expression, unitless.
		DTØXDM	Partial derivative of oxidizer vaporization time delay with respect to mixture ratio, holding the vaporization blockage term, drop diameter, and Nusselt number constant, sec.
	3 (6E12.8)	XIMPØX	Oxidizer jet impingement point, inches.
		XKFU	Klystron constant for fuel jet, inches.
		TAUBFU	Steady-state vaporization time delay, sec
VBFU		Steady-state fuel injection velocity, ft/sec.	



TABLE 2. (Concluded)

<u>CONTROL SECTION</u>	<u>CARD NO./ FORMAT</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Combustion Dynamics Program (Cont.)		DELHFU	Pseudo energy term for fuel, Btu/lbm.
		TDRAGF	Steady-state fuel drag time delay, sec.
		ADVFU	Velocity exponent for fuel atomization process, unitless.
	4 (5E12.8)	ADDFU	Fuel liquid jet diameter exponent, unitless.
		DELVFU	Steady-state velocity difference between fuel droplets and gas stream normalized to the sonic velocity at the overall mixture ratio, unitless.
		NUBFU	Average steady-state fuel Nusselt number used in vaporization expression, unitless.
		DTFUDM	Partial derivative of fuel vaporization time delay with respect to mixture ratio, holding the vaporization blockage term, droplet diameter, and Nusselt number constant, sec.
		XIMPFU	Fuel jet injection impingement point, inches.
	5 (5E12.8)	MWG	Steady-state molecular weight of the gas at the overall mixture ratio, lbm/lbm-mole.
		CS	Characteristic velocity at the overall mixture ratio, ft/sec.
		DRGDMR	Partial derivative of gas constant with respect to mixture ratio evaluated at the overall mixture ratio, ft-lb/lb <sup>0</sup> R.
		DCSDMR	Partial derivative of characteristic velocity with respect to mixture ratio evaluated at the overall mixture ratio, ft/sec.
	DHDMR	Partial derivative of gas reference enthalpy with respect to mixture ratio averaged over the mixture range during steady-state operation, Btu/lbm.	

table of hydrodynamic coupling terms versus frequency already resides on file ITAPH in the format used during generation of such a table when INPHYD is input as two. The program will interpolate in this table in order to obtain the hydrodynamic coupling terms each time the frequency changes. The use of interpolation, once a table has been generated, substantially reduces the computer run time for each case run.

The other two control flags input on the third card control the datasets for the combustion dynamics coefficients and the nozzle admittance factors. These flags are similar to INPHYD. Their description is given in Table 2.

Also input on the third card of the main control section input are the file numbers of the datasets discussed above and print control flags for the various forms of output one can obtain. These are all self-explanatory. The user need only refer to Table 2 in order to determine the values that should be input for the case being considered.

The final entry on the third card is for the variable NXP, the number of points to be used for the axial distance and area arrays. This controls the step size that will be taken during the integration of the chamber dynamics equations; i.e.,  $\text{step size} = (XN\emptyset Z - X0)/(NXP-1)$  where  $X0$  is the axial location of the start of vaporization plane and  $XN\emptyset Z$  is the axial location of the nozzle inlet plane. The values of  $X0$  and  $XN\emptyset Z$  are both input on the very next card read in (card 4 in Table 2).

The start of vaporization plane ( $X0$ ) is calculated by plotting the percent unburned of both fuel and oxidizer that is calculated by the DER program (or equivalent steady-state combustion model) as a function of distance from the injector face (Fig. 14). These plots are then extrapolated back to 100% unburned and the axial location of this point is  $X0$ .

Also input on card 4 are the radius of the combustion chamber at the injector face, RINJ, the ratio of specific heats ( $C_p/C_v$ ), GAM $\emptyset$ , the sonic velocity,  $C\emptyset$ , and the oscillatory non-dimensional pressure amplitude desired at the injector face ( $\Delta P/P$ ), DELP. The variables GAM $\emptyset$  and  $C\emptyset$  should be evaluated at the overall

TEST #12    FUEL  
                   OXIDIZER

$\tau_{x_{ox}} = 2.318 \text{ IN. (0.059 m)}$   
 $\tau_{x_{fuel}} = 1.952 \text{ IN. (0.049 m)}$   
 $\tau_{ox} = 0.00669 \text{ SEC}$   
 $\tau_{fuel} = 0.00463 \text{ SEC}$

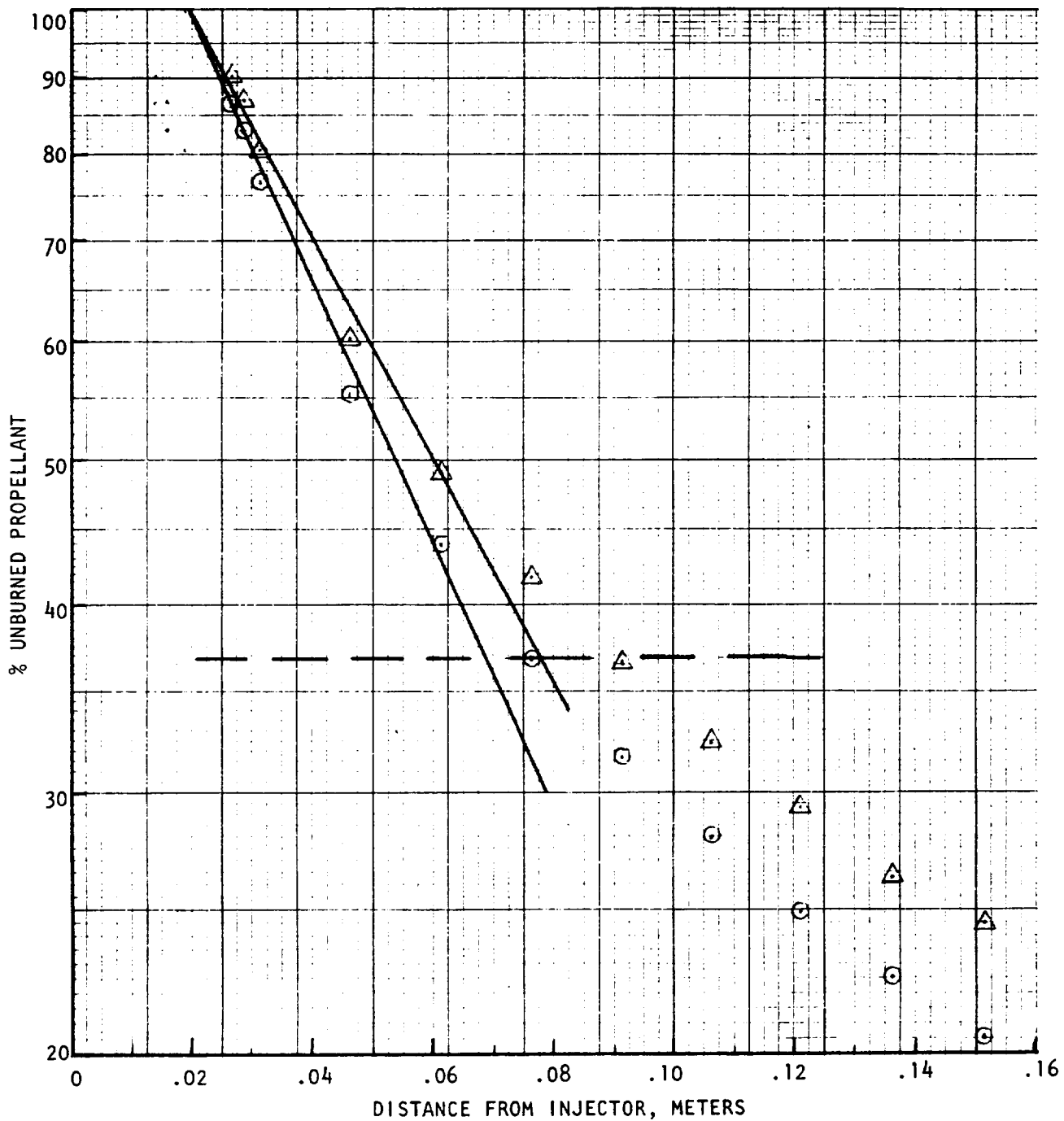


Figure 14. Percent Unburned Propellant as a Function of Distance from Injector Face

mixture ratio. The variable DELP will scale the amplitude of the oscillatory waves solved for in subroutine CHAMDY. A value of 0.1 is recommended.

The first variable input on the fifth card of the MAIN program control section input is NROOT. The absolute value of this variable controls the maximum number of frequency solutions the program will try to find. The program begins its search at the frequency implied by the input variable OMEGAR. It will stop looking once it has found |NROOT| solutions or if the frequency is above FRQMAX. The variables IWRT and IWSKP are the next two variables input on this card. They control the amount of intermediate output one desires. Their exact function is described in the Program Operation Section of this manual.

The last three variables input on the fifth card are KNTMX, KNTRMX, and KNTSMX. Their meaning is explained in Table 2. Recommended values for these variables are 50, 100, and 20 respectively.

The sixth card of the MAIN program control section input contains the variables OMEGAR, OMEGAI, FRQMAX, DELFRQ, DELMX, and CTEST. The first two of these variables specify the starting guess in the  $\omega$  plane for solution. No solutions will be sought below the frequency implied by OMEGAR. Note that OMEGAR and OMEGAI can be input as the frequency in Hertz times  $2\pi$  and the growth coefficient or the frequency in Hertz and the decrement depending upon whether NROOT was input as positive or negative. The variable FRQMAX, as mentioned earlier, is the maximum frequency allowed for the search algorithm to find solutions. The variable DELFRQ specifies the "stepsize" used by the search algorithm. Since there are sometimes many areas in the  $\omega$  plane which contain solutions, a fairly small stepsize is recommended, e.g., 5 Hz. The variable DELMX controls the maximum allowable change in the growth coefficient during successive iterations to minimize the error in the nozzle admittance boundary condition as a function of the imaginary part of  $\omega$ . A recommended value for this variable is  $50 \text{ sec}^{-1}$ . The last variable on this card, CTEST, is the upper bound on the condition number of the transposed Jacobian used to solve the nozzle admittance boundary condition. If the calculated condition number exceeds CTEST, then the search algorithm assumes that there is a singularity near the current value of  $\omega$  and hence, does not proceed further in that area to try and find a solution. A value of 50 to 80 is recommended.

The seventh card of the MAIN Program Control Section input contains error tolerances used in solving the nozzle admittance boundary condition. The first two, EPSF and EPSX, are used during the search algorithm and should be fairly large, e.g. 0.01 to 0.05 (1% to 5% error). The last two, EPSFS and EPSXS, control the final stages of iteration and should be fairly tight, e.g. 0.0005 (0.05%).

The eighth card of this control section contains the variables PC, MBØXI, and MBFUI. The first is the steady state chamber pressure, in PSIA, and the next two are the oxidizer and fuel injection flowrates, respectively (lbm/sec).

The last card, card number 9, in the MAIN Program Control Section input should be input only if the variable INPNØZ is less than or equal to three. If this is the case, the program needs to know the size and range of the frequency table it will use to generate tables for linear interpolation as discussed in the section describing the input variables INPHYD, INPCØM, and INPNØZ. The input variables on this ninth card are NFREQT, FREQMI, and FREQMX. Their meaning is described in Table 2.

The next control section to read data after the MAIN program is the Nozzle Admittance Program. The data for this control section should be input only if  $INPNØZ \leq 3$ . Otherwise, the information is not needed since the nozzle admittance information will be on tape ITAPN. Even when  $INPNØZ \leq 3$ , there is only one card input. This card contains information describing the nozzle geometry. Refer to Table 2 to determine the meaning of the variables on this card. Figure 15 shows exactly what portion of the nozzle each variable is applicable to.

The next control section that requires data is the Hydrodynamics Program section. This control section uses namelist input. The data for this control section are only input if  $INPHYD \leq 2$ . Otherwise, the hydrodynamics information will be on tape ITAPH. The Hydrodynamic Input section (page 102) describes the meaning of the variables to be input for the control section.

The last control section to require input data is the Combustion Dynamics Program. This input is contained on five cards. It should be omitted if INPCØM is greater than or equal to three, since then the combustion dynamics information will reside on tape ITAPC.

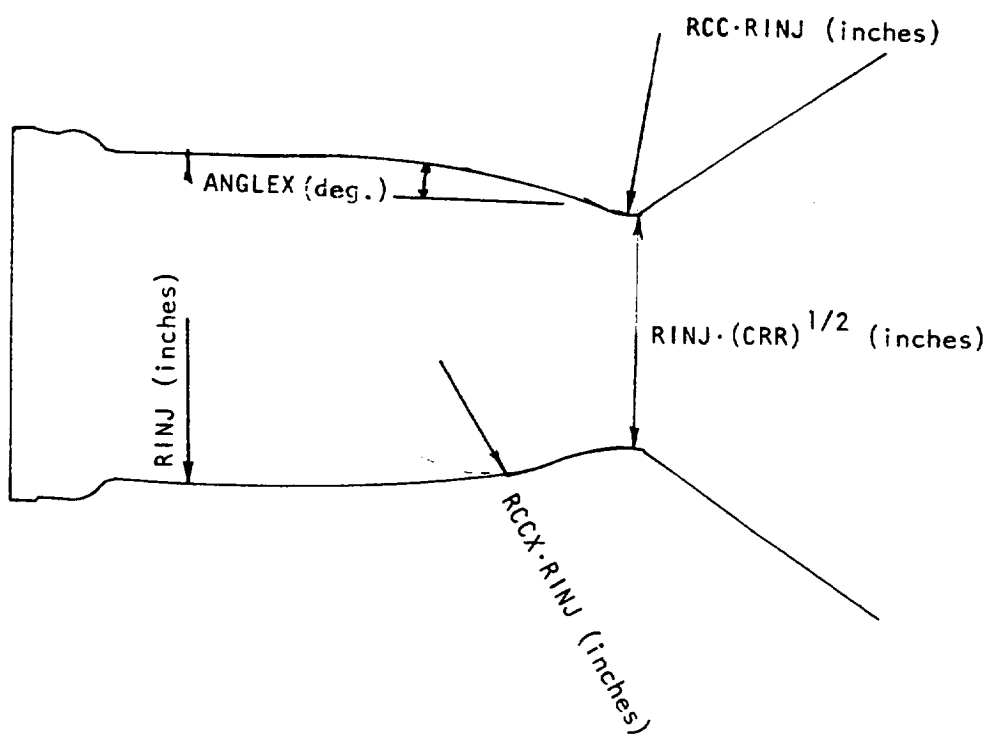


Figure 15. Chamber and Nozzle Geometry

The first two cards for this section contain information specific to the oxidizer; the next two cards contain information specific to the fuel; the last card contains information specific to the combustion gas. Much of the information required in this section is obtainable from the output of the DER program (Ref. 11). This program solves for the steady state behavior of the chamber under consideration. The single stream tube option of the DER program is sufficient for this application.

As mentioned above, the variables input on the first two cards of the Combustion Dynamics Program control section are applicable to the oxidizer. These variables are XKØX, TAUBØX, VBØX, DELHØX, TDRAGØ, ADVØX, ADDØX, DELVØX, NUBØX, DTØXDM, and XIMPØX. The first variable XKØX, is the Klystron constant for the oxidizer jet. This variable controls the distance downstream from the injector face that the Klystron effect will be allowed to occur. The exact method of calculating XKØX has not been determined but it is recommended that a distance corresponding to approximately 45% of the oxidizer vaporized be used.

The next variable, TAUBØX, is the steady state oxidizer vaporization time delay. Figure 14 gives the percent unburned for the sample input case of liquid oxidizer and fuel remaining in the chamber as a function of axial distance plotted on semi log graph paper. This output was derived from a DER computer run. The reciprocal of the average slope of the oxidizer curve during the initial burning phases should be taken as the oxidizer vaporization distance delay. For Figure 14, this is 2.318 inches. The time delay is obtained by dividing this by the average injector velocity, which, for the sample case presented in Figure 14 is 28.859 ft/sec. The result is 0.00669 seconds. The next variable, VBØX, is the average oxidizer injection velocity. This is also given by the DER program from injector orifice and steady state  $\Delta P$  considerations.

The variable DELHØX is the pseudo energy of the oxidizer droplets and is discussed at the end of this section. The variable TDRAGØ is the steady-state oxidizer drag time delay and a value of zero is currently recommended. If this variable is different from zero, the computed pressure and velocity profiles calculated in the chamber are unrealistic (see Conclusions and Recommendations in Ref. 1).

ADVØX and ADDØX are the velocity and orifice diameter exponents for the oxidizer atomization process, i.e.,

$$\bar{D}_{ox} \sim V^{ADVØX} D_{orif}^{ADDØX} \quad (216)$$

These variables are obtained from cold flow tests and for like-doublets -0.75 and 0.57 are recommended. It should be noted that these variables are on different input cards. The next input variable is DELVØX which is the steady-state velocity difference between the oxidizer droplets and the combustion gas stream normalized to the sonic velocity. Since the vaporization rate is highest near the injector face, this variable should be evaluated near the injector face. Because the gas and the droplet velocities are approximately equal to each other at this location, a value of 0.01 is recommended based on turbulence levels in a rocket chamber (Ref. 26).

The next variable, NUBØX, is the steady state oxidizer Nusselt number used in the vaporization expression. This variable should be computed from the relation

$$N_{u_{ox}} = 8/6 \frac{\bar{D}_{ox}^2}{k'_{ox} \bar{\tau}_{ox}} \quad (217)$$

where  $\bar{D}_{ox}$  is the average oxidizer droplet diameter near the injector face,  $k'_{ox}$  is the vaporization coefficient used in the  $k'$ -model evaluated at a mixture ratio near the injector face and  $\bar{\tau}_{ox}$  is the oxidizer time delay. Both  $\bar{D}_{ox}$  and  $k'_{ox}$  are obtainable from a DER computer run. For the sample case input, "near the injector face" was taken as the first axial step printed by the DER program.

The variable DTØXDM is the change in the oxidizer vaporization time delay with respect to mixture ratio. At the present time, a value of zero is recommended based on model verification cases (Ref. 1).

The last oxidizer variable input is XIMPØX. This is the impingement point in inches for the oxidizer jets.



The next two cards contain data for the fuel. These data are obtained the same way as they were for the oxidizer and are input in the same order.

The last card for this control section contains combustion gas data. The first variable input on this card, MWG, is the steady state molecular weight evaluated at the overall mixture ratio. This may be obtained from tables of molecular weight versus mixture ratio printed by the DER program. The next variable, CS, is the characteristic velocity evaluated at the overall mixture ratio. This is also obtainable from DER table output. The last three variables on this card are DRGDMR, DCSDMR, and DHDMMR. These are the partial derivatives with respect to mixture ratio of the gas constant, the characteristic velocity, and the reference enthalpy respectively. DRGDMR and DCSDMR can be calculated from equilibrium calculations but a value of zero is recommended for DCSDMR based on model verification cases (Ref. 1).

The variable DHDMMR, and also the variables DELH $\phi$ X and DELHFU, is calculated by curvfitting the steady-state energy equation with stagnation temperature/mixture ratio data calculated by an equilibrium program. The steady-state energy equation can be written as

$$\left( \frac{\gamma_{\phi}}{\gamma_{\phi} - 1} \right) R_{\phi} \left\{ T \left[ 1 + \frac{1}{R_{\phi}} \left( \frac{\partial R}{\partial MR} \right)_{\phi} (MR - MR_{\phi}) \right] (1 + MR) - (1 + MR_{\phi}) T_{\phi} \right\} = (MR - MR_{\phi}) (\Delta h_{ox}) - \left[ MR (1 + MR) - MR_{\phi} (1 + MR_{\phi}) \right] \left( \frac{\partial h}{\partial MR} \right)_{\phi} \quad (218)$$

where

$$(\Delta h_{fu}) = \left( \frac{\gamma_{\phi}}{\gamma_{\phi} - 1} \right) (1 + MR_{\phi}) R_{\phi} T_{\phi} - MR_{\phi} (\Delta h_{ox}) + (MR_{\phi}) (1 + MR_{\phi}) \left( \frac{\partial h}{\partial MR} \right)_{\phi} \quad (219)$$

and the subscript  $\phi$  indicates that the variable is to be evaluated based on the overall mixture ratio.

## HYDRODYNAMIC INPUT

This section describes data needed by the hydrodynamics subroutine, HYDRDY, to simulate the various feed system components. It is assumed that the feed system being modeled has been laid out on the generalized feed system schematic of Fig. 4 with an appropriate segment number assigned to each feed system component (or combination of components).

### Basic Feed-System Data

To describe the basic feed system it is necessary to know the length, area, resistance and wall compliance of each of the numbered segments of Fig. 4 which are being used. Also, the acoustic velocity and density of the fluid in each segment must be known. If there is dissolved or entrained gas in the system, then a preliminary calculation must be made for each feed system section to account for the effect of the gas on the fluid acoustic velocity.

Specific parameters required for the numbered segments are:

- A - Segment cross-sectional area - in.<sup>2</sup>
- CW - Segment wall compliance ( $\Delta V/\Delta P/V$  - in.<sup>2</sup>/lb)
- L - Segment length - in.
- R - Segment linearized hydraulic resistance ( $\Delta P/\dot{W}$ ) - sec/in.<sup>2</sup>
- V - Segment fluid acoustic velocity - in./sec
- RHØL - Segment fluid density - lb/in.<sup>3</sup>

### Valves, Fittings, Orifices, Screens, Flowmeters, etc.

These components can each be described in the model simply as lumped resistance at the end of a line segment. Rather than using all the attributes of one of the numbered segments (length, area, wall compliance, etc.) for one of these "resistance only" components, it is suggested that its resistance merely be added to that of the adjacent upstream pipe segment. The combination can then be entered as one of the numbered segments with the length, area and wall compliance values being primarily those of the pipe segment.

### Accumulators

A feed system accumulator can be represented as one of the side branch lines of the Fig. 4 schematic by specifying an appropriate length, area, acoustic velocity and fluid density for the fluid volume of the accumulator and also specifying an appropriate connecting resistance. The spring rate of the accumulator piston can be specified in terms of the segment wall compliance value.

### Propellant Tanks

A large tank will have the effect of constant fluid pressure at its outlet and can be represented simply as the input to segment number 1, 15, or 22. No descriptive parameters are required for these inputs. Small tanks can be represented as one of the side branch lines in a manner similar to an accumulator. Ullage volume in a small tank is represented by a reduced value for the segment acoustic velocity.

### Cavitating Venturies

The steady-state effect of a cavitating venturi is to have constant flow through the venturi as a function of variations in downstream pressure. For an oscillatory system, the vapor bubble downstream of the venturi throat makes the venturi look like a constant pressure boundary for small amplitude oscillations. To simulate this effect the effective acoustic velocity for the segment downstream of the cavitating venturi should be made very small ( $\approx 10$  inches/sec would be appropriate). The steady-state hydraulic resistance of the cavitating venturi can be lumped with that of the upstream pipe segment as described above for valves, fittings, etc.

### Regeneratively-Cooled Thrust Chamber

Regeneratively-cooled thrust chamber jackets can be represented as one or more of the numbered Fig. 4 segments. Because in most thrust chambers the coolant flow area changes continuously with length, as many segments as possible should be devoted to the jacket so as to improve the simulation

accuracy. The fluid temperature also may change significantly along the chamber length thereby necessitating the use of several segments with different acoustic velocities to achieve accurate simulation.

#### Lines, Ducts, Bends, Bellows, and Flex Lines

These components are described in the model in terms of the basic numbered segment input parameters of length, area, fluid acoustic velocity, fluid density, wall compliance and linearized hydraulic resistance. For a duct or line of constant diameter,  $D$ , wall thickness,  $h$ , and wall material bulk modulus,  $E$ , the program input wall compliance value,  $CW$ , is simply  $D/LE$ . For a bellows or flex line of volume,  $V$ , the wall compliance value,  $CW$ , may be calculated from  $\Delta V/\Delta P/V$  where  $\Delta V/\Delta P$  is the volume change per psi at the operating pressure.

#### Injectors

The hydrodynamics subprogram employs a separate set of equations to describe the hydrodynamic characteristics of the two injectors in the Fig. 4 generalized feed system schematic. The specific input parameters for the two injectors are the volume, linearized hydraulic resistance, orifice inertance ( $1/Ag$ ), fluid acoustic velocity, and a structural parameter defining the change in injector volume per psi of injector  $\Delta P$ . In terms of the program variable names the required injector parameters are:

- RF - Linearized hydraulic resistance for the "F" injector,  $(2 \Delta P/\dot{W}) - \text{sec}/\text{in.}^2$
- R $\emptyset$  - Linearized hydraulic resistance for the " $\emptyset$ " injector,  $(2 \Delta P/\dot{W}) - \text{sec}/\text{in.}^2$
- VOLF - Volume of the "F" injector -  $\text{in.}^3$
- VOL $\emptyset$  - Volume of the " $\emptyset$ " injector -  $\text{in.}^3$
- VF - Fluid acoustic velocity of the "F" injector -  $\text{in.}/\text{sec}$
- V $\emptyset$  - Fluid acoustic velocity of the " $\emptyset$ " injector -  $\text{in.}/\text{sec}$
- ZF - Inertance of the "F" injector orifices ( $1/Ag$ ) -  $\text{sec}^2/\text{in.}^2$
- Z $\emptyset$  - Inertance of the " $\emptyset$ " injector orifices, ( $1/Ag$ ) -  $\text{sec}^2/\text{in.}^2$
- KF - "F" injector deflection constant,  $(\Delta V/\Delta P) - \text{in.}^5/\text{lb}$
- K $\emptyset$  - " $\emptyset$ " injector deflection constant,  $(\Delta V/\Delta P) - \text{in.}^5/\text{lb}$

### Tees, Splitters and Capped Lines

No provision is made in the model for completely generalized input of tees and branched lines. However, a system of considerable complexity can be modeled by laying out an appropriate flow path on the generalized Fig. 4 schematic. For example, a feed system with up to seven side branch lines can be simulated by choosing the main flow path through segments 1, 3, 4, 14, 17, 21, 25, 26, 28, 29, 30 and the "F" injector in series.

### Input Variables

Data input to the hydrodynamics subprogram is from three sources: (1) Via the argument list in the CALL HYDRDY statement, (2) Through labeled common block/CØNTAP/and (3) By use of the NAMELIST data read routine.

The argument list variables, in order, are:

- IR - Data read flag - dimensionless
- INPHYD- Program function flag - dimensionless
- FRE - Single frequency for feed system frequency response calculation - Hz
- GINØ - Output value of oscillatory oxidizer injector flowrate for input frequency FRE - dimensionless
- GINF - Output value of oscillatory fuel injector flowrate for input frequency FRE - dimensionless
- PCIN - Injector end thrust chamber pressure - psia
- WØIN - Steady-state oxidizer injector flowrate - lb/sec
- WFIN - Steady-state fuel injector flowrate - lb/sec

Several HYDRDY input variables are transmitted via labeled common block/CØNTAP/. ITAPH is the logical unit number of the output device on which sub-routine HYDRDY tabulates output values of oscillatory injection flowrates for the specific frequencies (up to 100 separate values) given in the array FREQT in common block/CONTAP/. The value of the FREQT in common block/CØNTAP/ is the total number of frequencies stored in the array FREQT.

All other data required by HYDRDY, including all the feed system descriptive data, is read in by use of the NAMELIST routine. The local rules for using this routine should be checked to verify that the correct card or terminal format is being used. Table 3 shows a list of allowable FORTRAN names, the maximum values of subscripts, and a definition of the names. The name of the NAMELIST block is HYD.

TABLE 3. NAME LIST/HYD/DATA INPUT NAMES

<u>NAME</u>	<u>DEFINITION</u>
A(30)	Array containing segment cross-sectional area values
CW(30)	Array containing segment wall compliance values
FREQ	Frequency at which HYDRDY will compute feed system frequency response if INPHYD $\leq$ 1
FREQT(100)	Array containing frequencies at which HYDRDY will compute feed system frequency response of INPHYD > 1.
ICRT	ICRT = 1; injector flowrate gain and phase will be plotted vs frequency. ICRT = 0; no plot (default)
ID	Dummy name to allow for data card sequence numbers
IH(126)	Array containing control flags used by subroutine FREQD. Can be used to obtain printouts and plots of feed system frequency response for other variables in addition to the injector flowrates.
IRFLAG	IRFLAG = 0; read data from unit 5 (default) IRFLAG $\neq$ 0; read data from unit ITERM
ITERM	ITERM = 0; no terminal data input (default) ITERM > 0; read data from terminal (unit ITERM)
ITYPE	ITYPE = 1; both oxidizer and fuel feed systems are modeled simultaneously (default) ITYPE = 2; oxidizer feed system modeled on first pass of frequency response routine; fuel feed system modeled on second pass.
IWRITE	IWRITE = 0; HYDRDY input printed on unit 6 (default) IWRITE = 2; extensive printout of all HYDRDY input, intermediate output and final output on unit 6 IWRITE = 1; printout of HYDRDY input and final output on unit 6
KF	Injector face flexibility constant for "F" injector
K $\emptyset$	Injector face flexibility constant for " $\emptyset$ " injector.
L(30)	Array containing segment length values

TABLE 3. (Concluded)

<u>NAME</u>	<u>DEFINITION</u>
$\emptyset$ MI	Lowest frequency for injector flowrate gain/phase plot; default to FREQT(1) if not entered
$\emptyset$ MFL	Highest frequency for injector flowrate gain/phase plot; default to FREQT(NFREQT) if not entered
R(30)	Array containing segment hydraulic resistance values
RF	Hydraulic resistance for "F" injector
RH $\emptyset$ L(30)	Array containing segment propellant density values
R $\emptyset$	Hydraulic resistance for " $\emptyset$ " injector
V(30)	Array containing segment acoustic velocity values
VF	Acoustic velocity for "F" injector
V $\emptyset$	Acoustic velocity for " $\emptyset$ " injector
V $\emptyset$ LF	Volume of "F" injector
V $\emptyset$ L $\emptyset$	Volume of " $\emptyset$ " injector
ZF	Inertance of "F" injector
Z $\emptyset$	Inertance of " $\emptyset$ " injector

Required input in the NAMELIST/HYD/data is a value of A, CW, L, R, RH $\emptyset$ L, and V (see Table 3 for descriptions) for each numbered segment being included in the feed system. Values of KF and/or K $\emptyset$ , RF and/or R $\emptyset$ , VF and/or V $\emptyset$ , V $\emptyset$ LF and/or V $\emptyset$ L $\emptyset$  and ZF and/or Z $\emptyset$  are also required. It should be noted that, when possible, both oxidizer and fuel feed systems should simultaneously be laid out on the Fig. 4 schematic with the injector labeled " $\emptyset$ " being used for the oxidizer side (data values K $\emptyset$ , R $\emptyset$ , V $\emptyset$ , V $\emptyset$ L $\emptyset$  and Z $\emptyset$ ) and the injector labeled "F" being used for the fuel side (data values, KF, RF, VF, V $\emptyset$ LF and ZF). If this can be done, a single call to subroutine FRESP will generate frequency response data for both oxidizer and fuel feed systems. If feed system complexity requires that the oxidizer and fuel feed systems be laid out separately on the Fig. 4 schematic, then two sets of input data must be read and subroutine HYDRDY must call FRESP twice - first for the oxidizer feed system calculations and second for the fuel feed system calculations. To specify this option, variable ITYPE must be set equal to 2.

Variable INPHYD in the HYDRDY argument list controls the HYDRDY calculation process. If  $INPHYD \leq 1$  the oscillation injection flowrates are calculated for a single frequency, specified by variable FRE in the HYDRDY argument list. If  $INPHYD > 1$ , HYDRDY calculates oscillatory injector flowrates for the number of frequencies, NFREQT, which are contained in array FREQT. Both NFREQT and the array FREQT are stored in labeled common block/C $\emptyset$ MTAP/ prior to calling HYDRDY.

NAMELIST variable ICRT controls the option for generating CRT plots of the oscillatory injection flowrate gains and phase values as a function of frequency. If  $ICRT = 0$  (the default value) no plots are made. If  $ICRT \geq 1$  plot output is written to the output file named SYSCRT.

NAMELIST variable, ID, is a dummy name which can be used on each input card to provide an identification number field without violating the NAMELIST restriction that the entire card is read as data. For example, ID = 00000010 could be in columns 70-80 of a HYDRDY data card and ID = 00000020 in columns 70-80 of the next card. The NAMELIST routine would then interpret each card's sequence number as a new value for the dummy variable, ID. The value of ID is not used in any way by subroutine HYDRDY.



NAMelist variable, ITYPE, is used to indicate to HYDRDY the format of the feed system modeling. If ITYPE = 1 (the default value) it is assumed that both oxidizer and fuel feed systems are modeled simultaneously with only one set of HYDRDY input values (for the 30 segments and 2 injectors of the Fig. 4 schematic). If ITYPE = 2, HYDRDY will send two consecutive sets of input data; the first set will be assigned to the oxidizer feed system and the second set to the fuel feed system. For either value of ITYPE the program will assume that the injector labeled "Ø" on Fig. 4 is the oxidizer flow outlet and the injector labeled "F" is the fuel flow outlet. Therefore, this convention must be followed when laying out the feed system model.

NAMelist variables IRFLAG and ITERM are optional HYDRDY inputs which indicate that data input will be provided from a timesharing terminal. If IRFLAG = 0 (the default value), input data will be read only from FORTRAN logic unit number, ITERM. It should be noted that the default values for IRFLAG and ITERM are set up so that the initial data input will always be card input on unit 5.

After reading the initial NAMelist data on unit 5, HYDRDY checks the value of ITERM and, if non-zero, proceeds to read additional first case data from the terminal on unit ITERM. Thus, the first case card NAMelist input could consist of the single item ITERM = N, where N is the terminal logical unit number. If terminal input only is desired, block data program/F/ can be recompiled with the IRFLAG default value changed from 0 to 1 and the ITERM default value changed from 0 to the desired unit number.

NAMelist variable IWRITE (main program control variable IPRHYD) controls the printed output from HYDRDY. If IWRITE = 0 (the default value), only the NAMelist input to HYDRDY is printed on logical unit 6. If IWRITE < 0 the NAMelist input is printed and the normal HYDRDY output is printed as well as being saved on an output device in binary form. If IWRITE = 1 both HYDRDY input and normal output are printed. If IWRITE > 1 extensive print-outs of subroutine FRESP intermediate calculations are printed in addition to the normal HYDRDY input and output.

## PROGRAM OUTPUT

The output of the FSCSM computer program is provided as the usual tabular printout. A sample case is included in Appendix E which corresponds to the input dataset listed in Appendix D. As is also mentioned in the Program Operation section of this manual, the input case listed in Appendix D consists of two cases being run back to back. The output from the first case is given in Appendix E from pages E-2 through E-14. The first page of output consists of a title page identifying the current version of the FSCSM computer program. The input data are printed out as they are read in. This permits both a full documentation of the computer run conditions for later analysis as well as a convenient method to check for input errors if unusual results are calculated. Page E-3 of the listing in Appendix E gives the two alphanumeric cards identifying the case at the top of the page right under the program title. Subsequent to these two cards, the information on the cards read in by the main control section and the nozzle admittance control section are printed out. After reading and writing these cards, and since  $INPNØZ = 3$ , the program proceeds to the nozzle admittance table calculations. Information pertinent to these calculations is printed on page E-4. The frequency table goes from 150 Hz to 400 Hz as specified by the input variables  $FREQMI$  and  $FREQMX$ .

Since  $INPHYD = 2$ , the program proceeds to the hydrodynamic subroutines right after the nozzle admittance calculations. Input for this routine is in the form of NAMELIST data. The NAMELIST is output on pages E-5 and E-6. A printout of the feed system response table computed by subroutine  $HYDRDY$  and saved on file  $ITAPH$  is given on page E-7.

The next set of input required is used in subroutine  $CØMBDY$  and  $STEADY$ . This is output on page E-8. The steady-state profiles are then computed and printed on pages E-9 and E-10.

The program then begins its search for solutions to the nozzle admittance boundary condition. The first one it finds is at 210.42 Hz. The program then outputs the combustion dynamic coefficients, the frequency and decrement, and the feed system response for this solution on page E-14. On page E-12, the oscillatory profiles correspond to this solution are given. The program then proceeds to the next case.

Since the second case does not generate the data on files ITAPN and ITAPH (it only reads this information), these tables are not printed. The first page of output, page E- 15, in Appendix E, consists entirely of the data read by the Main Control Section and the Combustion Dynamics Control Section. Since the STEADY Control Section print code is zero (IPRSTE = 0), the program skips over the steady-state output (although of course, it still computes it) and proceeds directly to the section which solves the nozzle admittance boundary condition. The first root it finds above the input frequency of 265 Hz (given by the variable  $\omega$ MEGAR), is at 280.62 Hz. It prints out the frequency, decrement, nozzle admittances, and feed system response for this solution. Output of the combustion coefficients and oscillatory profiles is bypassed because the input flags IPRC $\omega$ M and IPRCHM were set to zero.

The final page of output is the title page. This indicates normal termination of the job.

## PROGRAM OPERATION

The FSCSM computer program is designed to read in an input case sequentially, perform the calculations for that case, and output the results. The program then transfers back to its beginning to read in the next case. In this manner, running jobs back-to-back is quite straightforward. The sample input case listed in Appendix D provides an example of two such cases run back to back. The first case, given by the first 24 cards, is run with no prior information residing on the hydrodynamic feed system, the combustion dynamics, or the nozzle admittance datasets. Since  $INPHYD = 2$  and  $INPNØZ = 3$ , this case generates tables of the hydrodynamic feed system response and nozzle admittance versus frequency and saves them on files ITAPH and ITAPN, respectively. The subsequent case (the last 13 cards in Appendix D) will use the information stored on these datasets. Although these two cases were run back to back, this was by no means necessary. The second case is self-contained and could be submitted separately. Of course, if this were the situation, the user must be sure there are datasets on files ITAPH and ITAPN which are applicable to that second case.

For the sample dataset run, the two input cases found solutions to the nozzle admittance boundary equation at 210.42 Hz and 280.62 Hz, respectively. If there are no other solutions between these two frequencies, the same effect could have been obtained by setting the input variable  $NRØØT$  equal to -2 for the first case instead of -1. The program would have then looked for the first two roots above the input frequency 190 Hz ( $ØMEGAR$ ) and found both solutions automatically. The second case would not be input for this situation.

### Program Size, Overlay Structure, and Timing

Without overlay, the FSCSM computer program load module requires 262.4 K Bytes of computer storage on the IBM 370 Model 165 computer. This storage does not count the buffers needed for input/output. If one allocates a 1 K Byte of buffer size for each of the three data sets used to store the feed system, combustion dynamic, and nozzle admittance data (which are all unformatted input/output), uses two buffers for each data set, and adds in the buffer requirements for his card input, printed output, and CRT output, then the total buffer space should be well under 10 K bytes on a 370/165 computer. With the overlay structure specified in Fig. 16, the total program requirement is 220 K bytes of storage on an IBM 370/165 computer, including two buffers for each of the three unformatted datasets at 1 K bytes each.

Computer run time has only been checked for an IBM 370/165 computer where the subroutines were compiled using the IBM procedure AFØRTRAN with the optimizing parameter, ØPT, equal to one. For this situation, each iteration during the search algorithm portion of the program (when ISCNT equals one or four) averaged 3.7 CPU seconds. When ISCNT=5, each iteration is about twice as fast. For the cases run during model verification a five Hz step size for the search algorithm was used (DELFRQ=5). For these cases, each solution to the nozzle admittance boundary equations averaged 0.85 minutes of CPU.

### Program Input/Output Dataset File Information

The case input dataset file number used by the FSCSM computer program is 5. The printed output dataset file number is 6. There are three auxiliary files used by the program. These are specified by the input parameters, ITAPC, ITAPH, and ITAPN, corresponding to the combustion dynamics, hydrodynamic feed system, and nozzle admittance datasets. Control of the reading from or writing on to these respective datasets is specified by the three input flags INPCØM, INPHYD, and INPNØZ. The

program uses unformatted input/output statements for transmitting information to and from these datasets. A convenient blocksize to use is 1K bytes.

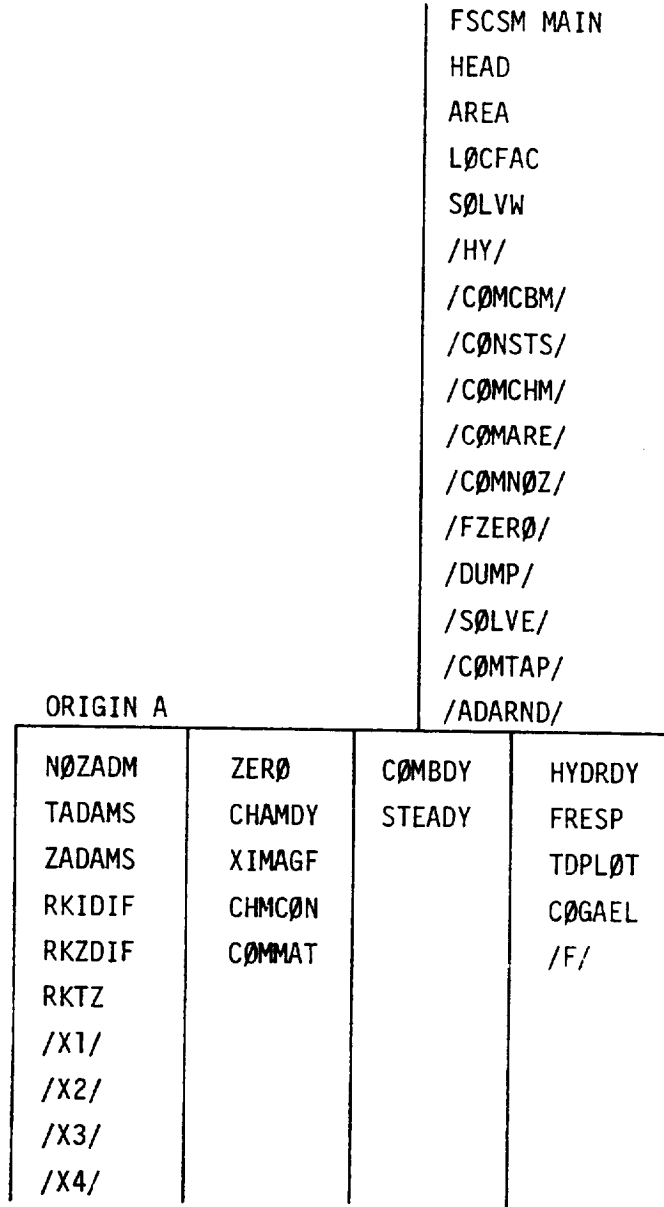


Figure 16. FSCSM Program Overlay Structure

## Diagnostics

The Feed System Coupled Stability Model computer program has been designed to operate as straightforward as possible with a minimum amount of user interaction for each case being run. There may be times however, when the program's results appear questionable or the algorithm used to find solutions in the frequency space to the nozzle admittance boundary equation runs into difficulty or does not find solutions that were expected. Many diagnostic messages are coded into the program to warn the user of such problems. Also, there are certain dump codes which enable the user to obtain intermediate output in order to debug most problems that may arise.

One of these dump codes is the variable IWSKP. When this is set to zero, no intermediate output is obtained. When it equals one, a certain amount of limited output will be generated. This output comes in two forms depending upon whether or not the program is within its search algorithm portion or its two-dimensional secant portion. For the first case, the variable ISCNT has the value of one or four. In the second case it has the value 5. When ISCNT equals one or four and IWSKP equals one, subroutine SOLVW will print the following variables in the order given: the iteration counter (KNTR), the control flag (ISCNT), the counter (KSCNT4), the current values of omega ( $\omega$ ), the upstream and downstream nozzle admittances (CNØZA and NØZA), the absolute value of the error in the nozzle admittance equation (HN), the value of the test function (FTST2), and the determinant and condition number of the transposed Jacobian (DET2 and CØND2). This printing will be performed every time the real part of  $\omega$  is incremented by  $2\pi \cdot \text{DELFRQ}$  right after the imaginary part of  $\omega$  has been chosen to minimize  $|F|$ , the absolute value of the error in the nozzle admittance boundary equation. The user can employ this output to determine if there is a region in the  $\omega$ -plane where a possible solution may have existed (e.g., the error became small but the test function did not change sign). He can then rerun his case while taking smaller frequency steps through the narrowed range where he suspects a solution may exist. Also, the program may jump over a solution if there is a singularity within DELFRQ of that solution. If this is the case, the program will sense the singularity and not proceed any further in its search in that range. Rerunning the case with a smaller value of DELFRQ will solve this problem.

When ISCNT equals five and IWSKP equals one, subroutine SØLVW prints after each two-dimensional secant method, these variables in the following order: ISCNT, KNTS, ØMEGA, CNØZA, NØZA, and HN. Although it did not happen for any of the cases performed during the checkout of the computer model, the two-dimensional secant method may diverge. The above computer output would be useful in determining the cause of the problem.

When the variable IWSKP equals two, all the above output is printed plus the following:

1. When ISCNT equals one or four, intermediate output is obtained during the iterations to minimize  $|FN|$  with respect to  $\text{Imag}(\omega)$ . For this case, one obtains the variables KNT, IER, X1, X2, F1, F2, ØMEGA, FN, GN, and HN. These are all described in Appendix A. This output may be useful in seeing how the error is changing as a function of the decrement when the real part of the frequency is held fixed. Further, when ISCNT equals one or four, the variables FN, DFRDX, DFIDX, DFRDY, and DFIDY are printed along with the output obtained when IWSKP = 1.
2. When ISCNT equals five and IWSKP equals two, one obtains the output for the case IWSKP equals one for the two-dimensional secant method plus the following variables in order: XR1, XI1, FR1, FI1, XR2, XI2, FR2, FI2, XR3, XI3, FR3, FI3, XR4, XI4, FR4, FI4, and FN. These variables correspond to the current values of  $\omega$  and FN being used by the two-dimensional secant method. They can be used to trace which points the algorithm is replacing as the iteration proceeds as well as how the error is behaving.

Another input variable which controls intermediate output is the FØRTRAN variable IWRT. This variable is input as zero, no intermediate output is obtained. If this variable is input as a positive number, then intermediate output from subroutine CHAMDY is obtained. This output consists of the oscillatory profiles for the variables P (pressure), RHØ (density), MR (mixture



ratio), and T (temperature) along with the current value of the complex frequency,  $\omega$ . This output is printed everytime subroutine CHAMDY is entered.

The diagnostic messages that are coded within the FSCSM computer program may be printed for several reasons.

Within subroutine SØLVW, there are three diagnostic messages coded which will appear when certain iteration counters are exceeded. The first is

```
*****  
WARNING, POSSIBLE ROOT IN FREQUENCY RANGE: --  
*****
```

When this message appears, it means that a potential root was bracketed but the error did not decrease sufficiently within ten additional iterations to warrant the program proceeding further with its search in that range. Moreover, the determinant did not change sign and the condition number remained less than CTEST in that range. Rerunning the case over the specified frequency range given in the message with IWSKP equal to one or two may prove beneficial if the user suspects there may be an actual solution in that range.

The second diagnostic message printed by subroutine SØLVW is

```
**** UNABLE TO FIND ROOT FOR IMAG PART OF F ****
```

Along with this message, the variables X1, F1, K2, F2, X3, F3, ANS, FANS, KNT, IER, and ØMEGA are printed in the order listed. When this message appears, it means the algorithm to minimize  $|FN|$  with respect to  $\text{Imag}(\omega)$  has failed. If this message appears, it usually means something is wrong with the input parameters. The only occurrence that the programmers are aware of when this is not the case is when the error attains a minimum as  $|\text{Imag}(\omega)| \rightarrow \infty$ . Since this happens only in the most extraordinary situations, the procedure should be to rerun the case and not include the frequency range where that anomaly is occurring.

The third diagnostic produced by subroutine SØLVW is

\*\*\*\* EXCEEDED CONVERGENCE LIMIT \*\*\*\*

Along with this message, the variables IER, KNTS, KNTR, ISCNT, XRI, XI1, FR1, FI1, ....., XR4, XI4, FR4, FI4, are printed.

This message will appear if KNTR is greater than KNTRMX or KNTS is greater than KNTSMX. In the former case, the usual error is that the user input too small a DELFRQ to cover the range between solutions to the nozzle admittance boundary equation in KNTRMX steps or too small a KNTRMX to allow that range to be covered in steps of length DELFRQ.

In the case where KNTS is greater than KNTSMX, it would probably mean that the two-dimensional secant method is diverging. The job should be rerun with IWSKP equal to one or two to obtain more information concerning the problem.

There is also a diagnostic message printed from subroutine CØMMAT. This is the subroutine that solves the four by four system of linear equations for subroutine CHAMDY. If any of the diagonal elements of the associated matrix are zero, then the message

\*\*\*\*\* DIVIDE CHECK IN CØMMAT \*\*\*\*\*

will appear along with a printout of the row number of the zero diagonal as well as the complex matrix being solved. If this error message appears, then there must be something very wrong with the case being run, e.g, the input data is in error, or a dimension has been exceeded. One should recheck his input carefully and then, if necessary, rerun the case with IWRT equal to one and IWSKP equal to one or two.

Two similar messages as the one above are printed by subroutine CØGAEL. The first of these messages is

\*\*\*\*ERROR IN CØGAEL SUBROUTINE, J AND JMAX EQUAL, RESPECTIVELY\*\*\*\*

and the second is

\*\*\*\*MATRIX IS SINGULAR, EXIT FROM CØGAEL. THE PIVOT ELEMENTS ARE...\*\*\*\*

The reasons for these errors are similar to the CØMMAT error message.

#### Conversion to UNIVAC

The following cards must be changed to execute the FSCSM program on a UNIVAC computer (see Appendix C for code listing):

1. Change CØMPLEX\*16...to CØMPLEX....

<u>Routine</u>	<u>Card Number</u>
CHAMDY	80
CØMMAT	150

2. In subroutine CØMMAT, change CDABS to CABS on card No. 24.
3. In subroutine TDPLØT, replace card 11310 with  
31 CØNTINUE.....0011310
4. In subroutine NØZADM, replace card 1730 with  
8 NØZA = CMPLX(SYR, SYI)...00001730
5. In the main program, change ATAND( ) to 57.296\*ATAN( )  
on card numbers 3370, 3410, 3450, and 3490.



## REFERENCES

1. Kahn, D. R. et al., Orbital Maneuvering Engine Feed System Coupled Stability Investigation, Final Report, R-9807, NAS9-14315, September 1975.
2. Ezekiel, F. D., Control Engineering, 111-112, May 1958.
3. Summerfield, M., "A Theory of Unstable Combustion in Liquid Propellant Rocket Systems," ARS Journal, Vol. 21, No. 5, 108-114, September 1951.
4. Crocco, Luigi, and Sin-I Cheng, "Theory of Combustion Instability in Liquid Propellant Rocket Motors," AGARDograph No. 8, Butterworths Sci. Publ., 1956.
5. Hurrell, H. G., "Analysis of Injection Velocity Effects on Rocket Motor Dynamics and Stability," NASA TR R-43, 1959.
6. Wenzel, L. M. and J. R. Szuch, "Analysis of Chugging in Liquid-Bipropellant Rocket Engines Using Propellants With Different Vaporization Rates," NASA TN D-3080, October 1965.
7. Zinn, B. T. and E. A. Powell, "Nonlinear Combustion Instability in Liquid-Propellant Rocket Engines," Thirteenth Symposium (International) on Combustion, 491-503, August 1970.
8. Szuch, J. R., "Digital Computer Program for Analysis of Chugging Instabilities," NASA TN D-7026, December 1970.
9. Lores, M. E. and B. T. Zinn, "Nonlinear Longitudinal Combustion Instability in Rocket Motors," Combustion Science and Technology, Vol. 7, 245-256, 1973.
10. Priem, R. J. and M. F. Herdmann, "Propellant Vaporization as a Design Criterion for Rocket-Engine Combustion Chambers," NASA TR R-67, 1960.
11. Combs, L. P., "Liquid Rocket Performance Computer Model With Distributed Energy Release," NASA CR-114462, June 1972.
12. Zajac, L. J., "Correlation of Spray Droplet Distribution and Injector Variables," R-8455, Rockwell International, Rocketdyne Division, Canoga Park, California, 1969.
13. Nurick, W. H., "Study of Spray Disintegration in Accelerating Flow Fields," NASA CR-114479, June 1972.

R-9808/121

**PRECEDING PAGE BLANK NOT FILMED**

14. Zajac, L. J., "Droplet Breakup in Accelerating Gas Flows, Part I - Primary Atomization, Part II - Secondary Atomization," NASA CR-134478 (Part I), NASA CR-134479 (Part II), October 1973.
15. Fenwick, J. R. and G. J. Bugler, "Oscillatory Flame-Front Flowrate Amplification Through Propellant Injection Ballistics (The Klystron Effect)," ICRPG 3rd Combustion Conference, CPIA No. 138, Vol. 1, 417-427, October 1966.
16. Goldsmith, M. and S. S. Penner, Jet Propulsion, Vol. 24, p. 245, July-August 1954.
17. Williams, F. A., Combustion Theory, Addison Wesley, Reading, Mass., 1965.
18. Combs, L. P., W. D. Chadwich, and D. T. Campbell, "Liquid Rocket Performance Computer Model With Distributed Energy Release," Interim Final Report, NASA CR-111000, September 1970.
19. Ranz, W. E. and W. R. Marshall, Jr., Chem. Eng. Progress, Vol. 48, 141-146 and 173-180, 1952.
20. Herdmann, M. F., "Amplification by Wave Distortion of the Dynamic Response of Vaporization Limited Combustion," NASA TN D-6287, May 1971.
21. Kenworthy, M. J., "Augmentor Combustion Stability Investigation," AFAPL-TR-74-61, August 1974.
22. Wilkenson, J. H., The Algebraic Eigenvalue Problem, Oxford Univ. Press, Ely House, London, 1967.
23. Bell, W. A. and B. T. Zinn, "The Prediction of Three-Dimensional Liquid Propellant Rocket Nozzle Admittances," NASA CR-121129.
24. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill Book Co., New York, 1965.
25. Muller, D. E., "A Method for Solving Algebraic Equations Using a Computer," MTAC, Vol. 10, pp. 208-215, 1956.
26. Priem, R. J. and D. C. Guentert, Combustion Instability Limits Determined by a Nonlinear Theory and a One-Dimensional Model, NASA TN D-1409, 1962.

APPENDIX A

FORTRAN SYMBOL TABLE

<u>FORTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
A(100)	A	R	CØMARE	Axial distance array, $X(I) = X_0 + (I-1) \cdot \text{DELX}$ , m
ADDFU	$a_{fu}$	R	CØMCBM	Fuel liquid jet diameter exponent, unitless
ADDØX	$a_{\phi x}$	R	CØMCBM	Oxidizer liquid jet diameter exponent, unitless
ADVFU	$b_{fu}$	R	CØMCBM	Velocity exponent for fuel atomization process, unitless
ADVØX	$b_{\phi x}$	R	CØMCBM	Velocity exponent for the oxidizer atomization
AINJ	$A_{inj}$	R	CØMARE	Cross sectional area at injector face, $m^2$
AMA(4,5)	--	C	CHAMDY	Array used to store coefficients of finite difference equations approximat- ing the oscillatory differential equations
ANGLEX	--	R	CØMNØZ	Nozzle convergence half angle, degrees (see Fig. 15)
ANS	--	R	SØLVW	Solution to minimization of error in nozzle admittance boundary equations. Returned to SØLVW from subroutine ZERO.
CFU1,...., CFU16	$c_{1fu}, \dots,$	C	CØMCBM	Fuel combustion coefficients, unitless
CMA(4)	$c_{16fu}$	C	CHAMDY	Right hand side of finite difference equation approximating the oscillatory differential equations. In equivalence with CMA(1,5).
CNØZA	$A_{Nu}$	C	CØMCHM	Upstream nozzle admittance
CØ	$c_{\phi}$	R	CØMCBM	Sonic velocity evaluated at overall mix- ture ratio, m/sec (ft/sec)
CØND1	--	R	SØLVW	Condition number of the transpose of the Jacobian of the difference between the upstream and downstream nozzle admit- tances with respect to $\omega$ .



<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
CØND2	--	R	SØLVW	Condition number of the transpose of the Jacobian of the difference between the upstream and downstream nozzle admittances with respect to $\omega$ .
CØX1, ..., CØX16	$c_{1\text{ox}}$	C	CØMCBM	Oxidizer combustion coefficients, unitless
CRR		R	CØMNØZ	Contraction ratio, cross-sectional area of chamber/throat area, unitless (see Fig. 15)
CS	$c^*$	R	CØMCBM	Characteristic velocity evaluated at the overall mixture ratio, m/sec (ft/sec)
CTEST	--	R	SØLVE	Upper bound on the condition number of the transpose of the Jacobian of the difference between the upstream and downstream nozzle admittances with respect to the complex frequency. If the condition number of that matrix exceeds CTEST for a given frequency, then it is assumed that Jacobian is singular near that frequency and hence a solution will not be sought at that point, unitless.
DA(100)	$\partial A/\partial x$	R	CØMARE	Slope of area of chamber at XM(I), m
DCSDMR	$\partial c^*/\partial MR$	R	CØMCBM	Partial derivative of the characteristic velocity with respect to mixture ratio holding, m/sec.
DELFRQ	--	R	SØLVE	Frequency increment used during the search procedure to solve the nozzle admittance boundary equation, Hz.
DELHFU	$\Delta h_{fu}$	R	CØMCBM	Pseudo energy term for fuel, J/kg (Btu/lbm)
DELHØX	$\Delta h_{ox}$	R	CØMCBM	Pseudo energy term for oxidizer, J/kg (Btu/lbm)
DELMX	--	R	SØLVE	Maximum change in $\omega$ allowed between iterations
DELP	$\Delta P$	R	CØMCHM	Oscillatory pressure at injector face, dimensionless

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
DELVFU	--	R	CØMCBM	Steady-state velocity difference between fuel droplets and gas stream normalized to the sonic velocity at the overall mixture ratio, unitless
DELVØX	--	R	CØMCBM	Steady-state velocity difference between oxidizer droplets and gas stream normalized to the sonic velocity at the overall mixture ratio, unitless
DELX	$\Delta x$	R	CØMARE	Axial distance between successive X(I), m
DET1	--	R	SØLVW	Determinant of the Jacobian of the difference between the upstream and downstream nozzle admittances with respect to $\omega$ .
DET2	--	R	SØLVW	Determinant of the Jacobian of the difference between the upstream and downstream nozzle admittances with respect to $\omega$ .
DFIDX	--	R	SØLVW	Derivative of Imag (FN) with respect to real ( $\omega$ ).
DFIDY	--	R	SØLVW	Derivative of real (FN) with respect to real ( $\omega$ ).
DHDMR	$(\partial h / \partial MR)_{\phi}$	R	CØMCBM	Partial derivative of gas reference enthalpy with respect to mixture ratio averaged over the mixture ratio range during steady-state operation, J/kg (Btu/lbm).
DM1	--	C	ADARND	Used to store intermediate values needed to compute boundary conditions and coefficients of finite difference equations approximating the oscillatory differential equations.
DM2	--	R	ADARND	Used to store intermediate values needed to compute boundary conditions and coefficients of finite difference equation approximating the oscillatory differential equations.

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
DM3	--	C	ADARND	Used to store intermediate values needed to compute boundary conditions and coefficients of finite difference equations approximating the oscillatory differential equations.
DM4	--	C	ADARND	↓
DM5	--	C	ADARND	
DM6	--	C	ADARND	
DM7FU	--	C	ADARND	
DM7ØX	--	C	ADARND	
DM8FU	--	C	ADARND	
DM8ØX	--	C	ADARND	
DM9FU	--	C	ADARND	
DM9ØX	--	C	ADARND	
DM22	--	R	ADARND	
DMRB(100)	$\partial \bar{M}R / \partial x$	R	CØNSTS	Derivative of steady-state mixture ratio with respect to distance, $m^{-1}$
DRGDMR	$\partial R / \partial x$	R	CØMCBM	Partial derivative of gas constant with respect to mixture ratio evaluated at the overall mixture ratio, J/kmole/°K (ft-lb/lb/°R)
DRHQB(100)	$\partial \bar{\rho} / \partial x$	R	CØNSTS	Derivative of steady-state density with respect to distance, $kg/m^3/m$
DTFU DM	$\partial \tau_{fu} / \partial MR$	R	CØMCBM	Partial derivative of fuel vaporization time delay with respect to mixture ratio, holding the vaporization blockage term, droplet diameter, and Nusselt number constant, sec

<u>FORTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
DTØXDM	$\partial \bar{\tau}_{OX} / \partial MR$	R	CØMCBM	Partial derivative of oxidizer vaporization time delay with respect to mixture ratio, holding the vaporization blockage term, drop diameter, and Nusselt number constant, sec
EPSF	--	R	SØLVE	Relative error criterion used during the search algorithm for the portion of the program that minimizes the error, HN, in the nozzle admittance boundary equation with respect to the growth coefficient, $\omega_I$ . To obtain convergence, it is necessary that

$$\frac{\partial |HN|^2}{\partial \omega_I} < EPSFS$$

Unitless.

EPSFS	--	R	SØLVE	Tightened relative error criterion used to determine if convergence has been obtained while iterating to solve the nozzle admittance boundary equation. To obtain convergence, it is necessary that
-------	----	---	-------	---

$$|N_{A_U} - N_{A_D}| / |N_{A_D}| < EPSFS.$$

EPSX	--	R	SØLVE	Relative error criterion used during the search algorithm for the portion of the program that minimizes the error, HN, in the nozzle admittance boundary equation with respect to the growth coefficient, $\omega_I$ . To obtain convergence, it is necessary that
------	----	---	-------	--

$$|\omega_{I_1} - \omega_{I_2}| / |\omega_{I_2}| < EPSX$$

where the subscripts 1 and 2 refer to two successive iterations, unitless.

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
EPSXS	--	R	SØLVE	Tightened relative error criterion used to determine if convergence has been obtained while iterating to solve the nozzle admittance boundary equation. To obtain convergence, it is necessary that  $ \omega_1 - \omega_2 / \omega_2  < \text{EPSXS}$ where the subscripts 1 and 2 refer to successive iterations, unitless.
F1	--	R	SØLVW	Derivatives of absolute value squared of the difference between the upstream and downstream nozzle admittances with respect to Imag ( $\omega$ ) corresponding to X1, X2, and X3.
F2	--	R	SØLVW	↓
F3	--	R	SØLVW	
FANS	--	R	SØLVW	Derivative of squared error in nozzle admittance boundary equation at ANS.
FI	--	R	MAIN	Interpolating factor used in the main program.
FI1	--	R	SØLVW	Used to store successive values of Imag (FN) during the 2-dimensional secant method.
FI2	--	R	SØLVW	↓
FI3	--	R	SØLVW	
FI4	--	R	SØLVW	
FN	--	C	FZERØ	Difference between the upstream and downstream nozzle admittances.
FNF	--	R	FZERØ	Imaginary (FN)
FNR	--	R	FZERØ	Real (FN)
FR1	--	R	SØLVW	Used to store successive values of Real (FN) during the 2-dimensional secant method.

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
FR2	--	R	SØLVW	Used to store successive values of Real (FN) during the 2-dimensional secant method.
FR3	--	R	SØLVW	↓
FR4	--	R	SØLVW	
FREQ	$2\pi\omega_R$	R	SØLVE	
FREQMI	--	R	CØMNØZ	Minimum frequency, used for generation of frequency table FREQT, Hz
FREQMX	--	R	CØMNØZ	Maximum frequency used for generation of frequency table FREQT, Hz
FREQT(100)	--	R	CØMTAP	Table of terms used for computation of downstream nozzle admittance.
FRQMAX	--	R	SØLVE	Maximum frequency above which no solutions to the nozzle admittance boundary equation will be sought, Hz
FTST1	--	R	SØLVW	Test function used to determine if a solution to the nozzle admittance boundary equation has been bracketed.
FTST2	--	R	SØLVW	Test function used to determine if a solution to the nozzle admittance boundary equation has been bracketed.
G1FU	--	C	ADARND	Coefficient of oscillatory pressure in fuel oscillatory vaporization expression.
G1ØX	--	C	ADARND	Coefficient of oscillatory pressure in oxidizer oscillating vaporization expression.
G2FU	--	C	ADARND	Coefficient of oscillatory density in fuel oscillatory vaporization expression.
G2ØX	--	C	ADARND	Coefficient of oscillatory density in oxidizer oscillating vaporization expression.

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
G3FU	--	C	ADARND	Coefficient of oscillatory mixture ratio in fuel oscillatory vaporization expression.
G3ØX	--	C	ADARND	Coefficient of oscillatory mixture ratio in oxidizer oscillating vaporization expression.
G4FU	--	C	ADARND	Coefficient of oscillatory velocity in fuel oscillatory vaporization expression.
G4ØX	--	C	ADARND	Coefficient of oscillatory velocity in oxidizer oscillating vaporization expression.
GAMØ	$\gamma_\delta$	R	CØMCMØ	Specific heat ratio evaluated at the overall mixture ratio, unitless.
GINJFT(100)	$G_{injfu}$	C	CØMTAP	Fuel feed system response table. Real (GINJFT) is the amplitude of the response and Imag (GINJFT) is the phase angle of the response.
GINJOT(100)	$G_{injox}$	C	CØMTAP	Oxidizer feed system response table. Real (GINJØT) is the amplitude of the response and Imag (GINJØT) is the phase angle of the response.
GN	--	C	FZERØ	Variable used to store the value of the derivative of $CNØZA-NØZA$ with respect to $\omega_I$ or the value of $CNØZA-NØZA$ itself.
HN	--	R	FZERØ	$FN*FN$
I	--	I	-	Used throughout the program as a do loop index.
INPNØZ	--	I	CØMNØZ	Code used for the downstream nozzle admittance term calculation.
INRT	--	I	DUMP	Code used to determine whether or not intermediate output from CHAMDY is desired.
IPASS	--	I	SØLVE	Internal code no longer in use.

<u>FORTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
IPRNØZ	--	R	CØMNØZ	Code used for downstream nozzle admittance term printout
IR	--	I	MAIN	Flag set by MAIN program to indicate the first pass through it. After reading in a new case.
ISCNT	--	I	FZERØ	Code used to determine logical flow in subroutine SØLVW.
ISLP	--	I	FZERØ	Code used to determine whether or not the derivative of FN with respect to Imag ( $\omega$ ) is needed.
ISTRT	--	I	SØLVE	Code used to indicate the first iteration after a solution to the nozzle boundary equation.
II		C	ADARND	The imaginary number i.
ITAPC	--	I	CØMTAP	File number used to save combustion coefficients table.
ITAPH	--	I	CØMTAP	File number used to save hydrodynamic coupling term table.
ITAPN	--	I	CØMTAP	File number used to save nozzle admittance term table.
IWSKP	--	I	SØLVE	Intermediate output dump code used in subroutine SØLVW.
J	--	I	-	Used throughout the program as a do loop index.
KNTMX	--	I	SØLVE	Maximum number of iterations allowed to minimize the error in the nozzle admittance boundary equation with respect to the imaginary part of $\omega$ .
KNTR	--	I	SØLVE	Counter used to control the number of iterations used during the search algorithm between solutions.
KNTRMX	--	I	SØLVE	Maximum number of times the frequency will be allowed to be changed by DELFRQ during the searching algorithm between each solution.



<u>FORTTRAN</u> <u>VARIABLE</u> <u>NAME</u>	<u>ENGINEERING</u> <u>VARIABLE</u> <u>SYMBOL</u>	<u>TYPE</u>	<u>CONTROL</u> <u>SECTION</u>	<u>DESCRIPTION</u>
KNTSMX	--	I	SØLVE	Maximum number of iterations allowed for the convergence of the two-dimensional secant method used in SØLVW.
KSCNT4	--	I	SØLVE	Counter used to control the number of iterations used when ISCNT = 4.
KWHERE	--	I	MAIN	Flag to control logical flow in the MAIN program after a call to subroutine SØLVW.
MBFUI	$\bar{m}_{fu}$ <sub>inj</sub>	R	CØMCBM	Fuel injection mass flowrate, kg/sec (lb/sec).
MBØXI	$\bar{m}_{ox}$ <sub>inj</sub>	R	CØMCBM	Oxidizer injection mass flowrate, kg/sec (lb/sec).
MGI	$\bar{m}$ <sub>inj</sub>	R	CØNSTS	Steady-state gas flowrate at injector face, kg/sec.
MR(100)	$\overline{MR}$	R	CØMCHM	Oscillatory mixture ratio, dimensionless.
MRB(100)	$\overline{MR}$	R	CØNSTS	Steady-state mixture ratio, unitless.
MRGI	$\overline{MR}$ <sub>inj</sub>	R	CØNSTS	Steady-state gas mixture ratio at injector face, unitless.
MRNTFU	--	C	ADARND	Mixture ratio integral in fuel oscillatory vaporization expression.
MRNTØX	--	C	ADARND	Mixture ratio integral in oxidizer oscillatory vaporization expression.
MWG	$MW_{\phi}$	R	CØMCBM	Molecular weight of the gas evaluated at the overall mixture ratio, kg/kmole (lbm/lb-mole).
NFREQT	--	I	CØMTAP	Number of points in frequency table.
NØZA	$A_{ND}$	C	FZERØ	Downstream nozzle admittance
NØZAMR	$A_{NMR}=\text{constant}$	C	FZERØ	Term used in computation of downstream nozzle admittance
NØZAT(100)	--	C	CØMTAP	Table of terms used for computation of downstream nozzle admittance.

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
NRØØT	--	I	MAIN	Number of solutions to the downstream nozzle admittance boundary equation being sought.
NRT	--	I	MAIN	Used as do loop index for MAIN program. Counts the number of solutions to the nozzle admittance boundary equation.
NUBFU	$\overline{Nu}_{fu}$	R	CØMCBM	Average steady-state fuel Nusselt number used in vaporization expression, unitless.
NUBØX	$\overline{Nu}_{Øx}$	R	CØMCBM	Steady-state oxidizer Nusselt number used in vaporization expression, unitless.
NXP	--	I	CØMARE	Number of points in axial distance array, inclusion between XØ and the start of nozzle inlet.
NXPM1	--	I	CHAMDY	NXP-1
ØMEGA	$\omega$	C	CØMCHM	Complex frequency.
P(100)	$\rho'$	R	CØMCHM	Oscillatory pressure, dimensionless.
PI	$\pi$	R	-	pi (3.141593)
PC	$\bar{p}$	R	CØMCBM	Steady-state chamber pressure, N/in. <sup>2</sup> (psia).
PINTFU	--	C	ADARND	Pressure integral in fuel oscillatory vaporization expression.
PINTØX	--	C	ADARND	Pressure integral in oxidizer oscillatory vaporization expression.
RBSØX	--	C	ADARND	Collection of terms used in oxidizer oscillatory vaporization expression.
RCCX	--	R	CØMNØZ	Ratio of the radius of curvature at the nozzle inlet to the chamber radius at nozzle inlet, unitless (see Fig. 15).
RCTX	--	R	CØMNØZ	Ratio of the radius of curvature upstream of the throat to the chamber radius at nozzle inlet, unitless (see Fig. 15).

<u>FORTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
RGØ	$R_\phi$	R	CØMCBM	Gas constant evaluated at the overall mixture ratio, J/kmole/°K (Btu/lb mole/°R).
RHNTFU	--	C	ADARND	Density integral in fuel oscillatory vaporization expression.
RHNTØX	--	C	ADARND	Density integral in oxidizer oscillatory vaporization expression.
RHØ(100)	$\rho'$	R	CØMCHM	Oscillatory density, dimensionless.
RHØB(100)	$\bar{\rho}$	R	CØNSTS	Steady-state density, kg/m <sup>3</sup> .
RHØGI	$\bar{\rho}_{inj}$	R	CØNSTS	Steady-state gas density at injector face, kg/m <sup>3</sup>
RHØINJ	$\rho'_{inj}$	R	ADARND	Oscillatory density at injector face, unitless.
RINJ	--	R	CØMNØZ	Radius of the chamber at the injector, in.
RPSFU	--	C	ADARND	Collection of terms used in fuel oscillatory vaporization expression.
SSV1(100)	--	R	CØNSTS	Steady-state parameters computed in subroutine STEADY for use by subroutine CHAMDY
SSV2(100)	--	R	CØNSTS	
SSV3(100)	--	R	CØNSTS	
SSV4(100)	--	R	CØNSTS	
SSV5(100)	--	R	CØNSTS	
SSV6(100)	--	R	CØNSTS	
SSV7(100)	--	R	CØNSTS	
SSV8(100)	--	R	CØNSTS	
SSV9FU(100)	--	R	CØNSTS	
SSV9ØX(100)	--	R	CØNSTS	

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
SSV10(100)	--	R	CØNSTS	Steady-state parameters computed in subroutine STEADY for use by subroutine CHAMDY.
SSV11(100)	--	R	CØNSTS	↓
SSV12(100)	--	R	CØNSTS	
SSV13(100)	--	R	CØNSTS	
SSV14(100)	--	R	CØNSTS	
SSV15(10)	--	R	CØNSTS	
T(100)	$\bar{T}$	R	CØMCHM	
TAUBFU	$\bar{\tau}_{fu}$	R	CØMCBM	Fuel vaporization time delay, sec.
TAUBØX	$\bar{\tau}_{ox}$	R	CØMCBM	Oxidizer vaporization time delay, sec.
TB(100)	$\bar{T}$	R	CØNSTS	Steady-state temperature, °K.
TDRAGF	$\tau_{drag_{fu}}$	R	CØMCBM	Steady-state oxidizer drag time delay, sec.
TDRAGØ	$\tau_{drag_{ox}}$	R	CØMCBM	Steady-state oxidizer drag time delay, sec.
TITLE(18,2)	--	R	MAIN	Array containing 2 card records of alpha-numeric title information.
V(100)	$v'$	R	CØMCHM	Oscillatory velocity, dimensionless.
VAPBFU(100)	$\bar{m}_{vap_{fu}}$	R	CØNSTS	Steady-state fuel vaporization rate, kg/sec/m <sup>3</sup> .
VAPBØX(100)	$\bar{m}_{vap_{ox}}$	R	CØNSTS	Steady-state oxidizer vaporization rate, kg/sec/m <sup>3</sup> .
VB(100)	$\bar{v}$	R	CØNSTS	Steady-state velocity, m/sec.
VBFU	$v_{j_{fu}}$	R	CØMCBM	Steady-state fuel liquid injection velocity, m/sec (ft/sec).
VBØX	$v_{j_{ox}}$	R	CØMCBM	Steady-state oxidizer liquid injection velocity, m/sec (ft/sec).
VGI	$\bar{v}_{inj}$	R	CØNSTS	Steady-state gas velocity at injector face, m/sec.

<u>FORTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
VINTFU	--	C	ADARND	Velocity integral in oxidizer oscillatory vaporization expression.
VINTOX	--	C	ADARND	Velocity integral in oxidizer oscillatory vaporization expression.
VX0	$v'_{x=0}$	C	C0MCHM	Oscillatory velocity at X=0, dimensionless.
X(100)	X	R	C0MARE	Axial distance array, $X(I) = X0 + (X-1)* DELX$ , m.
X0	$X_0$	R	C0MARE	Start of vaporization point, m
X1	--	R	S0LVW	Used to store successive values of $Imag(\omega)$ during the iteration to minimize the error in the nozzle admittance boundary equation with respect to $Imag(\omega)$ .
X2	--	R	S0LVW	
X3	--	R	S0LVW	
XIMPFU	$ximp_{fu}$	R	C0MCBM	Fuel jet injection impingement point, m (in.).
XIMP0X	$ximp_{ox}$	R	C0MCBM	Oxidizer jet injection impingement point, m(in.).
XI1	--	R	S0LVW	Used to store successive values of $Imag(\omega)$ during the 2-dimensional secant method.
XI2	--	R	S0LVW	
XI3	--	R	S0LVW	
XI4	--	R	S0LVW	
XKFU	$xk_{fu}$	R	C0MCBM	Fuel Klystron distance, m(in.).
XK0X	$xk_{ox}$	R	C0MCBM	Oxidizer Klystron distance, m(in.).

<u>FORTTRAN VARIABLE NAME</u>	<u>ENGINEERING VARIABLE SYMBOL</u>	<u>TYPE</u>	<u>CONTROL SECTION</u>	<u>DESCRIPTION</u>
XM(100)	--	R	CØMARE	Axial distance midpoints, $XM(I) = (X(I) + X(I-1))/2$ , m.
XNØZ	--	R	CØMARE	Nozzle inlet point, m.
XR1	--	R	SØLVW	Used to store successive values of Real ( $\omega$ ) during the 2-dimensional secant method.
XR2	--	R	SØLVW	↓
XR3	--	R	SØLVW	
XR4	--	R	SØLVW	

APPENDIX 3

PROGRAM FLOW CHARTS

<u>FLOW CHART INDEX</u>	<u>PAGES*</u>
Cross Reference Index	1-10
MAIN (Procedures)	01-09
AREA	10-11
CHAMDY	12-14
CHMCØN	15-16
COBAEL	17-19
CØMBDY	20-22
CØMMAT	23-24
FRESP	25-38
HEAD	39-40
HY (Block Data)	41-42
HYDRDY	43-49
LØCFAC	50-52
NØZADM	53-56
RKTDIF	57-58
RKTZ	59-60
RKZDIF	61-62
SØLVW	63-70
STEADY	71-72
TADAMS	73-76
TDPLØT	77-79
XIMAGE	80-81
ZADAMS	82-85
ZERØ	86-89

\*Flow chart page numbers in upper right-hand corner.

FORTRAN MODULE COMPUTER PROGRAM

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - PROCEDURES

0000690	2.01	1	00002090	3.19	00003980	7.14
00003600	2.12	9000	00000750	2.05	00000860	2.10
00001480	2.35		00001460	2.33		
00001710	3.09		00001720	3.10		
00001720	3.10	1050				
00001760	3.11	1060	00001600	3.02		
00001920	3.15		00003570	7.14		
00002080	3.18	10	00002990	6.01		
00002150	3.21	15	00002990	6.01		
00002190	3.24		00002150	3.22		
00002250	3.30		00002245	3.27		
00002260	4.01	20	00002220	3.24		
00002300	4.06		00002310	4.08		
00002310	4.08	22				
00002320	4.09	25	00002250	3.30		
00002330	4.10	30	00002260	4.01		
00002340	4.11	40	00002240	3.26		
00002410	4.18	45	00002405	4.15		
00002420	4.21	50	00002390	4.13		
00002420	4.21		00002410	4.18		
00002460	4.26		00002500	4.28		
00002500	4.28	55				
00002510	5.01	60	00002420	4.21		
00002530	5.02	70	00002408	4.17		
00002600	5.06	80	00002570	5.03		
00002670	5.11	82	00002600	5.06		
00002690	5.13	84				
00002700	5.14	86	00002750	5.21		
00002710	5.15	88				
00002710	5.15		00002710	5.16		
00002730	5.19	90	00002790	5.26		
00002750	5.21	92	00002680	5.12		
00002780	5.24	94				
00002780	5.24		00002780	5.26		
00002800	5.27	96	00002680	5.12	00002740	5.20
00002820	5.29		00002830	5.31		
00002830	5.30	100				
00002940	5.32	110	00002590	5.05		
00002890	5.35		00002880	5.33		
00003050	6.02	225	00002990	6.01	00002990	6.01
00003090	6.06		00003070	6.03		
00003130	6.09		00003160	6.11		
00003160	6.11	1065				
00003180	6.12	1090	00003080	6.06		
00003280	6.20		00003275	6.17		
00003340	6.22		00002530	7.13		
00003360	6.25		00003350	6.23		
00003370	6.27		00003360	6.25		
00003400	6.30		00003390	6.26		
00003410	6.32		00003400	6.30		
00003440	7.03		00003430	7.01		
00003450	7.05		00003440	7.03		
00003480	7.08		00003470	7.06		
00003490	7.10		00003480	7.08		
00003530	7.13	300				
00003570	7.14	5000	00002990	6.01	00003270	6.16

ORIGINAL PAGE IS  
 OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE AREA

00000030	10.01	AREA	00001560	3.01-X
00000160	10.03		00000190	10.04
00000190	10.04	10		



CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
00000220	10.06		00000230 10.07
00000230	10.07		20

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE CHANDY

00000030	12.01	CHANDY	00000480 00.05-X
00001010	12.10	10	00001940 13.13
00001010	12.10		00000970 12.15
00001800	13.07		00001760 13.04

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE CHMCON(1)

00000030	15.01	CHMCON	00001050 12.20-X
00000510	15.01	10	

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE COGAEL(A;N)

00012030	17.01	COGAEL	00009020 33.44-X
00012130	17.03		00012760 18.12
00012200	17.05	25	
00012210	17.06		00012250 17.09
00012230	17.08	3	
00012250	17.09	2	00012220 17.07
00012300	17.11	6	00012190 17.04 00012660 18.04
00012340	17.14	4	00012290 17.10
00012350	17.15		00012420 17.17
00012420	17.17	7	
00012460	17.18	5	00012190 17.04 00012290 17.10
00012480	17.20		00012610 18.03
00012500	17.22	14	
00012510	17.23	15	
00012570	18.01	13	00012490 17.21 00012500 17.22
00012610	18.03	8	
00012670	18.05	26	
00012690	18.07		00012750 18.11
00012700	18.08		00012750 18.10
00012750	18.10	9	
00012760	18.12	27	00012660 18.04

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE COMEDY(1R,FREQ,GINJGX,GINJFU,1PRCON,1NPRCON)

00000030	20.01	COMEDY	00002580 5.04-X
00000910	20.19	100	00000330 20.01
00001050	20.25	200	00000990 20.21 00001600 21.15
00001570	21.12		00001550 21.10
00001610	21.16	900	00001580 21.14

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE COMMAT(A,NRA,N)

00000310	23.02	20	
00000390	23.05		00000430 23.10
00000420	23.07	40	
00000420	23.07		00000420 23.08
00000430	23.09	50	
00000030	23.12	COMMAT	00001530 12.35-X
00000230	23.14		00000310 23.02
00000270	23.17		00000290 23.01
00000290	23.19	10	
00000290	23.19		00000290 23.20
00000490	23.21	100	00000240 23.15

## CHART TITLE - NON-PROCEDURAL STATEMENTS

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE	FRESP (CRT,IMRITE,IX)				
00006010	25.01	FRESP	00005300	48.08-X	
00006210	25.01	10			
00006220	25.02		00006230	25.05	
00006230	25.03	2001			
00006230	25.03		00006230	25.04	
00006260	25.08		00006270	26.01	
00006270	25.09	2002			
00006270	25.09		00006270	25.10	
00006290	26.03		00006300	26.06	
00006300	26.04	2003			
00006300	26.04		00006300	26.05	
00006310	26.07	2006	00006240	25.06	
00006320	26.08		00006340	26.12	
00006330	26.09		00006340	26.11	
00006340	26.10	2004			
00006380	26.17		00006360	26.14	
00006390	26.18	12:16			
00006400	26.19	12:18			
00007300	26.20	444	00006350	26.13	
00007340	26.23	445			
00007340	26.23		00007340	26.24	
00007360	26.27		00007350	26.25	
00007370	26.28	45	00007300	26.20	00007100 29.23 00007290 30.09
00006430	27.01	12:17	00006390	26.18	
00006450	27.04	12:15	00006380	26.17	
00006450	27.04		00006430	27.01	
00006460	27.05	12:20			
00006490	27.06	12:19	00006450	27.04	
00006510	27.09		00006490	27.06	
00006520	27.10	12:22			
00006550	27.11	12:21	00006510	27.09	
00006560	27.12	3			
00006680	27.13	16:71	00006550	27.11	
00006690	27.14	16:75			
00006750	27.15	4	00006780	27.16	
00006760	27.16	5			
00006580	28.01	16:67	00006550	27.11	
00006610	28.02	16:68	00006640	28.04	
00006630	28.04	16:69			
00006650	28.05	16:70	00006620	28.03	
00006790	29.01	8	00006750	27.15	
00006820	29.03	12	00006930	29.12	00006980 29.13 00007020 29.16
00006850	29.05	23			
00006860	29.06	24	00006840	29.04	
00006880	29.08	27			
00006890	29.09	25	00006870	29.07	
00006920	29.10	26	00006870	29.07	00006880 29.08
00006930	29.12	11	00006920	29.10	
00006940	29.13	13			
00006990	29.14	15	00006930	29.12	
00007010	29.15	16	00006920	29.10	
00007030	29.17	18			
00007050	29.18	17	00006670	28.05	
00007070	29.20	20	00007100	29.23	
00007090	29.22	30			
00007110	29.24	29	00007060	29.19	
00007130	29.25	35	00007180	29.28	00007260 30.06 00007270 30.07
00007150	29.27	20			
00007160	29.28	33			
00007190	30.01	34	00007140	29.26	00007150 29.27
00007220	30.03	37			
00007230	30.04	31			
00007250	30.05	36	00007210	30.02	
00007270	30.07	38	00007240	30.04	
00007280	30.08	380	00007220	30.03	
00007400	30.10	32	00007360	26.27	00007080 29.21 00007270 30.07 00007290 30.09
00007410	30.11	431			
00007420	30.12	432			
00007440	30.13	43	00007410	30.11	00007510 30.17

06/25/75 TABLE OF CONTENTS AND REFERENCES

AUTOFLOW CHART SET - FSCSM  
REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
00007450	30.14	39	
00007480	30.15	40	00007440 30.13
00007490	30.16	41	00007470 30.14
00007500	30.17	42	
00007520	31.01	44	00007490 30.16 00007680 31.14
00007540	31.04		00007520 31.01
00007560	31.06		00007670 31.13
00007570	31.07	301	
00007580	31.08	48	
00007620	31.10	300	00007560 31.06
00007630	31.11	302	
00007670	31.13	47	00007570 31.07 00007610 31.09 00007620 31.10
00007680	31.14	49	
00007690	31.15	60	00007540 31.04 00007610 31.09 00007660 31.12
00007740	31.18		00007710 31.16
00007750	31.20		00007740 31.18
00007770	31.23		00007750 31.20
00007780	31.24	111	
00007780	31.24		00007780 31.27
00007780	31.27		00007780 31.24
00007820	31.30		00007800 31.28
00007830	31.31	311	
00007830	31.31		00007830 31.34
00007830	31.34		00007830 31.31
00008010	32.01	69	00007980 32.19
00008080	32.03	55	00008170 32.07
00008110	32.05	56	
00008180	32.08	57	00008100 32.04
00007850	32.09	309	
00007860	32.12	75	00007840 31.35 00009690 35.20
00007860	32.12		00007850 32.09
00007870	32.13		00008520 33.04
00007920	32.14	76	00008510 33.03
00007940	32.16	62	
00007950	32.17	65	
00007970	32.18	67	00008000 32.24
00008460	32.20	7	00008100 32.24 00008170 32.07 00007930 32.15 00008350 32.28 00008420 32.31
00008210	32.21	64	00007940 32.16 00008240 32.23
00008230	32.23	71	
00007990	32.24	68	
00008250	32.25	70	00008220 32.22
00008320	32.27	772	00008420 32.31
00008360	32.29	773	
00008430	32.32	63	00008350 32.28
00008490	33.01	610	00007930 32.15
00008500	33.02	61	00008200 32.08 00008450 32.32
00008510	33.03	320	
00008520	33.04	662	
00008550	33.07		00008650 33.14
00008560	33.08		00008650 33.13
00008590	33.10	316	
00008610	33.11	312	00008580 33.09
00008650	33.13	313	00008580 33.09
00008680	33.17		00008740 33.21
00008700	33.19	72	
00008740	33.21	73	00008690 33.18
00008770	33.24	303	
00008780	33.26		00008770 33.24
00008790	33.27		00008800 33.31
00008800	33.28	304	
00008800	33.31		00008800 33.28
00008810	33.32	305	00008760 33.23
00008870	33.35		00008910 33.39
00008900	33.37		00008910 33.38
00008910	33.38	202	
00008950	33.42		00008980 33.43
00008980	33.43	203	
00009140	34.04		00009260 34.11
00009190	34.06		00009220 34.08
00009220	34.08	206	
00009260	34.10	205	
00009280	34.13	82	

ORIGINAL PAGE IS  
OF POOR QUALITY

06/25/75 TABLE OF CONTENTS AND REFERENCES

AUTOFLOW CHART SET - FSCSH  
REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
00009350	34.17	770	
00009350	34.17		00009290 34.14
00009420	34.19	771	
00009420	34.19		00009350 34.17
00009450	34.21		00009510 35.06
00009460	34.22	79	
00009470	34.23	80	00009450 34.21
00009480	35.01	81	00009450 34.21
00009490	35.03		00009480 35.01
00009500	35.05		00009490 35.03
00009510	35.06	78	00009470 34.23
00009530	35.08	83	00009270 34.12
00009540	35.09		00009570 35.10
00009570	35.10	84	
00009580	35.11	85	
00009590	35.12	90	
00009590	35.12		00009610 35.14
00009600	35.13	96	
00009610	35.14	87	
00009630	35.16	960	
00009670	35.19	961	00009620 35.15
00009670	35.19		00009630 35.16
00009660	35.20	85	
00009700	35.21	95	00009670 35.19
00009770	35.23	97	00009660 35.32
00009780	35.24	98	
00009790	35.27		00009780 35.24
00009800	35.28		00009620 35.31
00009820	35.31	99	
00009820	35.31		00009800 35.28
00009870	36.01	100	00009770 35.23
00009910	36.03		00010220 36.35
00009920	36.04		00009990 36.10
00009970	36.08		00009950 36.06
00009990	36.10	102	00009940 36.05
00009990	36.10		00009970 36.08
00010010	36.16		00010000 36.14
00010020	36.18		00010010 36.16
00010040	36.20	103	
00010120	36.25	104	00010030 36.19
00010190	36.32	105	00010110 36.13
00010220	36.35	400	00010020 36.18
00010230	36.36	999	
00010240	37.01	9	00006420 26.19 00008480 32.20 00007350 26.28 00006480 27.05 00006540 27.10 00006570 27.12

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE HEAD

00000030 39.01 HEAD 00000710 2.02-X

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - BLOCK DATA

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE HYDRDY(1R,INPHYD,FRE,GIND,GINF,PCIN,MOIN,WFINI)

00000010	43.01	HYDRDY	00002400 4.14-X
00001120	43.05		00001111 43.03
00001170	43.06	5	00005309 48.14
00001180	43.07		00001190 43.10
00001190	43.08	10	
00001190	43.08		00001190 43.09
00001210	43.12		00001220 43.15
00001220	43.13	20	
00001220	43.13		00001220 43.14
00001240	43.17	30	
00001240	43.17		00001240 43.18

06/25/75 TABLE OF CONTENTS AND REFERENCES

AUTOFLOW CHART SET - FSCSM  
REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

CARD ID	PAGE/BOX	NAME	REFERENCES	(SOURCE SEQUENCE NO. AND PAGE/BOX)
00001410	43.22	35		
00001440	43.25		00001490	44.01
00001490	43.26	40		
00001600	44.04	45	00001420	43.23
00001610	44.07		00001600	44.04
00001620	44.09		00001610	44.07
00001650	44.11	60	00001600	44.06
00001650	44.11		00001620	44.09
00001660	44.14		00001650	44.11
00001680	44.16		00001660	44.14
00001685	44.18	100	00001410	43.22
00001685	44.18		00001680	44.16
00001930	44.19	999	00001650	44.13
00001690	44.21		00001685	44.18
00001710	44.22	101		
00001710	44.22		00001710	44.23
00005280	48.06	400		
00005300	48.08	500	00005280	48.06
00005306	48.11	502	00005301	48.09
00005308	48.13	503		
00005308	48.13		00005308	48.14
00005303	48.15		00005304	48.17
00005304	48.16	501		
00005310	48.18	510	00005306	48.11
00005340	48.22		00005320	48.19
00005400	48.26	700	00005310	48.18

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE LOCFAC(J,K,X, TX, NX, JX, FX)

00000030	50.01	LOCFAC	00002150	3.23-X	00002320	4.09-X		
00000180	50.06		00000170	50.04				
00000610	50.10	200	00000150	50.02	00000440	51.05	00000460	51.10
00000270	50.11	30	00000220	50.09				
00000280	50.12		00000290	50.13				
00000290	50.13	40						
00000240	50.15		00000250	50.16				
00000250	50.16	20						
00000320	50.17	50	00000280	50.12	00000240	50.15		
00000340	50.18	70	00000590	51.17				
00000400	51.01	90	00000190	50.07	00000300	50.13	00000260	50.16
00000430	51.04		00000450	51.07				
00000450	51.06	100						
00000460	51.08	110	00000410	51.02	00000440	51.05		
00000470	51.09		00000490	51.11				
00000490	51.11	120						
00000500	51.12	130	00000480	51.10				
00000510	51.14		00000500	51.12				
00000560	51.17	150	00000520	51.15	00000530	51.16		

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE NOZADM(1R,GAPX,CO,FREQ,NOZA)

00000030	53.01	NOZADM	00002230	3.25-X				
00000290	53.03		00000270	53.01				
00000410	53.08	2	00000290	53.03				
00000470	53.12		00000460	53.09				
00000540	53.15	2000	00000490	53.13				
00000570	53.16	2010	00000530	53.14				
00000620	53.20	5	00000660	53.22				
00000670	53.23	7	00000600	53.18				
00000700	53.27		00000680	53.24				
00000710	53.28	20						
00000710	53.28		00001810	55.24				
00000720	53.29	25						
00001110	54.09		00001220	54.13				
00001220	54.12	30						
00001440	55.01	35	00001390	54.19				
00001490	55.03	40	00001430	54.20				
00001940	55.06	45						

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
00001680	55.13		00001660 55.10
00001690	55.14 50		00001530 55.05
00001730	55.18 8		00001660 55.13
00001730	55.18		00001710 55.15
00001810	55.24 10		00001290 54.16 00001740 55.19
00001810	55.24		00001800 55.21
00001900	55.27		00001890 55.25
00001910	55.29		00001900 55.27

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE RKTDF(P,G,GP)

00002650	57.01	RKTDF	00003750 59.07-X 00003860 59.20-X 00003950 59.31-X
00002620	57.04 22		
00002630	57.05 25		
00002920	57.06 50		00002810 57.03
00002930	57.07 55		00002910 57.15
00002950	57.12 30		00002820 57.04
00002860	57.13 35		
00002880	57.14 40		00002850 57.12
00002890	57.15 45		00002840 57.05 00002870 57.13

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE RKTZ(NU,H,T,U,DUM,JOPT)

00003580	59.01	RKTZ	00001110 54.09-X 00001110 75.07-X 00002410 64.07-X
00003720	59.03		00003730 59.05
00003730	59.04 10		
00003770	59.09 15		00003740 59.06
00003780	59.10 20		00003760 59.08
00003790	59.11 25		
00003790	59.11		00003790 59.12
00003810	59.14		00003910 56.26
00003830	59.16		00003840 59.18
00003840	59.17 35		
00003880	59.22 40		00003850 59.19
00003890	59.23 45		00003870 59.21
00003900	59.24 50		
00003900	59.24		00003900 59.25
00003910	59.26 30		
00003930	59.28 55		
00003930	59.28		00003930 59.29
00003950	59.31 60		00003940 59.30
00003970	59.33 65		00003940 59.30
00003980	59.34 70		00003960 59.32
00004050	59.39 75		00003980 59.34

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE RKZDF(P,G,GP)

00003070	61.01	RKZDF	00003770 59.09-X 00003880 59.22-X 00003970 59.33-X
00003210	61.03 10		
00003270	61.04 15		00003200 61.02
00003320	61.07		00003310 61.05
00003330	61.09		00003320 61.07
00003340	61.10 22		
00003350	61.11 25		
00003440	61.12 50		00003330 61.09
00003450	61.13 55		00003430 61.20
00003540	61.16 20		00003260 61.03
00003370	61.17 30		00003340 61.10
00003380	61.18 35		
00003400	61.19 40		00003370 61.17
00003410	61.20 45		00003360 61.11 00003390 61.18

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE SOLV(KOMHERE)

CARD 10	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
0000030	63.01	SOLVN	00002980 5.35-X
00003940	63.06	225	00000410 63.03
00000840	64.01	117	00000780 64.10
00000910	64.04	118	00001020 65.01
00000780	64.10		00000750 64.07
00000600	64.14	110	00000970 64.06 00000830 64.11
00000670	64.15	115	
00001310	64.19	145	00000720 64.17 00001050 65.04 00001210 65.09
00001010	65.01	120	00000840 64.01 00000850 64.02
00001030	65.02	125	00000670 64.15
00001050	65.03	135	
00001140	65.07	137	00001070 65.05
00001220	65.10	140	00000920 64.05
00001530	65.12	146	00000520 64.12
00001760	66.01		00001730 65.18
00001780	66.04		00001760 66.01
00001940	66.13	147	00002590 66.34
00001990	66.15	150	00001800 66.06 00002950 67.25
00002140	66.19	160	00001880 66.11
00002160	66.21	170	00001810 66.07
00003780	66.25	205	00001900 66.12 00003360 68.13
00002360	66.27	180	00002140 66.19
00002470	66.29	181	00002660 67.17
00002510	66.30	182	00002740 67.09
00002560	66.33	1825	00002530 66.31
00002220	67.01	175	00002180 66.23
00002240	67.02	177	00002210 66.24
00003000	67.03	185	00002540 67.15
00003020	67.04	190	00003530 68.16 00003660 68.20 00003720 68.22 00003770 68.23
00002600	67.10	193	00001790 66.05
00002790	67.19	184	00002610 67.11 00002650 67.16
00002970	67.25		00002890 67.22
00003200	67.26	220	00002160 66.22 00002620 67.13 00003330 68.11
00003170	68.01	195	00000460 63.04
00003220	68.05		00003200 68.02
00003260	68.08		00003220 68.05
00003570	68.17	200	00003370 68.14
00003620	68.20	201	
00003670	68.21	202	00003600 68.18
00003730	68.23	203	00003610 68.19 00003670 68.21
00003880	69.01	10	00002100 66.17 00002130 66.18 00002260 67.02
00003900	69.02	15	00001470 64.22 00003160 67.08
00003960	69.03	5000	00001260 65.11 00003930 66.26
00003970	69.04	6000	00003950 63.06 00003930 67.26 00003890 69.01 00003910 69.02

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE STEADY (IPRSTE)

00000030	71.01	STEADY	00002880 5.34-X
00000440	71.03		00000360 71.01
00000520	71.07	15	00000470 71.04
00000530	71.08	16	00000510 71.06
00000590	71.10		00001240 71.34
00001110	71.29		00001080 71.27
00001240	71.34	30	00001130 71.29

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE TADAMS (N,H,X,Y,DY,102,10)

00000030	73.01	TADAMS	00002580 64.13-X
00000110	73.01	10	00000920 74.25
00000130	73.03		00000150 73.04
00000150	73.04	15	
00000270	73.08	17	
00000280	73.09	20	
00001300	73.10	100	00000260 73.07 00000660 74.09
00001320	73.11	105	00001290 75.12
00000300	73.12	25	00000270 73.08
00000310	73.13	30	
00000330	73.14	35	00000300 73.12

ORIGINAL PAGE IS  
OF POOR QUALITY

## 06/25/75 TABLE OF CONTENTS AND REFERENCES

## AUTOFLOW CHART SET - FSCSM

PAGE 9

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
0000340	73.15	40	0000290 73.09 0000320 73.13
0000480	73.21		0000490 73.23
0000490	73.22	45	
0000560	74.02	52	
0000570	74.03		0000590 74.05
0000590	74.04	55	
0000630	74.08	60	
0000670	74.10	62	
0000700	74.11	70	0000670 74.10
0000710	74.12	75	
0000680	74.13	65	
0000730	74.14	80	0000700 74.11
0000740	74.15	85	0000720 74.12 0000690 74.13
0000520	74.25	87	0000660 74.20
0000930	75.01	90	0000620 74.07
0001110	75.07		0001220 75.11
0001220	75.10	95	

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE TDPL0T(M,Y,NP,TL,XR,LL)

00011002	77.01	TDPL0T	00010090 36.11-X	00010050 36.21-X	00010130 36.26-X	00010170 36.30-X
00011050	77.03		00011070 77.04			
00011070	77.04	100				
00011120	77.07	5				
00011150	77.08	201	00011110 77.06			
00011190	77.09	200	00011140 77.07			
00011220	77.11	10	00011270 77.19			
00011260	77.17		00011250 77.15			
00011270	77.19		00011260 77.17			
00011280	77.20	20	00011240 77.14			
00011290	77.21	30	00011230 77.12			
00011310	77.23	31				
00011330	77.25	50				
00011350	77.27	60	00011300 77.22	00011320 77.24		
00011360	77.28	61	00011340 77.26			
00011380	77.30	180	00011430 78.03	00011430 78.03	00011430 78.03	
00011390	77.31	181	00011380 77.30			
00011410	78.01	182	00011380 77.30			
00011420	78.02	40	00011400 77.32			
00011440	78.04	190	00011430 78.03			
00011450	78.05	191				
00011470	78.07	192	00011440 78.04			
00011480	78.08	193	00011460 78.06			

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - FUNCTION XIMAGF(XIMONG)

0000480	80.05	10	0000370 80.03	0000700 80.13
0000630	80.12	20	0000510 80.08	
0000620	80.14		0000610 80.10	
0000750	80.15	30	0000620 80.14	

## CHART TITLE - NON-PROCEDURAL STATEMENTS

## CHART TITLE - SUBROUTINE ZADAMS(M,H,X,Y,DY,IZ)

00001360	82.01	ZADAMS	00001280 84.15-X	
00001440	82.01	10	00002210 83.33	00002590 84.14
00001460	82.03		00001480 82.04	
00001480	82.04	15		
00001600	82.08	17		
00001610	82.09	20		
00001630	82.10	25	00001600 82.08	
00001640	82.11	30		
00001660	83.01	35	00001630 82.10	
00001570	83.02	40	00001620 82.09	00001650 82.11
00001780	83.07		00001800 83.09	
00001800	83.08	45		
00001870	83.11	52		



CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
00001880	83.12		00001800 83.14
00001900	83.13	55	
00001930	83.17	60	
00001970	83.19	62	
00001980	83.20	65	
00002000	83.21	70	00001970 83.19
00002010	83.22	75	
00002030	83.23	80	00002000 83.21
00002040	83.24	85	00001980 83.20 00002020 83.22
00002210	83.33	87	00002150 83.28
00002220	84.01	90	00001920 83.16
00002410	84.07		00002320 84.11
00002520	84.10	85	
00002610	84.15	100	00001980 82.07 00001950 83.18 00002580 84.14

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE ZERO(F1,T11,T21,F11,F21,ANS,FANS,EPSP,EPSPX,AF,ICNT,NDN

00000240	86.01	ZERO	00001190 85.08-X
00000160	86.04	5	00000240 86.07
00000200	86.05	10	
00000220	86.07	15	00000130 86.02
00000250	87.01	50	00000160 86.04
00000270	87.04		00000260 87.02
00000310	87.05	220	00000850 87.29 00000870 88.07
00000950	87.12	500	00000330 87.06
00000910	87.14	300	00000390 87.09
00000930	87.15	400	00000670 87.32 00000780 88.04
00000980	87.23	600	00000500 87.19
00000950	87.25	230	00000520 87.21
00000600	87.26	240	00000940 87.22
00000620	87.28	245	00000480 87.17
00000680	87.30	250	00000600 87.26
00000680	87.35	270	00000620 87.28
00000720	88.01	255	00000680 87.33
00000770	88.02	260	00000610 87.27
00000830	88.07	265	00000790 88.05

CHART TITLE - NON-PROCEDURAL STATEMENTS

LOCATION		DIAGNOSTIC
CARD ID	PAGE/BOX	
00010100	36.12	UNDEFINED - 'PRINTV' EXTERNAL REFERENCE
00010060	36.22	UNDEFINED - 'PRINTV' EXTERNAL REFERENCE
00010070	36.23	UNDEFINED - 'PRINTV' EXTERNAL REFERENCE
00010140	36.27	UNDEFINED - 'PRINTV' EXTERNAL REFERENCE
00010150	36.28	UNDEFINED - 'PRINTV' EXTERNAL REFERENCE
00010180	36.31	UNDEFINED - 'PRINTV' EXTERNAL REFERENCE
00010190	36.32	UNDEFINED - 'PRINTV' EXTERNAL REFERENCE
00010210	36.34	UNDEFINED - 'LABLV' EXTERNAL REFERENCE
00011310	77.23	UNDEFINED - 'CAMRAV' EXTERNAL REFERENCE
00011330	77.25	UNDEFINED - 'SETH1V' EXTERNAL REFERENCE
00011350	77.27	UNDEFINED - 'SETH1V' EXTERNAL REFERENCE
00011360	77.28	UNDEFINED - 'SHX1V' EXTERNAL REFERENCE
00011370	77.29	UNDEFINED - 'GRID1V' EXTERNAL REFERENCE
00011390	77.31	UNDEFINED - 'APLOTV' EXTERNAL REFERENCE
00011410	78.01	UNDEFINED - 'APLOTV' EXTERNAL REFERENCE
00011450	78.05	UNDEFINED - 'LABLV' EXTERNAL REFERENCE
00011470	78.07	UNDEFINED - 'LABLV' EXTERNAL REFERENCE

06/25/75

AUTOFLON CHART SET - FSCSP COMPUTER PROGRAM

PAGE 01

CHART TITLE - INTRODUCTORY COMMENTS

MAIN CONTROL PROGRAM FOR THE FEED SYSTEM COUPLED STABILITY MODEL.  
PROGRAM DEVELOPED BY ROCKETDYNE, A DIVISION OF ROCKWELL  
INTERNATIONAL, CANOGA PARK, CALIF 91304  
PROGRAMMED BY H. D. SCHUMAN, ROCKETDYNE, MAY 1975

ORIGINAL PAGE IS  
OF POOR QUALITY

R-9806/8-13

CHART TITLE - PROCEDURES

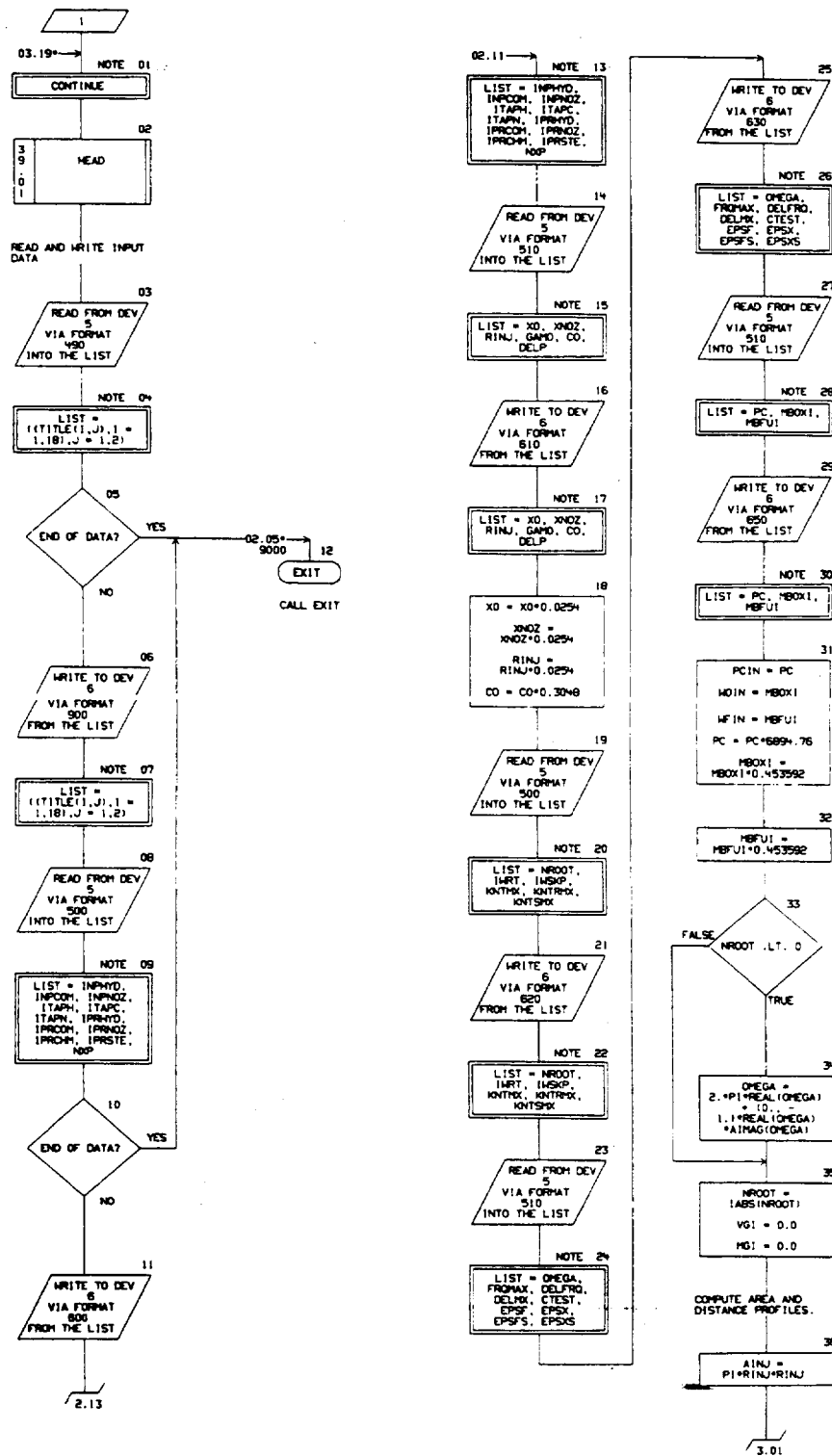
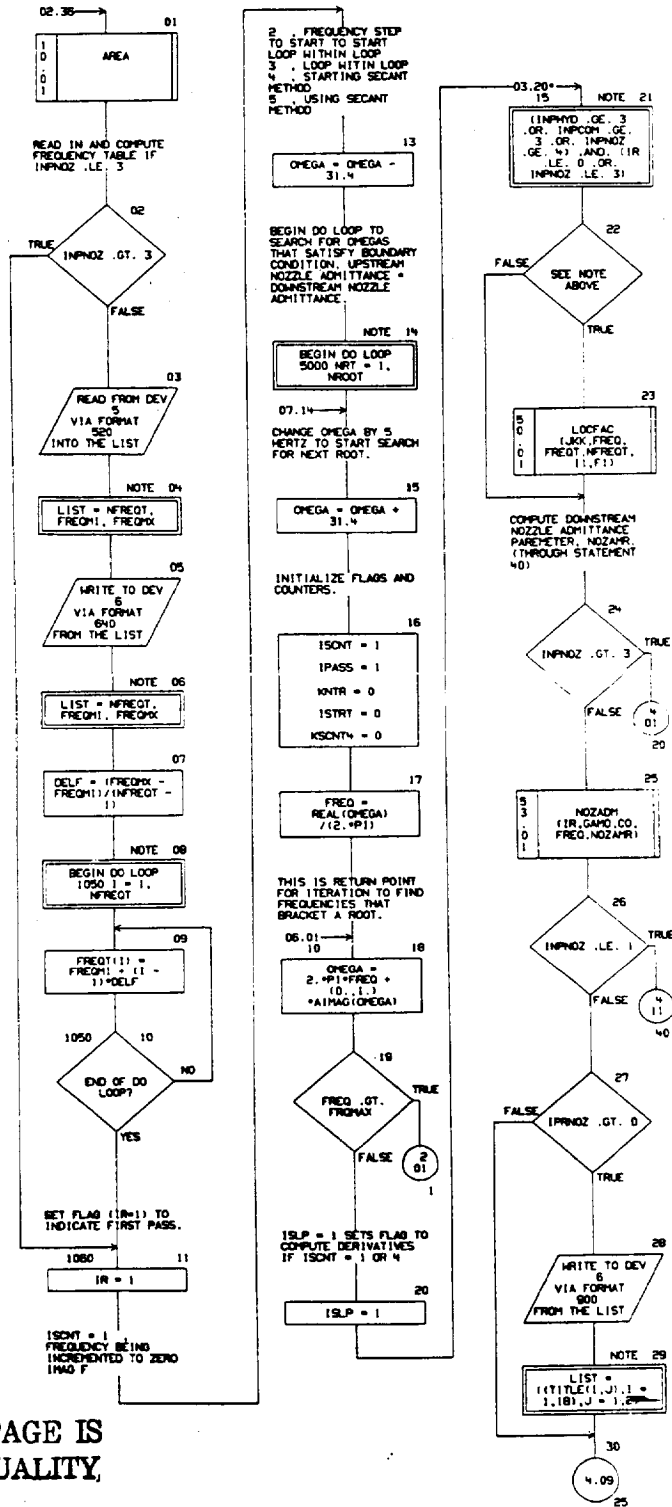


CHART TITLE - PROCEDURES



ORIGINAL PAGE IS OF POOR QUALITY.

CHART TITLE - PROCEDURES

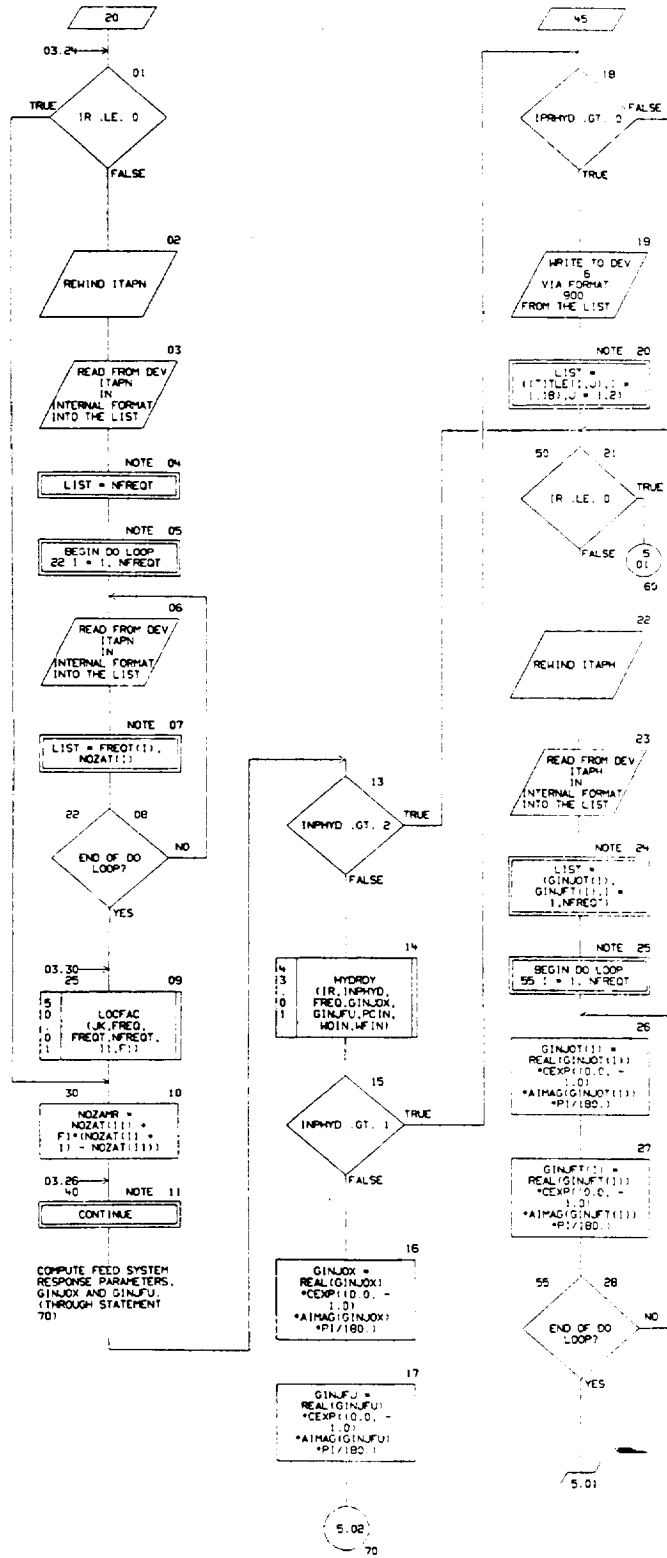


CHART TITLE - PROCEDURES

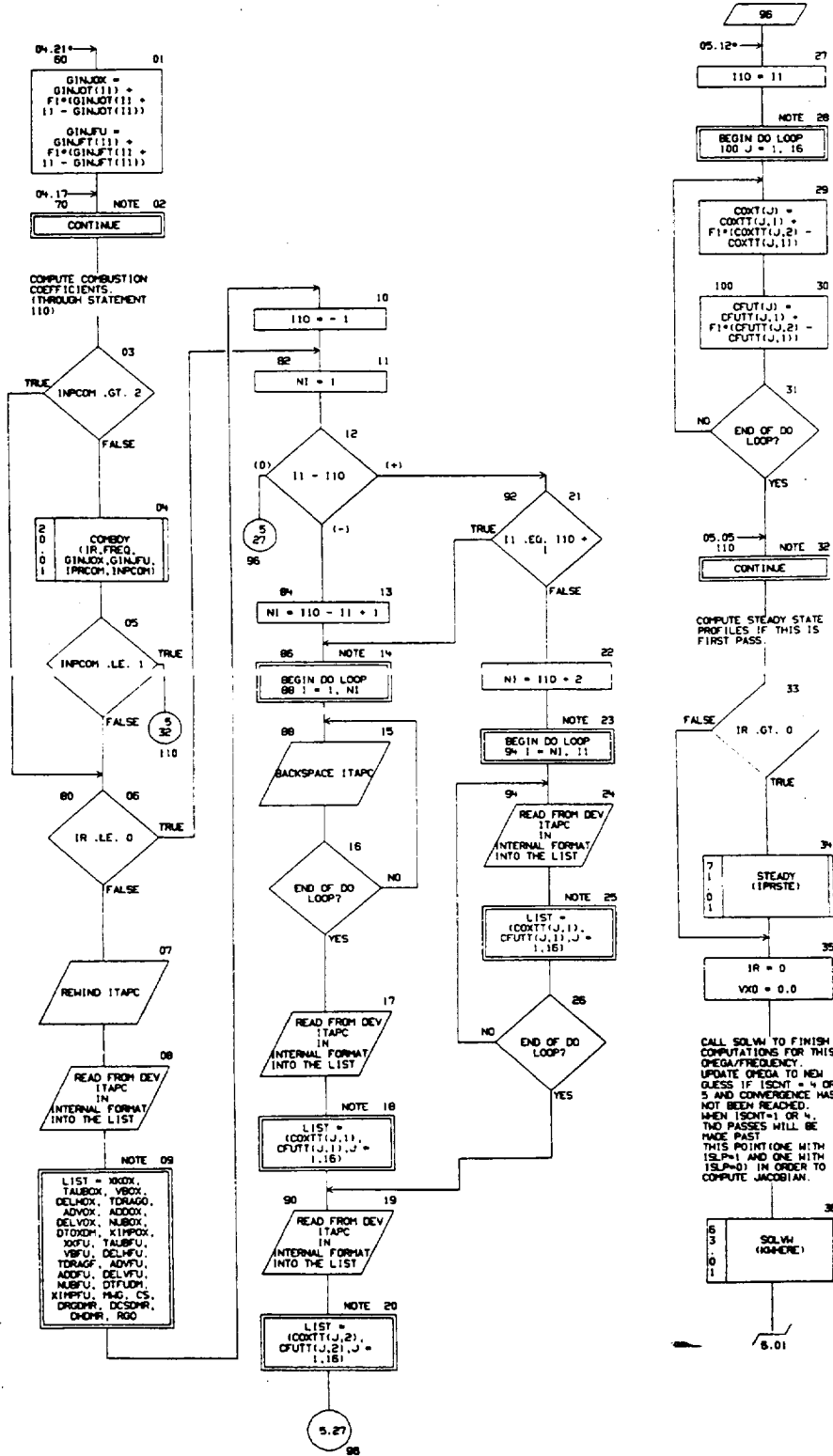


CHART TITLE - PROCEDURES

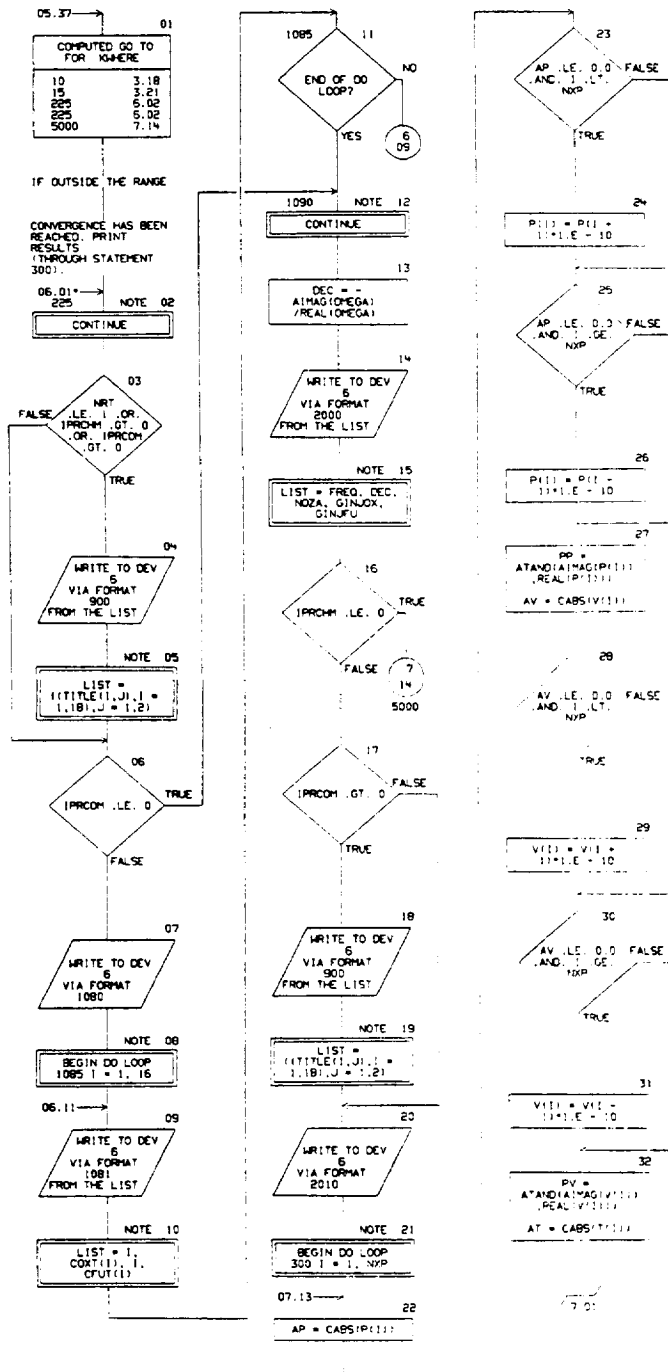
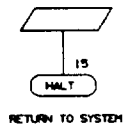
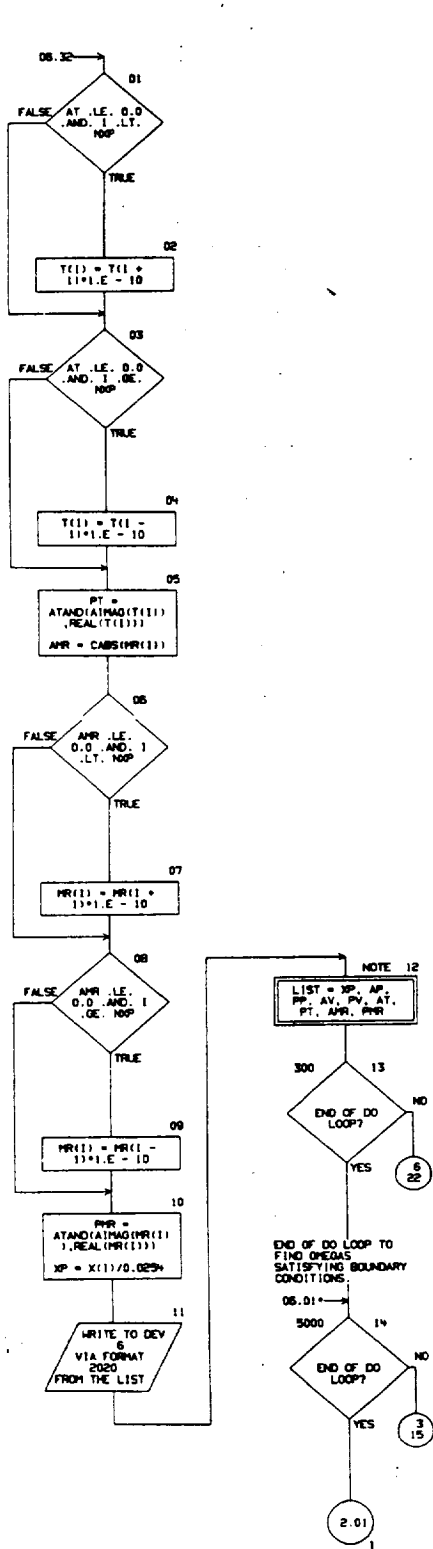




CHART TITLE - PROCEDURES



ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

```

DIMENSION TITLE(16,2)
COMPLEX OMEGA, P, RHO, V, MR, T, CHOZA, VXO,
  COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
  COX10, COX11, COX12, COX13, COX14, COX15, COX16,
  CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
  CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16,
  NOZA, GINJOX, GINJFU,
  NOZAT, GINJOT, GINJFT,
  FN, NOZHR, GN,
  COXT(16), CFUT(16),
  COXT(16,2), CFUT(16,2)
REAL HBOX1, HBFU1, HRB, HRG1, MG1, NUBOX, NUBFU, KF, KO, L, MAG
COMMON /CONCHM/ XBOX, XBFU, HBOX1, HBFU1, TALBOX, TALBFU, VBOX,
  VBFU, GAMO, RGO, DELHX, DELHFU, PC, CO,
  COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
  COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
  CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
  CFU12, CFU13, CFU14, CFU15, CFU16, MAG, XHFFU, XHPOX,
  CS, DCSHR, DCHR, DRGHR, ADVOX, ADDOX, TORAGO, DELVOX,
  NUBOX, DTOXH, ADVFU, ADDFU, TORAGF, DELVFU, NUBFU, DTFUH
EQUIVALENCE (COXT(1),COX1), (CFUT(1),CFU1)
COMMON /CONSTS/ HRB(100), TB(100), RHO(100), VB(100),
  DRHB(100), DRHB(100), DVB(100), VAPBOX(100), VAPBFU(100),
  SSV1(100), SSV2(100), SSV3(100), SSV4(100),
  SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9(100),
  SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),
  SSV14(100), SSV15(100), SSV16(100),
  RHOG1, VG1, HRG1, MG1
COMMON /CONCHV/ P(100), RHO(100), V(100), MR(100), T(100),
  VXO, OMEGA, CHOZA, DELP
COMMON /CONHRE/ NXP, XI(100), XM(100), A(100), OA(100), DELX,
  XO, XNOZ, AINU
COMMON /CONHZZ/ RCX, RCTX, ANGLEX, ORR, RINU, INPHOZ, FREQOX,
  FREQHI, IPRHOZ
COMMON /FZERO/ NOZA,NOZHR,GN,FN,FNR,FNI,HN,ISONT,ISLP
COMMON /DUMP/ IHRT
DATA JK/1/, JOK/1/, P1/3.141593/
COMMON /SOLVE/ FREQ, DELFRO, DELHX, EPSF, EPSX, EPSFS, EPSXS,
  FROMAX, CTEST, IPASS, KNTR, ISTRT, KSCNTY, IHSKP,
  IONTHX, IONTRX, IONTRX
COMMON /CONHAP/ NFREQ, FREQ(100), NOZAT(100), GINJOT(100),
  GINJFT(100), ITAPN, ITAPC, ITAPH
COMMON /HY/ ICRT,IRFLAG,ITERM,ITYPE,IPRHYD,IX,AA(30),CH(30),KF,KO,
  L(30),R(30),RHOL(30),VV(30),VF,VO,VOLF,VOLO,ZF,ZO,OSAVE(100),
  FSAVE(100)
490 FORMAT(18A1)
900 FORMAT(1H1,///,26X,'FEED SYSTEM COUPLED STABILITY MODEL',/,
  5X,18A1,/,5X,18A1)
500 FORMAT(12I6)
600 FORMAT(/,5X,'INPHYD' =/,12,5X,'INPCOH' =/,12,5X,'INPHOZ' =/,12,
  5X,'ITAPN' =/,13,5X,'ITAPC' =/,13,5X,'ITAPH' =/,13,/,5X,
  'IPRHYD' =/,12,5X,'IPRCHM' =/,12,5X,'IPRHOZ' =/,12,5X,
  'IPRCHH' =/,12,5X,'IPRSTE' =/,12,5X,'NXP' =/,14)
510 FORMAT(6E12.8)
610 FORMAT(/,5X,'XO' =/,1PE11.4,7X,'XNOZ' =/,E11.4,5X,'RINU' =/,
  E11.4,5X,'GAMO' =/,E11.4,/,27X,'CO' =/,E11.4,7X,'DELP' =/,
  E11.4)
620 FORMAT(/,5X,'NR00T' =/,13,5X,'IHRT' =/,12,7X,'IHSKP' =/,12,6X,
  'IONTHX' =/,14,4X,'IONTRX' =/,14,3X,'IONTRX' =/,14)
630 FORMAT(/,5X,'OMEGA(R)' =/,1PE11.4,2X,'OMEGA(I)' =/,E11.4,2X,
  'FROMAX' =/,E11.4,3X,'DELFRO' =/,E11.4,/,27X,'DELHX' =/,
  E11.4,4X,'CTEST' =/,E11.4,/,5X,'EPSF' =/,E11.4,5X,
  'EPSX' =/,E11.4,5X,'EPSFS' =/,E11.4,4X,'EPSXS' =/,E11.4)

```

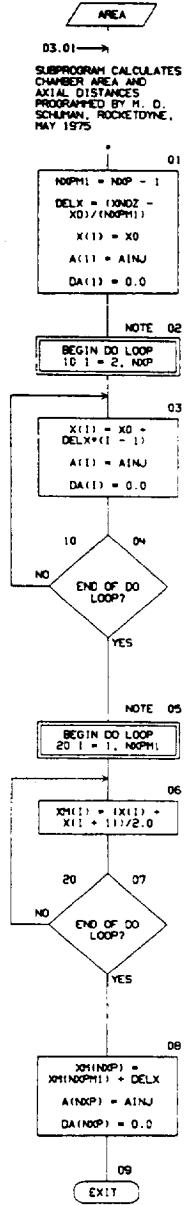
ORIGINAL PAGE IS OF POOR QUALITY

## CHART TITLE - NON-PROCEDURAL STATEMENTS

```
650  FORMAT(//,5X,'PC =',1PE11.4,7X,'MBOX1 =',E11.4,4X,'MRFUI =',  
      E11.4)  
520  FORMAT(112,9E12.8)  
640  FORMAT(//,5X,'NFREQ =',14,10X,'FREQH1 =',1PE11.4,3X,  
      'FREQH =',E11.4)  
1080 FORMAT(///,20X,'COMBUSTION DYNAMIC COEFFICIENTS',//)  
1081 FORMAT(5X,'COX(1,12,1) =',1PE11.4,',',E11.4,5X,'CFU(1,12,  
      '1) =',E11.4,',',E11.4)  
2000 FORMAT(///,10X,'FREQUENCY =',F8.2,' HZ',',',10X,  
      'DECORCH =',F8.5,///,  
      10X,'NOZZLE ADMITTANCE =',F8.5,',',2X,F8.5,///,10X,  
      'FEED SYSTEM RESPONSE',',',20X,'OXIDIZER =',F8.5,',',2X,F8.5,  
      ',',2X,'FUEL =',F8.5,',',2X,F8.5,///)  
2010 FORMAT(10X,4(11X,'OSCILLATORY'),',',5X,'DISTANCE',6X,  
      'PRESSURE RATIO',7X,'VELOCITY RATIO',6X,  
      'TEMPERATURE RATIO',7X,'MIXTURE RATIO',',',5X,'(INCHES)',  
      4(5X,'AMPLITUDE PHASE'),//)  
2020 FORMAT(5X,F8.4,4(5X,F8.5,2X,F7.2))
```

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE AREA



08/25/75

AUTOFLON CHART SET - FSCSH COMPUTER PROGRAM

PAGE 11

CHART TITLE - NON-PROCEDURAL STATEMENTS

COMMON /COMMON/ AOP, X(100), XH(100), A(100), DA(100), DELX,  
XD, XHQZ, AINU

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE CHANDY

CHANDY  
 SUBPROGRAM CALCULATES THE OSCILLATORY PRESSURE, TEMPERATURE, VELOCITY, MIXTURE RATIO, AND DENSITY PROFILES PLUS THE UPSTREAM NOZZLE ADMITTANCE FROM THE CHAMBER DYNAMIC EQUATIONS PROGRAMMED BY K. M. FERTIG AND M. D. SCHUMAN, ROCKWELL, MAY 1975

01  
 NMPH = NMP - 1  
 COMPUTE BOUNDARY CONDITIONS AT X=XD.

02  
 RHOINJ = DELP/GAM0  
 DHE = CO\*(2\*RHOOI/(GAM0\*PC)  
 DHE2 = SORT(DHE)

03  
 DM1 = CEXP((0., -1., -1.)  
 \*OMEGA/CO\*DHE2 - VG1/CO\*XD)  
 DMS = DHE2\*OMEGA\*XD/CO

04  
 P(1) = DM1\*(DELP\*CCOS(DHS) + (0., 1.)  
 \*GAM0\*DHE2\*VX0\*CSIN(DHS))  
 RHO(1) = P(1)/GAM0

05  
 V(1) = DM1\*((0., 1.)  
 \*DELP/(GAM0\*DHE2) \*CSIN(DHS) + VX0\*CCOS(DHS))  
 DM1 = CEXP((0., 1.)  
 \*OMEGA\*XD/VBOX)

06  
 DMS = CEXP((0., 1.)  
 \*OMEGA\*XD/VBFU)

07  
 DMH = DELP\*(COI\*DM1 - CFU1\*DHS) + P(1)\*COI2 - CFU2 + DELP/GAM0\*(COI3\*DM1 - CFU3\*DHS) + RHO(1)\*COI4 - CFU4 + VX0\*(COI7\*DM1 - CFU7\*DHS) + V(1)\*COI8 - CFU8)

08  
 DHE = (HRG1 + 1.)  
 \*HRG1\*HRFU/(A(1) \*HRG1\*CO)  
 DMH = DHE\*DMH - V(1)\*HRG1\*(HRG1 + 1.)\*(1. - HRG1)\*HRFU/VBOX1)/(HRG1 + 2.)

09  
 DMS = DMH\*(COI5\*DM1 - CFU5\*DHS - CFU6) + HRFU\*(HRG1 + 1.)/(A(1))  
 \*HRG1\*CO) - (0., 1.)  
 \*OMEGA\*TAUBFU\*HRFU/CO - DMS

10  
 HR(1) = DMH/DMS  
 T(1) = P(1) - RHO(1)  
 \*DRGDR/(RGO\*(1. + DRGDM\*(HRG1 + HRG1)\*HRFU)/(RGO) )

INITIALIZE INTEGRALS USED IN VAPORIZATION EXPRESSION.  
 11  
 PINTOX = (0., 0.)  
 RHINTOX = (0., 0.)  
 HRINTOX = (0., 0.)  
 VINTOX = (0., 0.)

12  
 PINTFU = (0., 0.)  
 RHINTFU = (0., 0.)  
 HRINTFU = (0., 0.)  
 VINTFU = (0., 0.)

13  
 II = (0., 1.)

14  
 I = 1

15  
 IHRT .GT. 0  
 FALSE  
 TRUE

16  
 WRITE TO DEV 6 VIA FORMAT 9000 FROM THE LIST

NOTE 17  
 LIST = OMEGA, I, P(1), RHO(1), HR(1), V(1), T(1)

NOTE 18  
 CONTINUE  
 CHANDY COMPUTES MOST OF THE VARIABLES IN /ADARND/

20  
 CHCOND (1)  
 21  
 DM1 = -11\*OMEGA\*DELP/CO

ENERGY EQUATION  
 22  
 AMA(1,1) = DM1 + 2.\*SSV1(1) + SSV7(1) - SSV2(1)\*SSV5(1) + G1OX + SSV6(1)\*G1FU)

23  
 AMA(1,2) = -SSV2(1)\*(SSV5(1) \*G2OX + SSV6(1)\*G2FU)  
 AMA(1,3) = 2.\*GAM0 + SSV8(1) - SSV2(1)\*(SSV5(1) \*G3OX + SSV6(1)\*G3FU)

24  
 AMA(1,4) = -SSV2(1)\*(SSV5(1) \*G3OX + SSV6(1)\*G3FU) - SSV9OX(1) + SSV9FU(1)

25  
 CHA(1) = SSV2(1)\*(SSV5(1) \*RBSOX + SSV6(1)\*RBSFU - SSV9OX(1)\*HR(1) + SSV9FU(1)\*HR(1)) - P(1)\*DM1 - 2.\*SSV1(1) + SSV7(1) - DELP\*(2.\*GAM0 + SSV8(1))

CONTINUITY EQUATION  
 26  
 AMA(2,1) = -SSV10(1)\*(G1OX + G1FU)  
 AMA(2,2) = DM1 + 2.\*SSV1(1) + SSV10(1)\*(G2OX + G2FU)

27  
 AMA(2,3) = 2.\*SSV1(1) + SSV10(1)\*(G3OX + G3FU)  
 AMA(2,4) = -SSV10(1)\*(G3OX + G3FU)

28  
 CHA(2) = SSV10(1)\*RBSOX + RBSFU - DM1\*HR(1) + 2.\*V(1) - V(1)\*SSV1(1) + 2.\*SSV1(1)\*HR(1) - RHO(1)\*SSV4(1)

MOMENTUM EQUATION

30  
 AMA(3,1) = SSV12(1)  
 AMA(3,2) = (0., 0.)  
 AMA(3,3) = DM1 + 2.\*SSV1(1) + SSV3(1)

31  
 AMA(3,4) = (0., 0.)  
 CHA(3) = V(1)\*(DM1 - SSV3(1) + 2.\*SSV1(1) + SSV12(1)\*P(1))

MIXTURE RATIO EQUATION  
 32  
 AMA(4,1) = -SSV13(1)\*(G1OX - HRB(1)\*G1FU)  
 AMA(4,2) = SSV14(1) - SSV13(1)\*(G2OX - HRB(1)\*G2FU)

33  
 AMA(4,3) = DELP\*(DMRB(1) - SSV13(1)\*G4OX - HRB(1)\*G4FU)  
 AMA(4,4) = DM1 + 2.\*SSV1(1) - SSV13(1)\*G3OX - HRB(1)\*G3FU - SSV15(1)

34  
 CHA(4) = SSV13(1)\*RBSOX - HRB(1)\*RBSFU - HR(1)\*(SSV15(1) + DM1) + 2.\*SSV1(1) - DELP\*(DMRB(1) + SSV15(1)) + V(1)

SOLVE 4 BY 4 SYSTEM OF COMPLEX EQUATIONS  
 35  
 2  
 3  
 1  
 2  
 COMMAT (AMA(4,4))

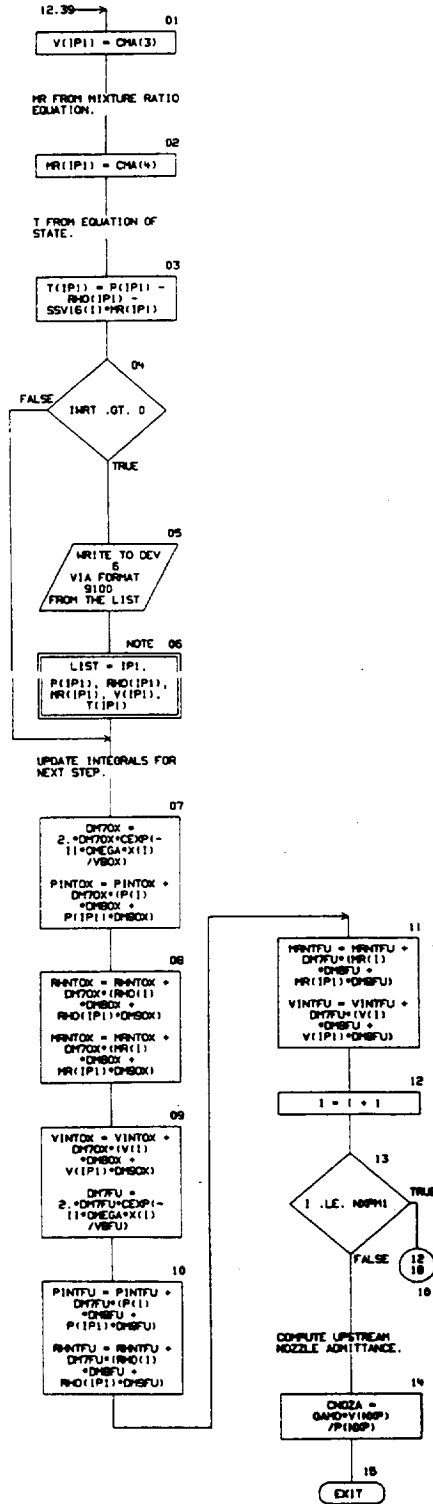
36  
 IP1 = I + 1

P FROM ENERGY EQUATION.  
 37  
 P(IP1) = CHA(1)

RHO FROM CONTINUITY EQUATION.  
 38  
 RHO(IP1) = CHA(2)

V FROM MOMENTUM EQUATION.  
 39  
 V(IP1) = CHA(3)

CHART TITLE - SUBROUTINE CHANDY



## CHART TITLE - NON-PROCEDURAL STATEMENTS

```

COMPLEX*16 ANA(4,5), CNA(4)
COMPLEX OMEGA, P, RHO, V, NR, T, CNOZA, YX0,
   COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
   COX10, COX11, COX12, COX13, COX14, COX15, COX16,
   CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
   CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
COMPLEX DH1, DH3, DH4, DHS, DHS, DHTOX, DHTFU, DHSOX,
   DHSFU, G1OX, G1FU, G2OX, G2FU, G3OX, G3FU, G4OX, G4FU,
   RBSOX, RBSFU, PINTOX, PINTFU, RHNTOX, RHNTFU, HRNTOX,
   HRNTFU, VINTOX, VINTFU, I1
   , DHSOX, DHSFU
REAL HBOX1, HBFU1, HRS, HRG1, HG1, HUBOX, HUBFU, HMG
COMMON /CONCH4/ XOX, XOFU, HBOX1, HBFU1, TAUOX, TAUOFU, VBOX,
   VBFU, G4O, GGO, DELFOX, DELHFU, PC, CO,
   COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
   COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
   CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
   CFU12, CFU13, CFU14, CFU15, CFU16, HMG, XIMFOX, XIMFU,
   CS, DCSOH, DCHO, DRGHR, ADVOX, ADDOX, TORAGO, DELVOX,
   HUBOX, DTOXH, ADVFU, ADDFU, TORAGF, DELVFU, HUBFU, DTFUO
COMMON /CONST/ HRB(100), TB(100), RHOB(100), VB(100),
   DHRB(100), DRHO(100), DVB(100), VAPBO(100), VAPBFU(100),
   SSV1(100), SSV2(100), SSV3(100), SSV4(100),
   SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9OX(100),
   SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),
   SSV14(100), SSV15(100), SSV16(100),
   RHOG1, VG1, HRG1, HG1
COMMON /CONCH4/ P(100), RHO(100), V(100), NR(100), T(100),
   YX0, OMEGA, CNOZA, DELP
COMMON /CONARE/ NXP, X(100), XH(100), A(100), DA(100), DELX,
   XO, XNOZ, AINU
COMMON /ADARN0/ G1OX, G2OX, G3OX, G4OX, G1FU, G2FU, G3FU, G4FU, PINTOX,
   RHNTOX, HRNTOX, VINTOX, PINTFU, RHNTFU, HRNTFU, VINTFU, RBSOX,
   RBSFU, DH1, DH3, DH4, DHS, DHS, DHTOX, DHSOX, DHSFU,
   DHSFU, DHSFU, I1, DME, DME2, RHO1NJ
COMMON /DUMP/ IHRT
EQUIVALENCE (ANA(1:5), CNA(1:5))
9000 FORMAT(1, OMEGA = ', IPE13.5//3X, '1', 11X, 'P', 19X, RHO',
   20X, 'NR', 20X, 'V', 21X, 'T//1X, 13, IPI0E(11, 4)
9100 FORMAT(1X, 13, IPI0E(11, 4)

```



CHART TITLE - SUBROUTINE CHANDY(11)

CHANDY

12.20

ROUTINE TO CALCULATE  
PARAMETERS FOR  
SUBPROGRAM CHANDY  
PROGRAMMED BY K. H.  
FERTIG, ROCKWELL,  
MAY 1975

10  
D1 =  
CEXP(11)\*OMEGA\*  
(XN(11))/VBOX  
D2 = (XN(11) -  
X01)/(TAUBOX\*VBOX)  
D3 =  
CEXP(11)\*OMEGA\*  
DELX/VBOX

02  
D4 =  
CEXP(11)\*OMEGA\*  
(11/VBOX)\*(1. +  
D3)/2.

CALCULATE OXIDIZER  
VAPORIZATION  
PARAMETERS.

03  
D7OX =  
VBOX/(2.\*OMEGA\*  
2\*DELX+TAUBOX)  
D8OX = 1. -  
11\*OMEGA\*DELX/  
VBOX - 1./D7OX

04  
D9OX = 1. +  
11\*OMEGA\*DELX/  
VBOX - D8OX  
D4 =  
D7OX\*D8OX\*D8OX  
D5 = D7OX\*D9OX

05  
D6OX = D8OX/D5

06  
RBSOX =  
DELX\*D1\*(COX1 +  
COX9\*D2) +  
RHO(1)\*D1\*(COX3 +  
COX11\*D2) +  
HR(1)\*D1\*(COX5 +  
COX13\*D2) +  
VKD\*D1\*(COX7 +  
COX15\*D2)

07  
RBSOX = RBSOX +  
D4\*(COX10 +  
PINTOX +  
COX12\*HRINTOX +  
COX14\*HRINTOX +  
COX16\*VINTOX)

08  
RBSOX = RBSOX +  
P(1)\*(COX2/2. +  
COX10\*D4) +  
RHO(1)\*(COX4/2. +  
COX12\*D4) +  
HR(1)\*(COX6/2. +  
COX14\*D4) +  
V(1)\*(COX8/2. +  
COX16\*D4)

09  
RBSOX =  
VAPBOX(11)\*RBSOX  
G1OX =  
VAPBOX(11) +  
COX2/2. +  
COX10\*D5  
G2OX =  
VAPBOX(11) +  
COX4/2. +  
COX12\*D5

10  
G3OX =  
VAPBOX(11) +  
(COX8/2. +  
COX14\*D5)  
D4OX =  
VAPBOX(11) +  
(COX6/2. +  
COX16\*D5)

11  
D1 =  
CEXP(11)\*OMEGA\*  
(XN(11))/VBFU  
D2 = (XN(11) -  
X01)/(TAUBFU\*VBFU)  
D3 =  
CEXP(11)\*OMEGA\*  
DELX/VBFU

12  
D4 =  
CEXP(11)\*OMEGA\*  
(11/VBFU)\*(1. +  
D3)/2.

CALCULATE FUEL  
VAPORIZATION  
PARAMETERS.

13  
D7FU =  
VBFU/(2.\*OMEGA\*  
2\*DELX+TAUBFU)  
D8FU = 1. -  
11\*OMEGA\*DELX/  
VBFU - 1./D7FU

14  
D9FU = 1. +  
11\*OMEGA\*DELX/  
VBFU - D8FU  
D4 =  
D7FU\*D8FU\*D8FU  
D5 = D7FU\*D9FU

15  
D6FU = D8FU/D5

16  
RBSFU =  
DELX\*D1\*(CFU1 +  
CFU9\*D2) +  
RHO(1)\*D1\*(CFU3 +  
CFU11\*D2) +  
HR(1)\*D1\*(CFU5 +  
CFU13\*D2) +  
VKD\*D1\*(CFU7 +  
CFU15\*D2)

17  
RBSFU = RBSFU +  
D4\*(CFU10 +  
PINTFU +  
CFU12\*HRINTFU +  
CFU14\*HRINTFU +  
CFU16\*VINTFU)

18  
RBSFU = RBSFU +  
P(1)\*(CFU2/2. +  
CFU10\*D4) +  
RHO(1)\*(CFU4/2. +  
CFU12\*D4) +  
HR(1)\*(CFU6/2. +  
CFU14\*D4) +  
V(1)\*(CFU8/2. +  
CFU16\*D4)

19  
RBSFU =  
VAPFU(11)\*RBSFU  
G1FU =  
VAPFU(11) +  
CFU2/2. +  
CFU10\*D5  
G2FU =  
VAPFU(11) +  
CFU4/2. +  
CFU12\*D5

19  
RBSFU =  
VAPFU(11)\*RBSFU  
G1FU =  
VAPFU(11) +  
CFU2/2. +  
CFU10\*D5  
G2FU =  
VAPFU(11) +  
CFU4/2. +  
CFU12\*D5

20  
G3FU =  
VAPFU(11) +  
CFU8/2. +  
CFU14\*D5  
D4FU =  
VAPFU(11) +  
CFU6/2. +  
CFU16\*D5

21  
EXIT

ORIGINAL PAGE IS  
OF POOR QUALITY

## CHART TITLE - NON-PROCEDURAL STATEMENTS

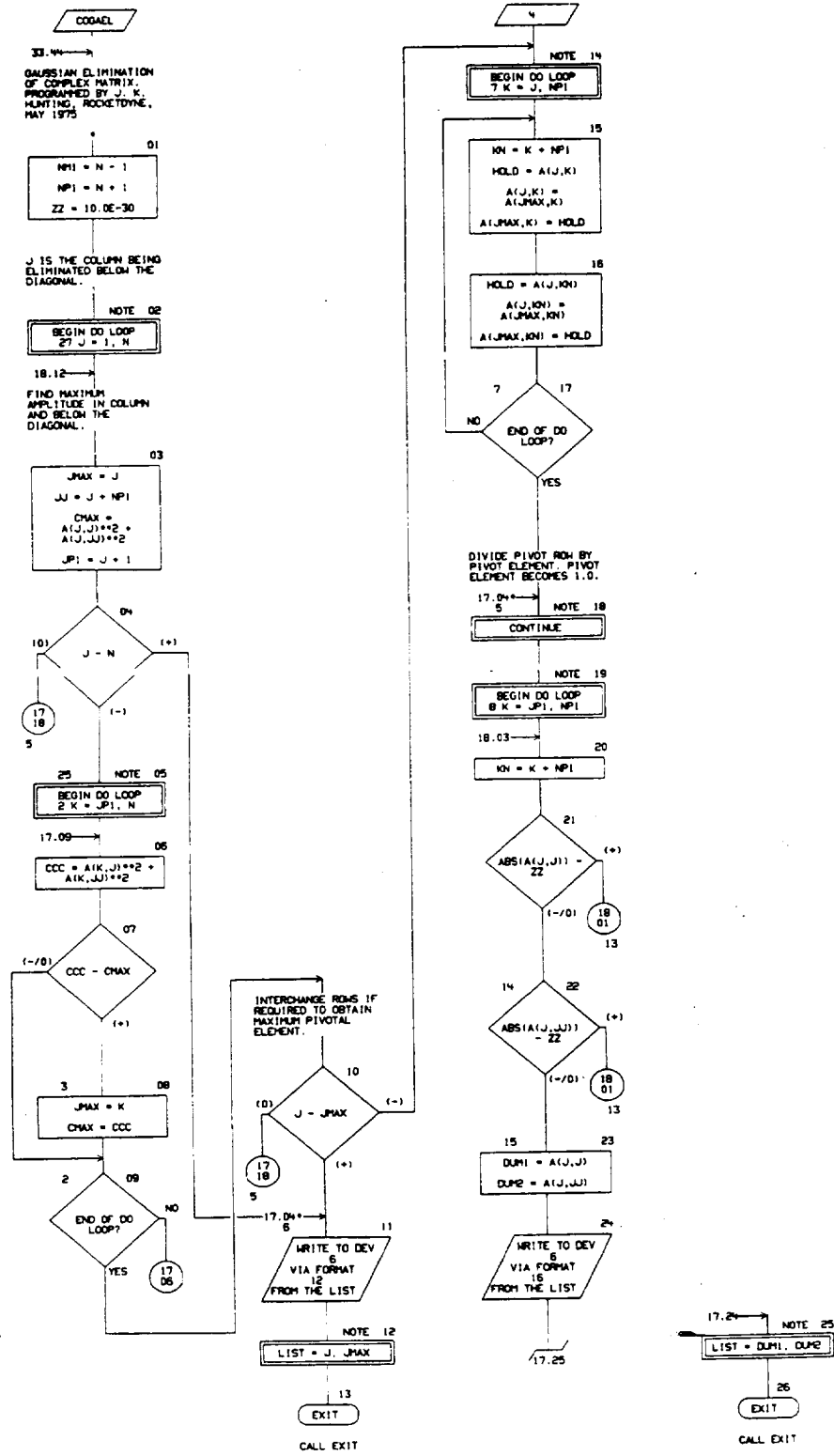
```

COMPLEX OMEGA, P, RHO, V, NR, T, CHOZA, VXO,
COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
COX10, COX11, COX12, COX13, COX14, COX15, COX16,
CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
COMPLEX DH1, DH3, DH4, DH5, DH6, DHTOX, DHTFU, DMSOX,
DMSFU, G1OX, G1FU, G2OX, G2FU, G3OX, G3FU, OHOX, OHFU,
RBSOX, RBSFU, PINTOX, PINTFU, RHNTOX, RHNTFU, HRNTOX,
HRNTFU, VINTOX, VINTFU, II
,DMSOX,DMSFU
REAL MBOX1, MEFU1, MRB, MRG1, HGI, NUBOX, NUBFU, HMG
COMMON /COMCSM/ XKOX, XKFU, MBOX1, MEFU1, TAUBOX, TAUBFU, YBOX,
YBFU, GANO, RGO, DELHGX, DELHFU, PC, CD,
COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
CFU12, CFU13, CFU14, CFU15, CFU16, HMG, XIHPFU, XIHPOK,
CS, DCSDHR, DHDHR, DRDHR, ADVOK, ADDOX, TORAGO, DELVOX,
NUBOX, DTOKDH, ADVFU, ADOFU, TORAGF, DELVFU, NUBFU, DTFUOH
COMMON /CONSTS/ MRB(100), TB(100), RHOB(100), VB(100),
DHRB(100), DRHOB(100), DVB(100), VAPBOX(100), VAPBFU(100),
SSV1(100), SSV2(100), SSV3(100), SSV4(100),
SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9OX(100),
SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),
SSV14(100), SSV15(100), SSV16(100),
RHOG1, VG1, MRG1, HGI
COMMON /COMCHM/ P(100), RHO(100), V(100), NR(100), T(100),
VXO, OMEGA, CHOZA, DELP
COMMON /COMHRE/ NXP, X(100), XH(100), A(100), DA(100), DELX,
XD, XNOZ, AINU
COMMON /ADARND/ G1OX,G2OX,G3OX,OHOX,G1FU,G2FU,G3FU,OHFU,PINTOX,
RHNTOX,HRNTOX,VINTOX,PINTFU,RHNTFU,HRNTFU,VINTFU,RBSOX,
RBSFU,DH1,DH3,DH4,DH5,DH6,DHTOX,DMSOX,DMSFU,DHTFU,
DMSFU,DMSFU,II,DH2,DH22,RHO1NU

```

ORIGINAL PAGE IS  
POOR QUALITY

CHART TITLE - SUBROUTINE COGACL(A,N)



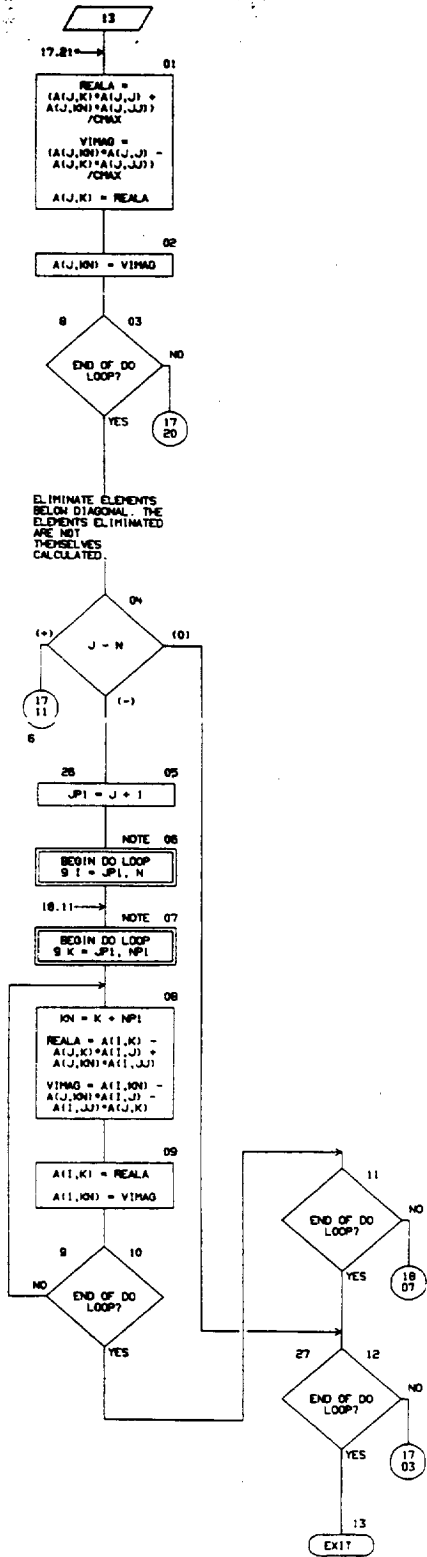


CHART TITLE - NON-PROCEDURAL STATEMENTS

DIMENSION A(62,126)  
12 FORMAT (61H)ERROR IN COGAEI. SUBROUTINE, J AND JMAX EQUAL, RESPECTI  
VELY. (21)2)  
16 FORMAT (61H)MATRIX IS SINGULAR. EXIT FROM COGAEI. THE PIVOT ELEMEN  
TS ARE (2E14.6)

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE COMBDY(IR,FREQ,GINJGX,GINJFU,INPCOH,INPCOH)

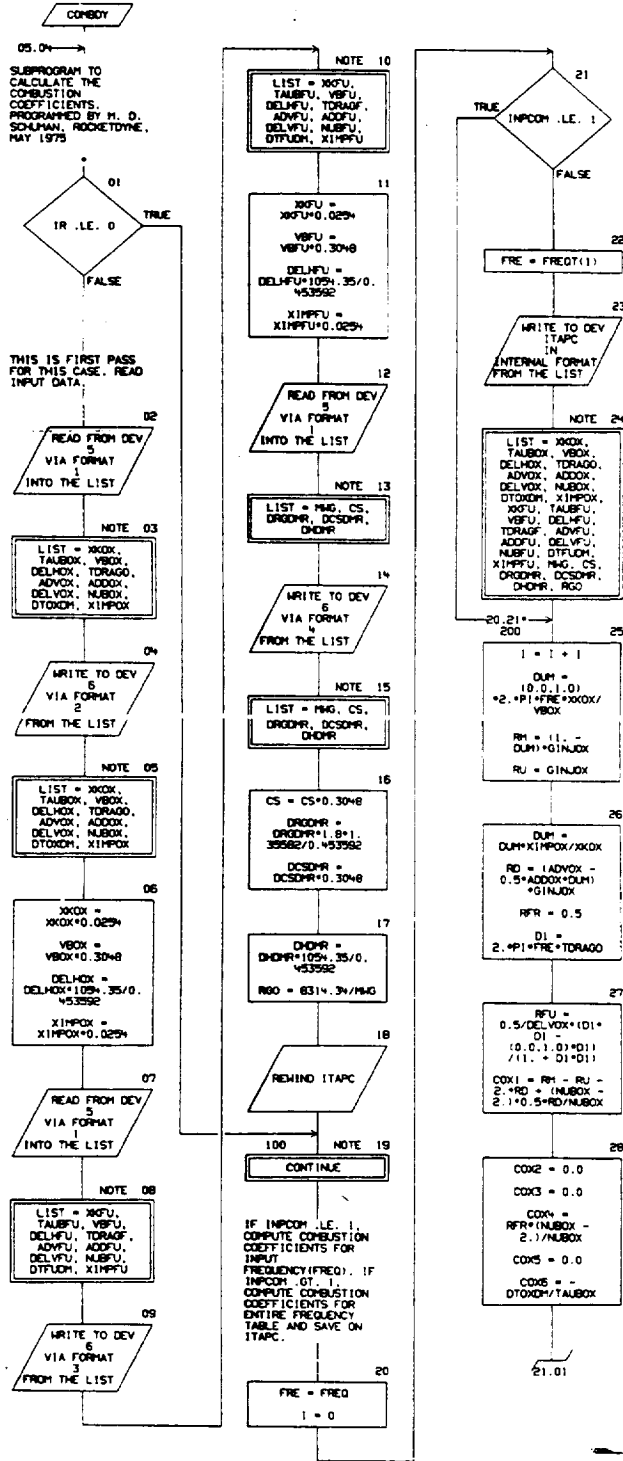
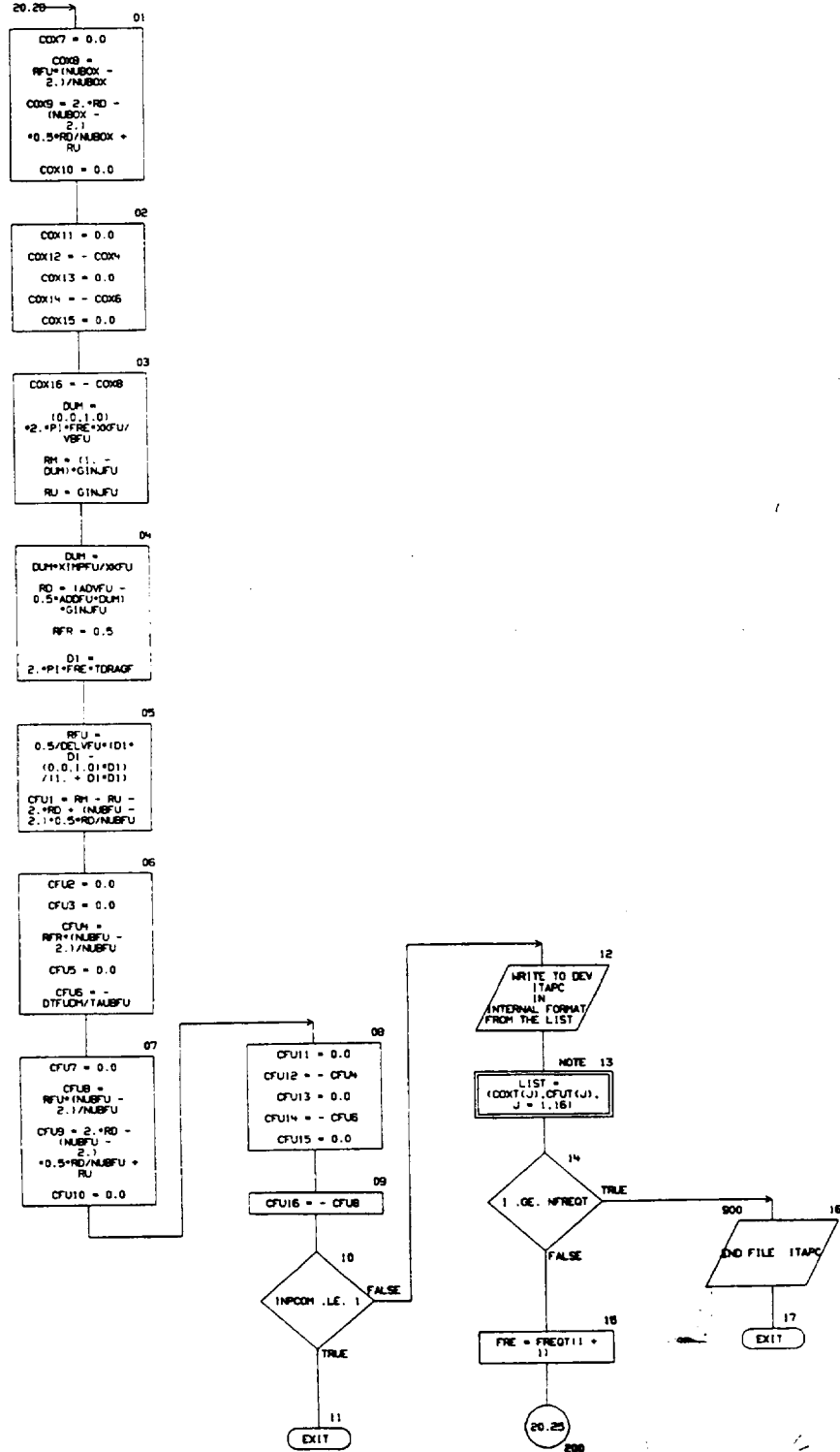


CHART TITLE - SUBROUTINE COMBOY(1R,FREQ,GINUX,GINUFU,IPROH,INPCOH)



## CHART TITLE - NON-PROCEDURAL STATEMENTS

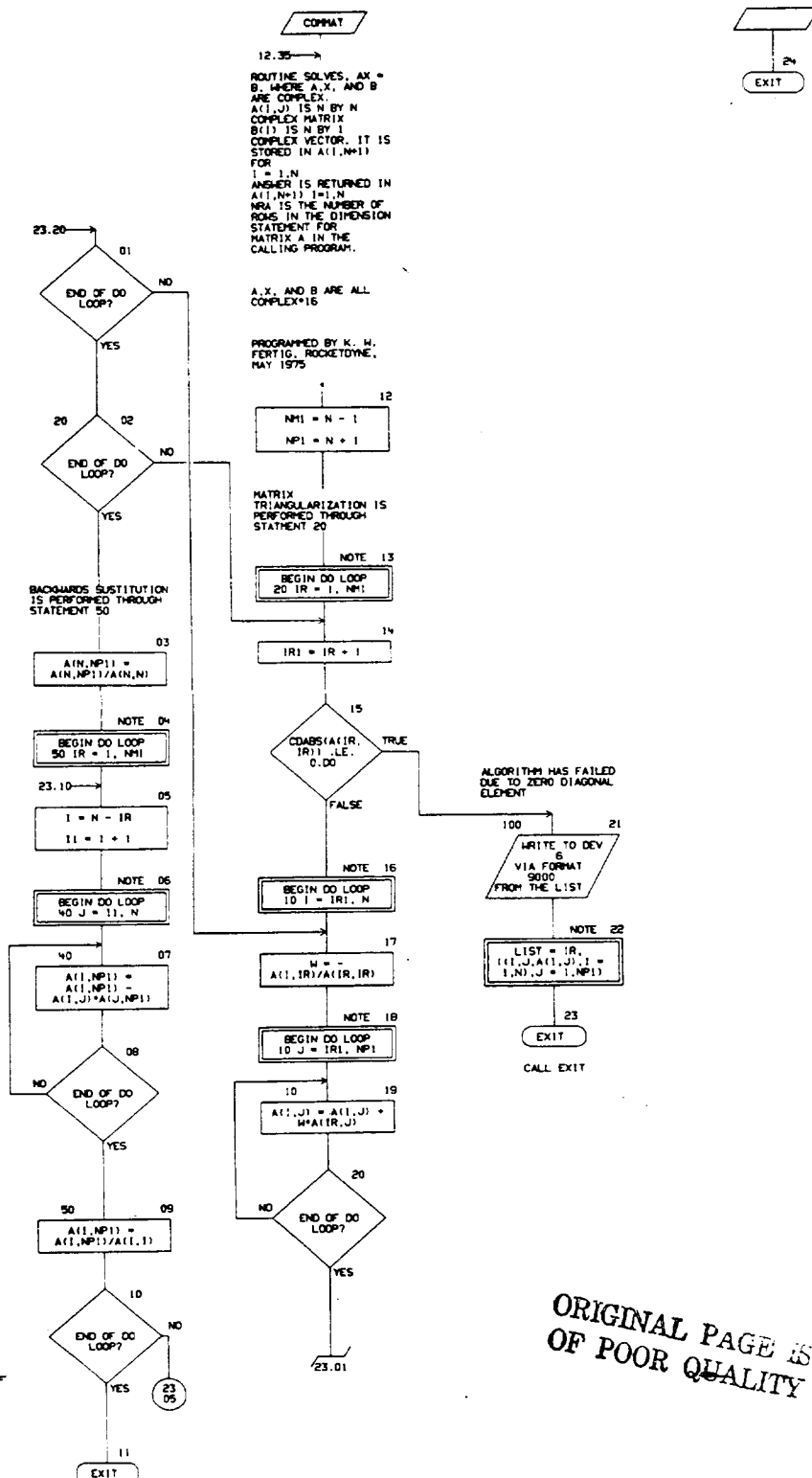
```

COMPLEX GINJGX, GINJFU, DUM, NOZAT, GINJOT, GINJFT,
      COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
      COX10, COX11, COX12, COX13, COX14, COX15, COX16,
      CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
      CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16,
      COXT(16), CFUT(16), RM, RU, RD, RFU
REAL MBOX1, MEFU1, MAG, NUBOX, NUBFU
COMMON /CONTAP/ NFREQT, FREQT(100), NOZAT(100), GINJOT(100),
      GINJFT(100), ITARN, ITAPC, ITAPH
COMMON /CONCHP/ XKOK, XKFU, MBOX1, MEFU1, TAUBOX, TAUBFU, VBOX,
      VBFU, GAMD, RGO, DELHOX, DELHFU, PC, CO,
      COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
      COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
      CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
      CFU12, CFU13, CFU14, CFU15, CFU16, MAG, XIMPFU, XIMPOX,
      CS, DCSDR, DCHDR, DRGDR, ADVOX, ADDOX, TORAGO, DELVOX,
      NUBOX, DTOXDM, ADVFU, ADFU, TORAGF, DELVFU, NUBFU, DTFUDM
EQUIVALENCE (COXT(1),COX1), (CFUT(1),CFU1)
DATA PI/3.14159/
1  FORMAT(BE12.8)
2  FORMAT(/5X,'XKOK' =,1PE11.4,5X,'TAUBOX' =,
      E11.4,3X,'VBOX' =,E11.4,5X,'DELHOX' =,E11.4,/,27X,
      'TORAGO' =,E11.4,3X,'ADVOX' =,E11.4,/,5X,'ADDOX' =,E11.4,
      4X,'DELVOX' =,E11.4,3X,'NUBOX' =,E11.4,4X,'DTOXDM' =,
      E11.4,/,27X,'XIMPOX' =,E11.4)
3  FORMAT(/5X,'XKFU' =,1PE11.4,5X,'TAUBFU' =,
      E11.4,3X,'VBFU' =,E11.4,5X,'DELHFU' =,E11.4,/,27X,
      'TORAGF' =,E11.4,3X,'ADVFU' =,E11.4,/,5X,'ADDFU' =,E11.4,
      4X,'DELVFU' =,E11.4,3X,'NUBFU' =,E11.4,4X,'DTFUDM' =,
      E11.4,/,27X,'XIMPFU' =,E11.4)
4  FORMAT(/5X,'MAG' =,1PE11.4,5X,'CS' =,E11.4,
      7X,'DRGDR' =,E11.4,3X,'DCSDR' =,E11.4,/,27X,'DCHDR' =,
      E11.4)

```



CHART TITLE - SUBROUTINE COMMAT(A,NRA,N)

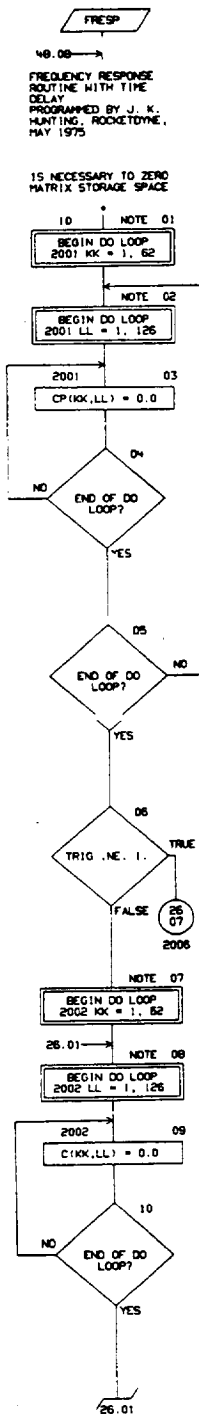


ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMPLD*16 AINRA,1),M
9000 FORMAT(// ' ***** DIVIDE CHECK IN COMBAT *****' /
           ' IR = ',I10/' MATRIX A(I,J) = ' /
           '(SX,216.1PE15.6, ' ',IPE13.6))
```

CHART TITLE - SUBROUTINE FRESP(CRT,WRITE,IX)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - SUBROUTINE FREQ(IORT, IWRITE, IX)

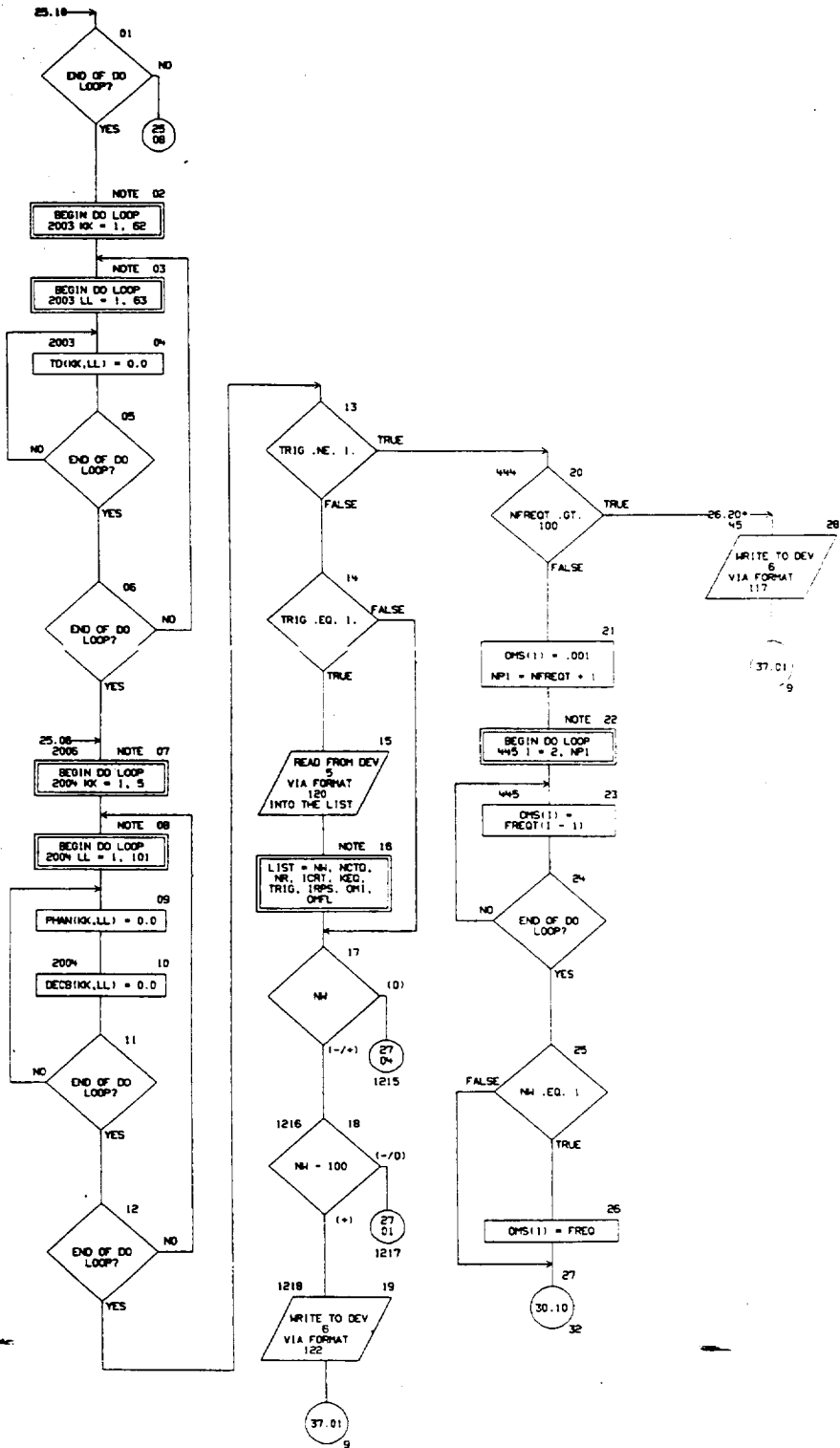


CHART TITLE - SUBROUTINE FRESP(CRT, IWRITE, IX)

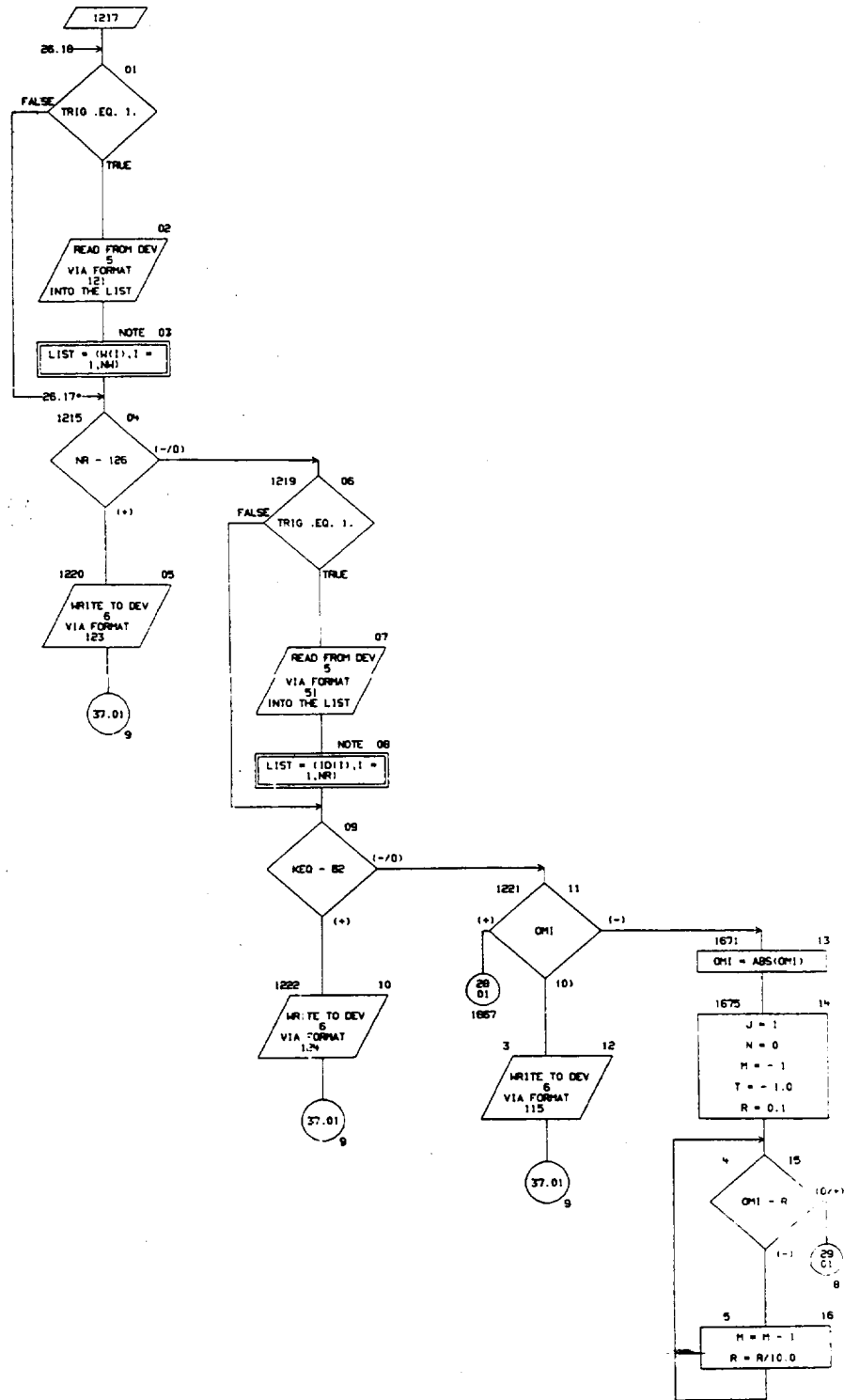


CHART TITLE - SUBROUTINE FRESP (ICRT, IWRITE, IX)

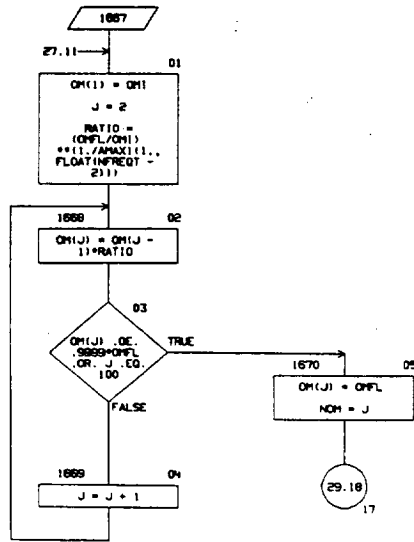
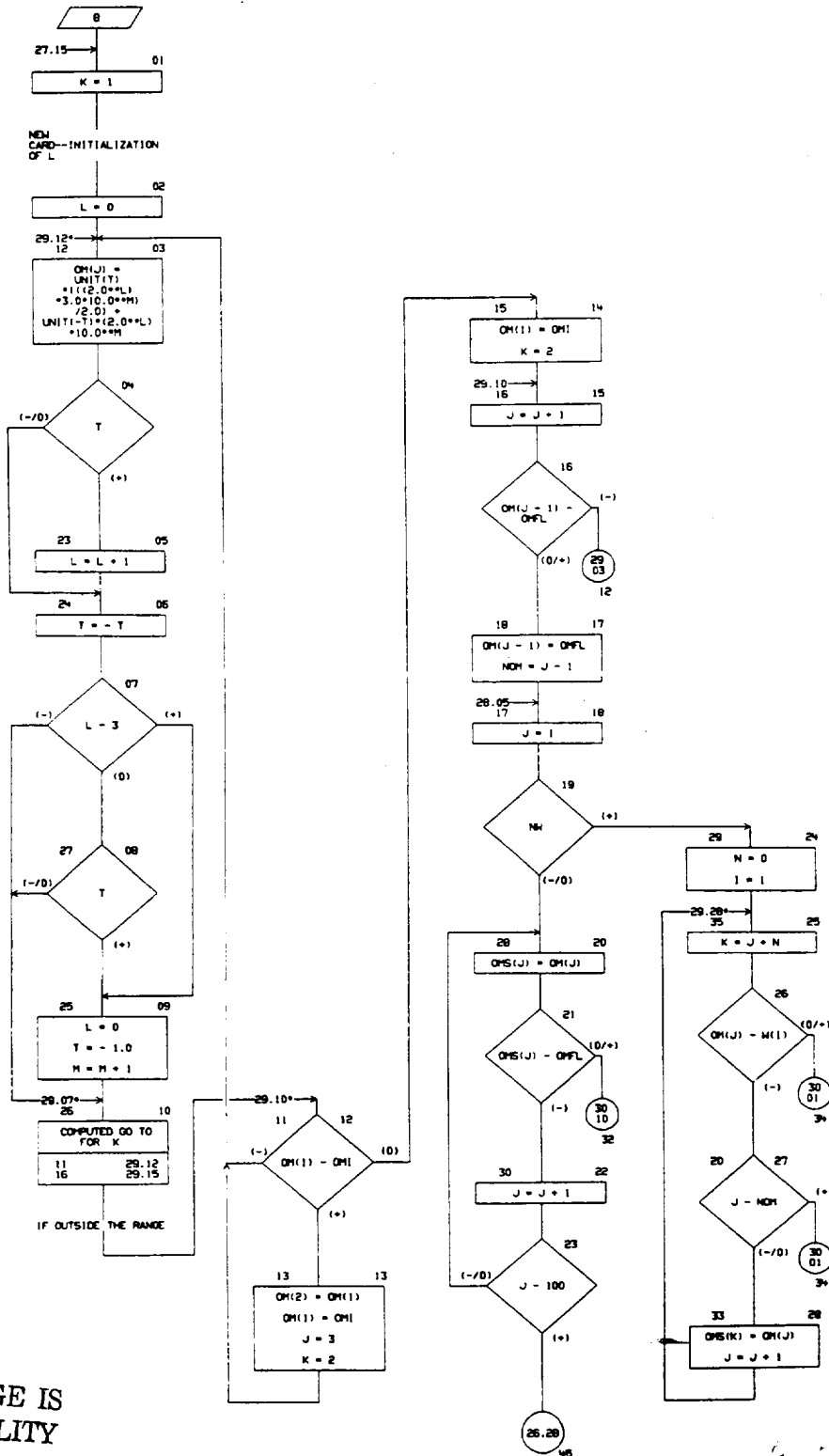


CHART TITLE - SUBROUTINE FRESP (ICRT, IWRITE, IX)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - SUBROUTINE FRESPI(ORT, IWRITE, IX)

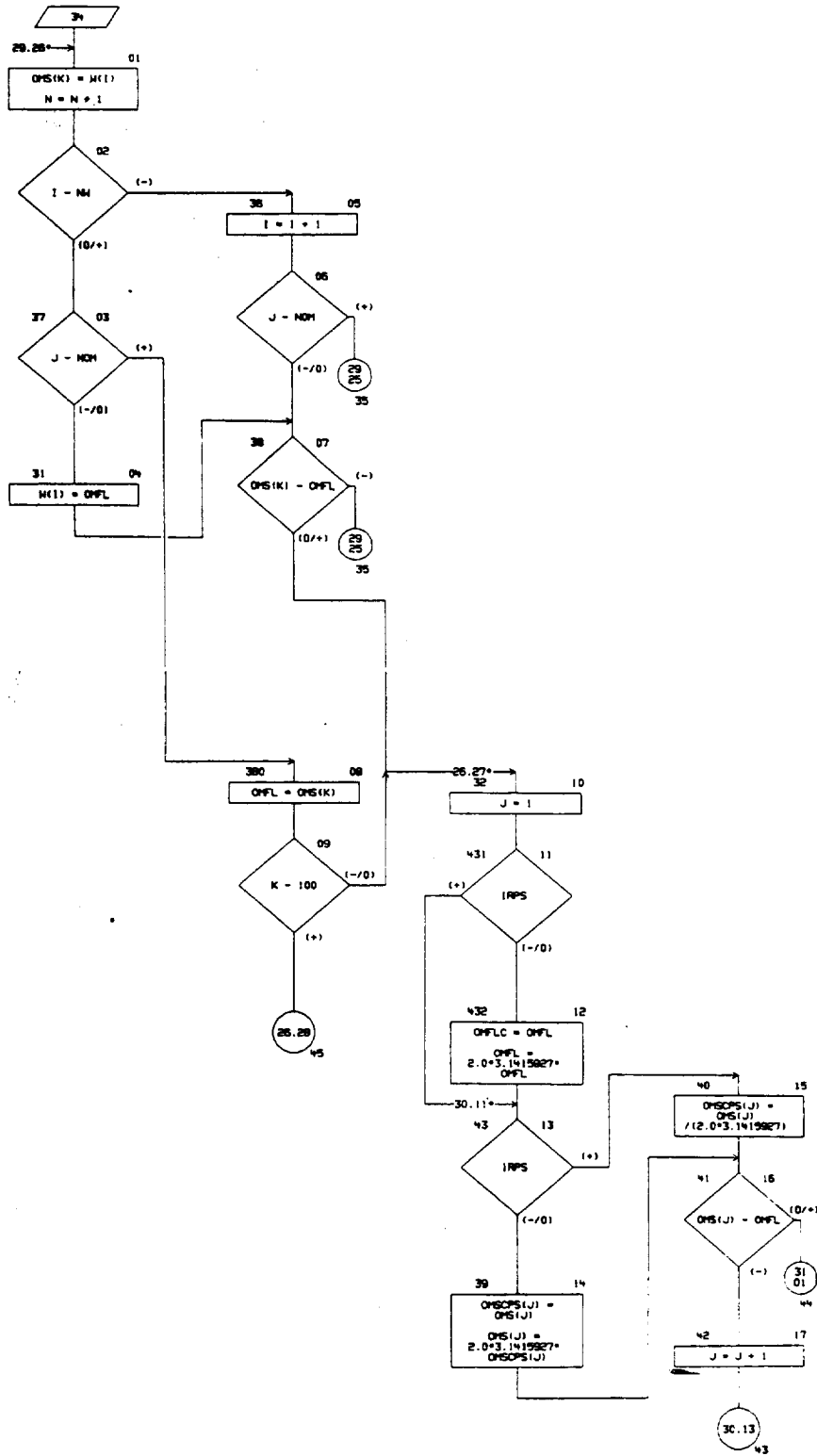




CHART TITLE - SUBROUTINE FRESP(ICRT, IWRITE, IX)

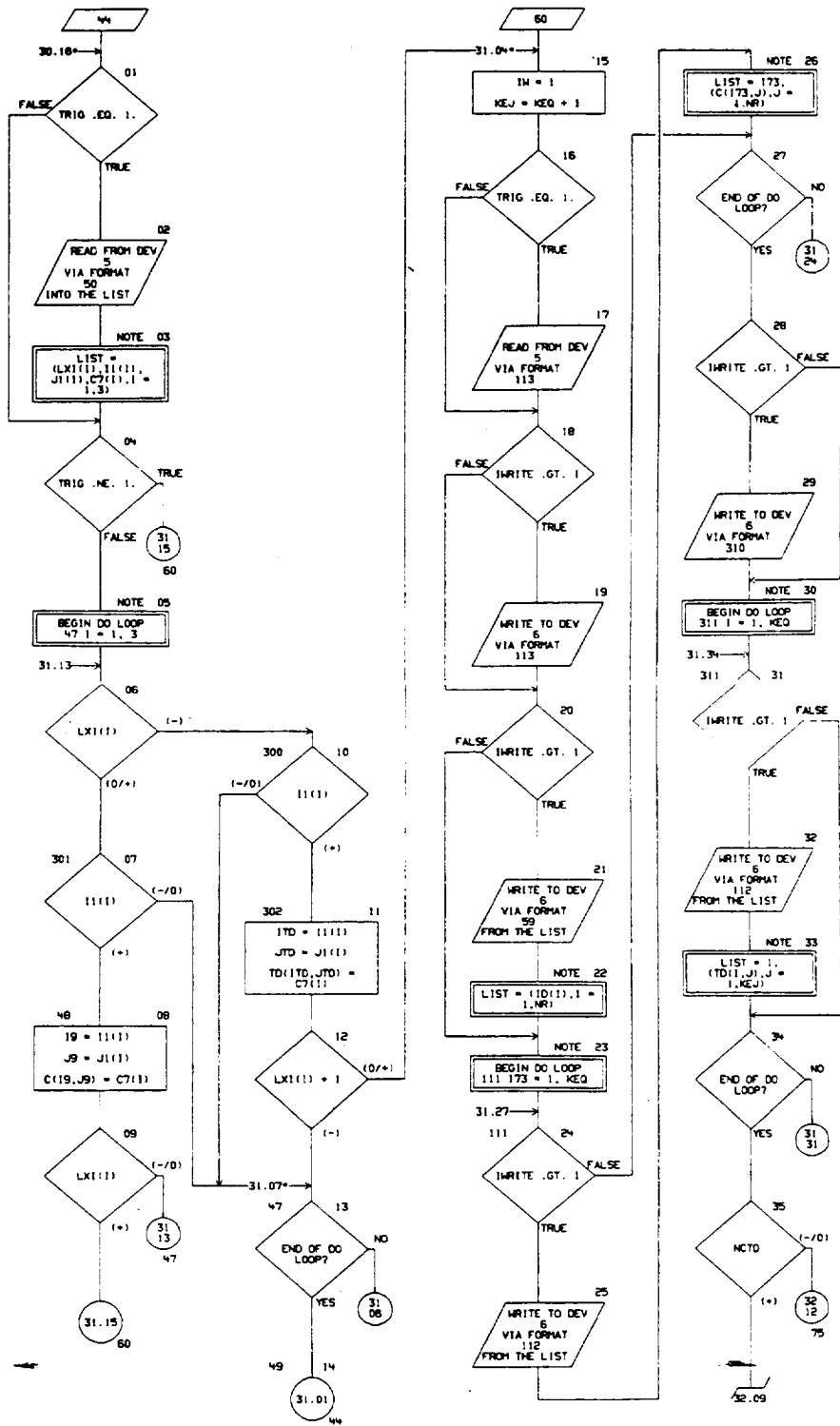


CHART TITLE - SUBROUTINE FRESF(ICRT, IWRITE, IX)

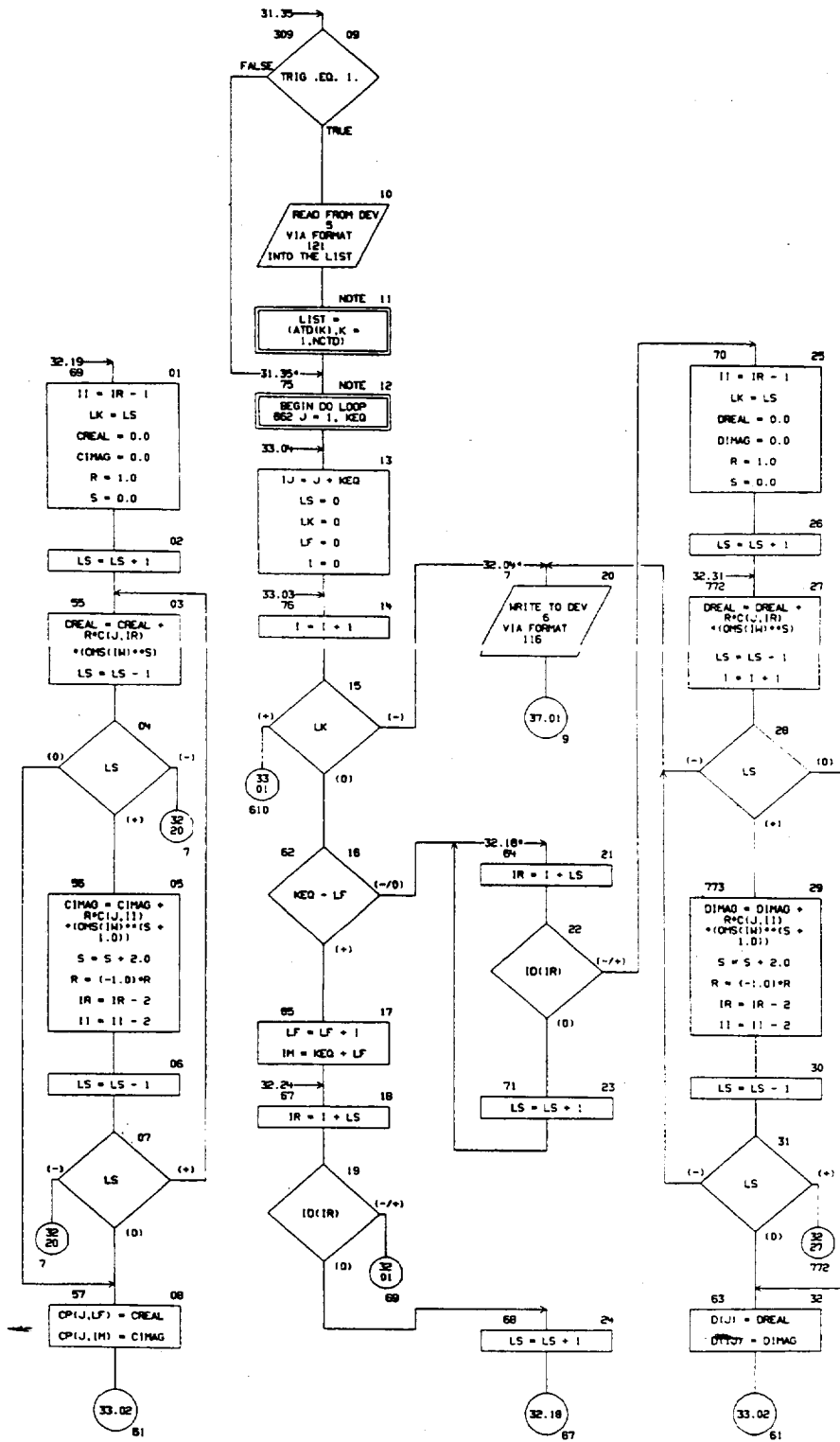
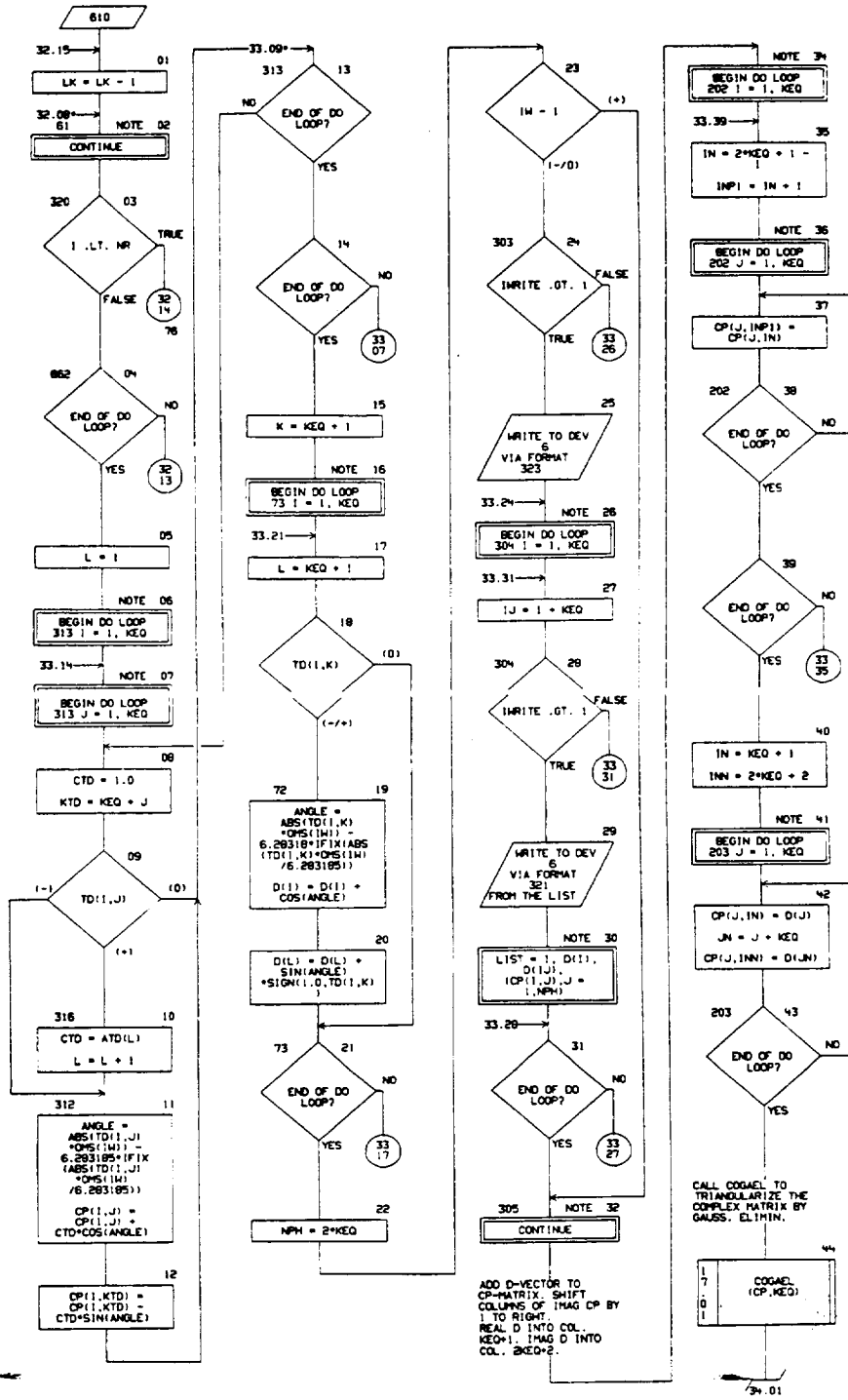


CHART TITLE - SUBROUTINE FRESP (ORT,WRITE,IX)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - SUBROUTINE FRESF(ICRT, IWRITE, IX)

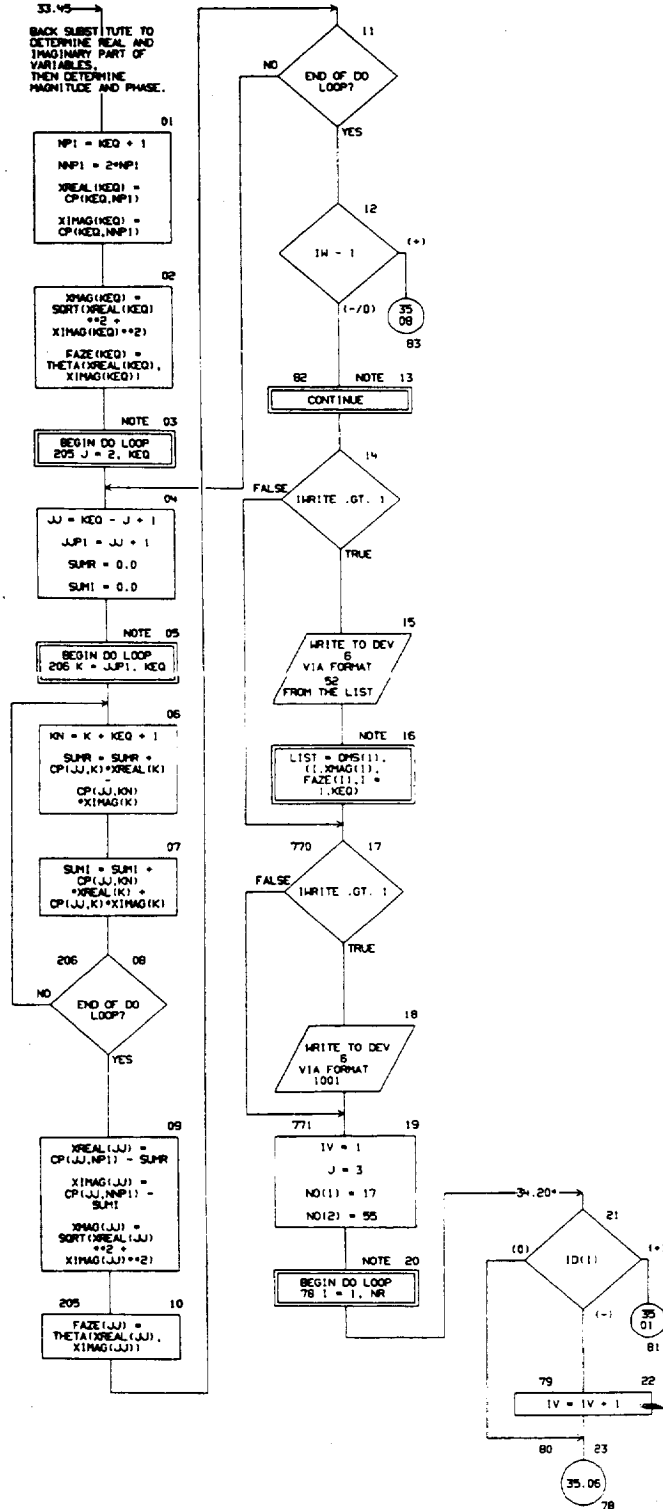


CHART TITLE - SUBROUTINE FRESF(I CRT, IWRITE, IX)

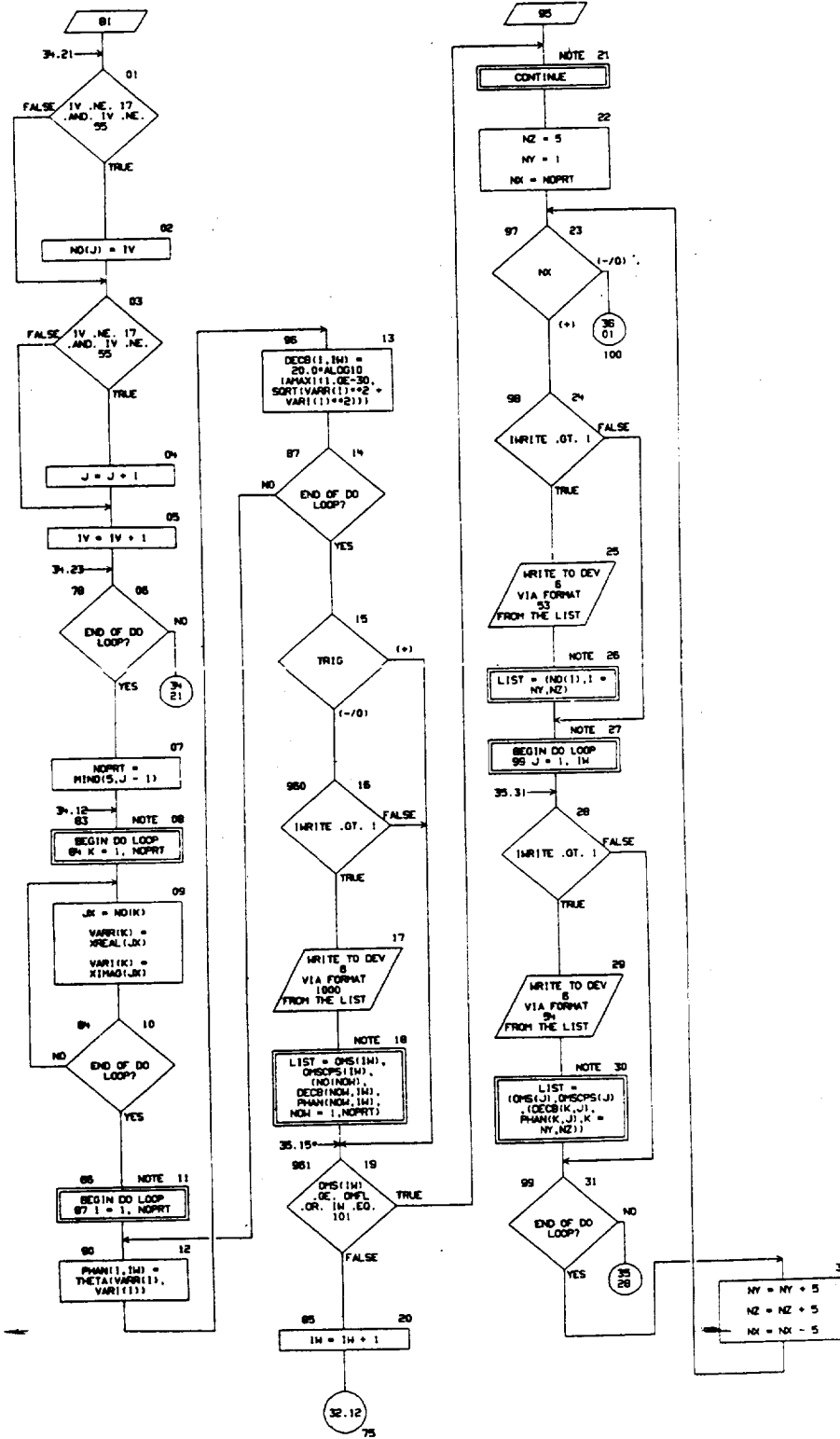


CHART TITLE - SUBROUTINE FRESP (ICRT, IWRITE, IX)

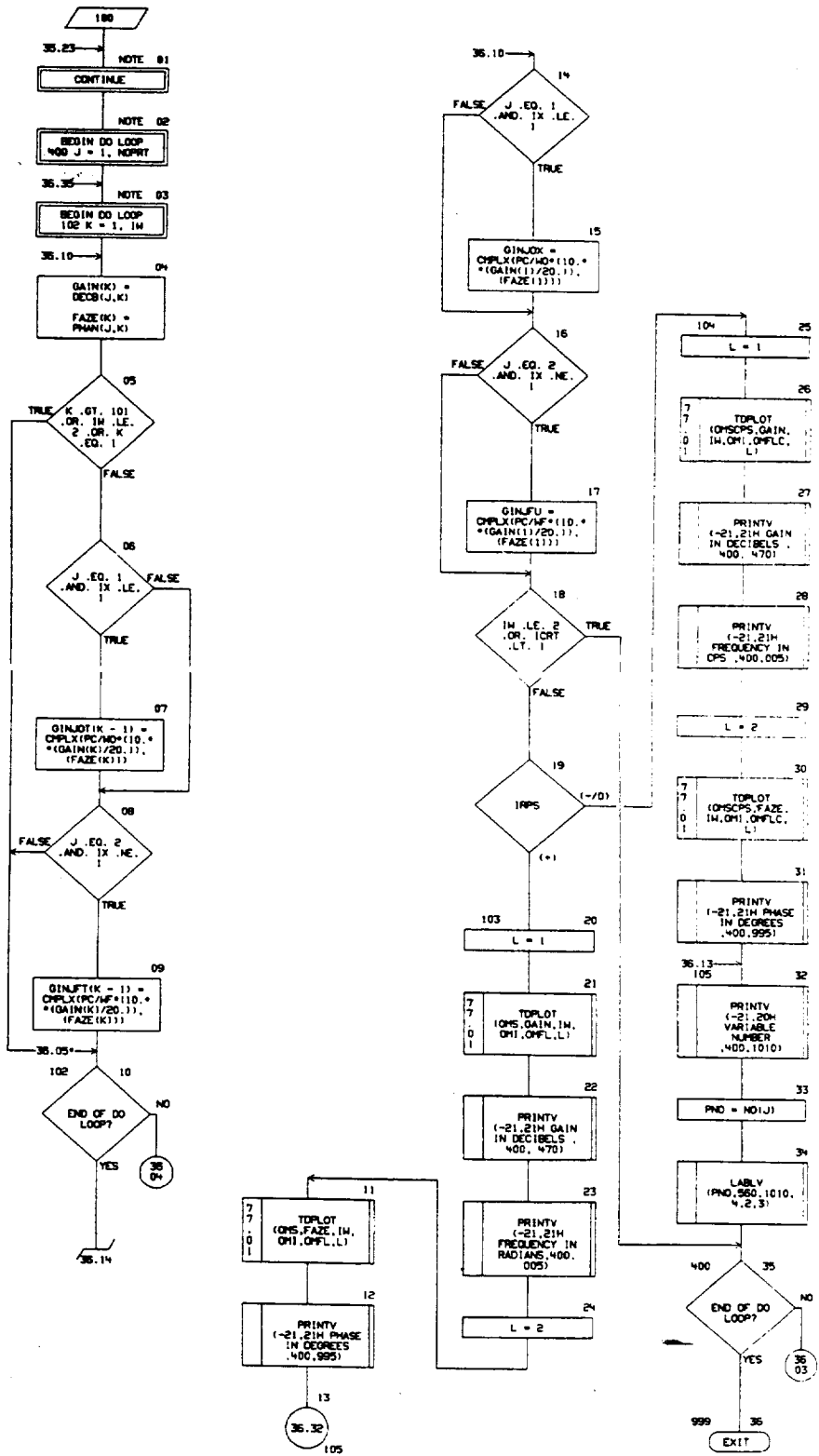
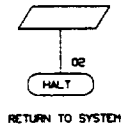
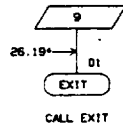


CHART TITLE - SUBROUTINE FRESP(CRT,WRITE,IX)



ORIGINAL PAGE IS  
OF POOR QUALITY

## CHART TITLE - NON-PROCEDURAL STATEMENTS

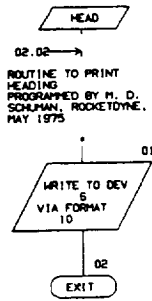
```

DIMENSION OM(101),ONSCPS(101),M(101),OMS(101),ID(126),OP(62),26)
DIMENSION LX(3),1(3),J(3),C(62,126),TD(62,63),D(126)
DIMENSION XMAG(62),NO(62),XREAL(62),XMAG(62),GAIN(101),FAZE(101)
DIMENSION VARR(62),VARI(62),PHAN(5,101),DECB(5,101)
DIMENSION ATD(126),IDF(18)
COMMON/FNH,NCTD,MR,REG,TRIG,IRPS,OM1,OPFL,N,IO,C,TD,ATD,
FREQ,GINJUX,GINUFU,PC,MO,WF
COMPLEX NOZAT(100),GINJOT(100),GINJFT(100),GINJUX,GINJUF
COMMON/CONTAP/FREQDT,FREQT(100),NOZAT,GINJOT,GINJFT,ITAPN,ITAPC,
ITAPH
STATEMENT FUNCTION DEFINITION. DELTA(0001FL) = 1.0 - ABS(0001FL)/AMAX1(ABS(0001FL),.1N693679E-38)
STATEMENT FUNCTION DEFINITION. UNIT(0002FL) = 0.5 + SIGN( 0.5,DELTA(0002FL) +
02FL)))
STATEMENT FUNCTION DEFINITION. THETA(0003FL,0004FL)
L))) + SIGN(1.0,0003FL*0004FL) * ATAN(((ABS(0004FL)*(1.0-DELTA(0003
FL))))/(ABS(0003FL)+DELTA(0003FL)))*180.0/3.141593)) * (1.0-DELTA
(0003FL)) + 180.0*DELTA(0003FL)*(1.0-DELTA(0004FL))*(.5 +UNIT(-0004
FL)))
120 FORMAT(112,13,19,13,19,F2.0,110,2F12.0)
122 FORMAT(48H-NUMBER OF SHUFFLED IN FREQUENCIES EXCEEDS 100)
121 FORMAT(8F12.0)
123 FORMAT(30H-NUMBER OF COLUMNS EXCEEDS 126)
51 FORMAT(6I12)
124 FORMAT(31H-NUMBER OF EQUATIONS EXCEEDS 62)
115 FORMAT(26H INITIAL FREQUENCY IS ZERO)
117 FORMAT(38H)THE NUMBER OF FREQUENCIES EXCEEDS 100)
90 FORMAT(3(12,14,16,F12.0))
113 FORMAT(72H)
)
99 FORMAT(18H I.D. VECTOR /(110,2414))
112 FORMAT(13H EQUATION ,13 /(14H ,1P7E14.6))
310 FORMAT(27H)MATRIX OF TIME DELAY TERMS)
115 FORMAT(24H LS HAS A NEGATIVE VALUE)
92 FORMAT(67H) VARIABLE MAGNITUDES USING A FRE
QUENCY OF ,1PE14.6,54 RPS./ 110 ,6H VARIAS
LE NUMBER MAGNITUDE PHASE /110 ,
127,1PE30.6))
1001 FORMAT(70 H) THESE ARE INTERMEDIATE RESULTS PRODUCED AFTER EACH MA
TRIX INVERSION./53H)INPUT FREQUENCY IN RADIAN/SECOND AND CYCLES/S
ECOND./59H) VARIABLE, GAIN(DB), PHASE(DEGREES) VARIABLE, E
TC. )
323 FORMAT(37H)INITIAL VALUES OF COEFFICIENT MATRIX)
321 FORMAT(//13,5X,1PE20.6/14H ,1P7E14.6))
1000 FORMAT(//F20.4,F20.5/((14,FB.2,F7.1,14,FB.2,F7.1,14,FB.2,F7.1,14,F
B.2,F7.1,14,FB.2,F7.1))
53 FORMAT(21H) FREQUENCY ,5(13H VARIABLE ,12,3H )/21H)
RPS OPS ,5(18H DECIBELS PHASE //)
94 FORMAT(1PE210.2,5(2H / , DP2F8.2))

```



CHART TITLE - SUBROUTINE HEAD



ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

10      FORMAT(1H1,////,4X,'ANALYTICAL DESCRIPTION',//,  
           4X,'FEED SYSTEM COUPLED',//,4X,'STABILITY MODEL',////,  
           4X,'COMPUTER MODEL',////,25X,'PROGRAM NAME, FSCSH, FIV VER',  
           'SION, MAY 1975',////,25X,'DEVELOPED BY, M. D. SCHUMAN',  
           ', J. K. HUNTING, AND K. H. FERTIG',//,4X,'ADVANCED PROGRAMS',  
           ', ROCKETDYNE',//,4X,'DIVISION OF ROCKWELL INTERNATIONAL',//,  
           42X,'CANDGA PARK, CALIF 91304',////,  
           25X,'SPONSORED BY, NASA/LYNDON B. JOHNSON SPACE CENTER',//,  
           42X,'HOUSTON, TEXAS 77058',//,4X,'UNDER CONTRACT NAS8-14315')

ORIGINAL PAGE IS  
 OF POOR QUALITY

C-3

08/25/75

AUTOFLON CHART SET - FBCSH COMPUTER PROGRAM

PAGE 41

CHART TITLE - BLOCK DATA

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMMON/RY/IORT, IREFLAG, ITERM, ITYPE, IWRITE, IX, A, CH, KF, KO, L, R, RHOL, V,  
VF, VO, VOLP, VOLO, ZF, ZO, OSAVE(100), FSAVE(100)  
REAL A(30), CH(30), KF, KO, L(30), R(30), RHOL(30), V(30)  
DATA IORT/0/, IREFLAG/0/, ITERM/0/, ITYPE/1/, IWRITE/0/
```

CHART TITLE - SUBROUTINE HYDRDY(IR,INPHYD,FRE,GIND,GINF,PCIN,MOIN,MFIN)

HYDRDY

04.14 ->
SUBROUTINE HYDRDY(IR,INPHYD, FREQ,GIND,GINF,PCIN,MOIN,MFIN)
PC,MO,MF

SUBPROGRAM TO CALCULATE FEED SYSTEM RESPONSE... PROGRAMMED BY J. K. HUNTING, ROCKETDYNE, MAY 1975... ARGUMENT LIST VARIABLES: IR - DATA FLAG, INPHYD - MODE FLAG, FREQ - RANGE, PCIN - INPUT FREQUENCY, MOIN - OUTPUT COMPLEX NUMBER, MFIN - OUTPUT COMPLEX NUMBER...

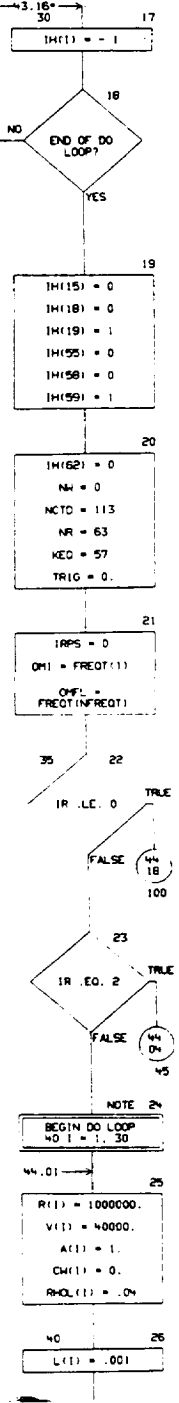
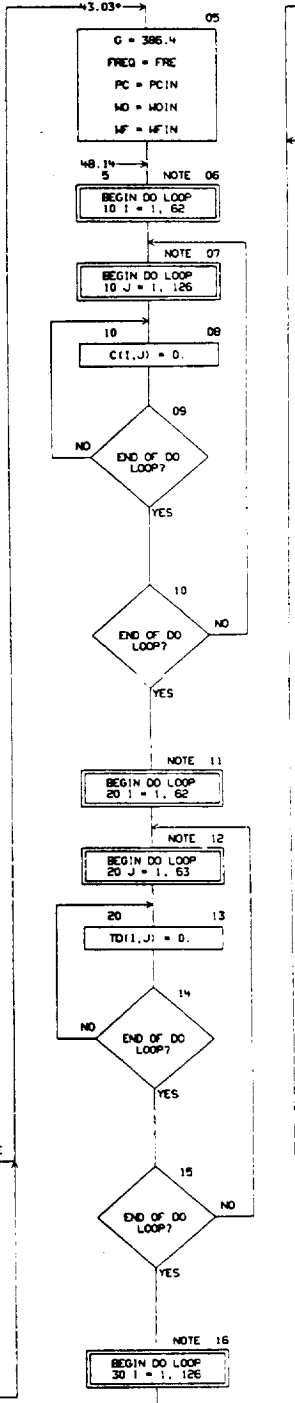
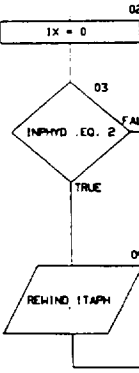
LABELLED COMMON BLOCK /COMTAP/ VARIABLES. FREQOT - ARRAY OF FREQUENCIES... GINFOT - TABLE OF COMPLEX NUMBERS WITH GAINS AND PHASE ANGLES OF OXID. INJECTOR FLOW OSCILLATIONS AT FREQOT FREQUENCIES...

NAMELIST /HYD/ VARIABLES. ICRT - PLOT FLAG, INPHYD - READ FLAG, INPHYD.EQ.2 - TERMINAL INPUT FROM ITERM, ITERM - TERMINAL INPUT FLAG, ITERM.EQ.10 - TERMINAL UNIT NO., ITERM.EQ.11 - SYSTEM TYPE FLAG, ITERM.EQ.12 - FUEL SYSTEMS DESCRIBED WITH 1 DATA READ, ITERM.EQ.13 - FUEL SYSTEM DATA READ AND CALCULATED TO ALLOW USE OF SEQUENCE NUMBERS ON NAMELIST

INPUT CARDS: A - ARRAY WITH FEED SYSTEM SEGMENT FLOW AREAS, CH - ARRAY WITH SEGMENT WALL COMPLIANCE VALUES, FREQ - (SEE COMMON BLOCK VARIABLES), FREQOT - (SEE COMMON BLOCK VARIABLES), KF - FUEL INJECTOR FACE FLEXIBILITY CONSTANT, KD - OXID. INJECTOR FACE FLEXIBILITY CONSTANT, L - ARRAY WITH FEED SYSTEM SEGMENT LENGTHS, NFREQOT - (SEE COMMON BLOCK VARIABLES), R - ARRAY WITH SEGMENT LINEARIZED RESISTANCES, RF - FUEL INJECTOR LINEARIZED RESISTANCE, RO - OXID. INJECTOR LINEARIZED RESISTANCE, RHO - ARRAY WITH SEGMENT FLUID DENSITY VALUES, V - ARRAY WITH SEGMENT FLUID ACOUSTIC VELOCITY VALUES, VF - FUEL INJECTOR FLUID ACOUSTIC VELOCITY, VO - OXID. INJECTOR FLUID ACOUSTIC VELOCITY, VOLFF - VOLUME OF FUEL INJECTOR, VOLVO - VOLUME OF OXIDIZER INJECTOR, ZF - FUEL INJECTOR INERTANCE, ZO - OXID. INJECTOR INERTANCE

DEFAULT VALUES FOR VARIABLES: ICRT=0, IWRITE=0, IPRINT=0, ITERM=0, ITYPE=1, A=1., CH=0., KF=0., KD=0., L=0.01, R=100000., RF=1., RO=1., RHO=0., V=0.000, VF=0.000, VO=0.000, VOLFF=0.1, VOLVO=0.1, ZF=0.0000, ZO=0.0000

NOTE: MAXIMUM OF 100 VALUES IN FREQOT ARRAY, MAXIMUM OF 30 VALUES IN A, CH, L, R, RHO, V ARRAYS



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - SUBROUTINE HYDRY(1R,1NPHD,FRE,0IND,0INF,PCIN,40IN,4FIN)

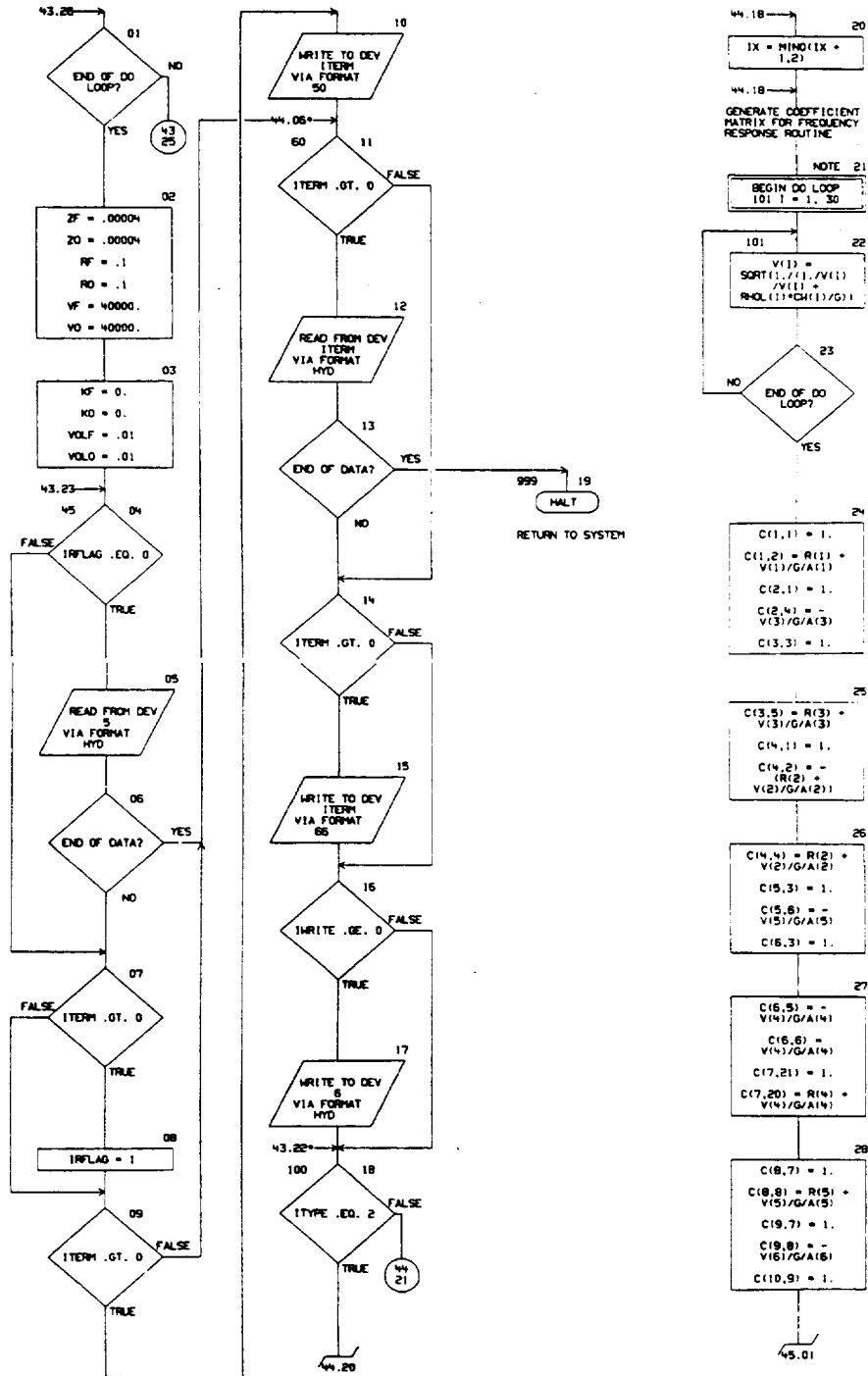
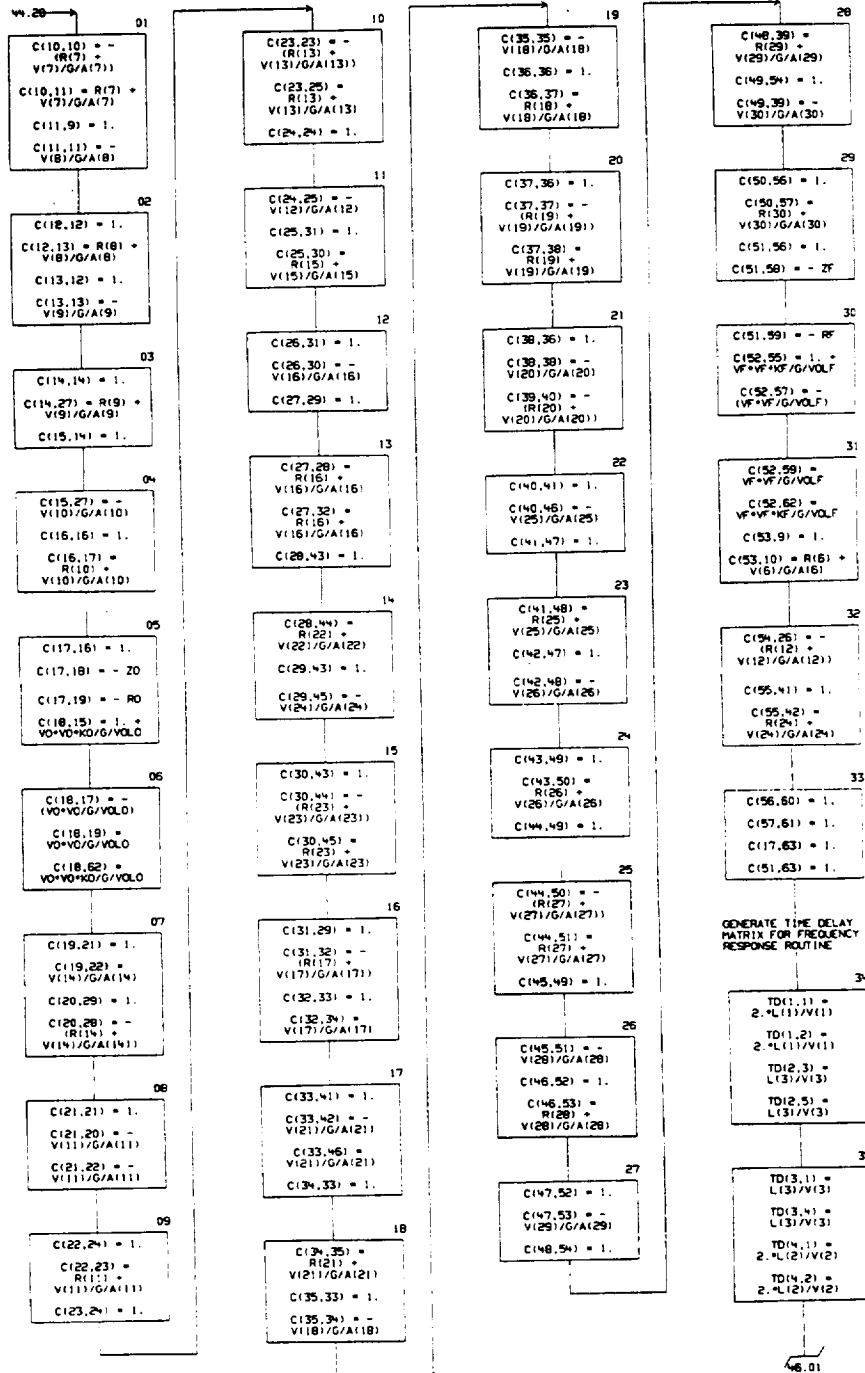


CHART TITLE - SUBROUTINE HYDRY(1R,1NPHYD,FRE,GIND,GINF,PCIN,MOIN,WFIN)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - SUBROUTINE HYDRD(1R,INPHYD,FRE,GIND,GINF,PCIN,MOIN,WFIN)

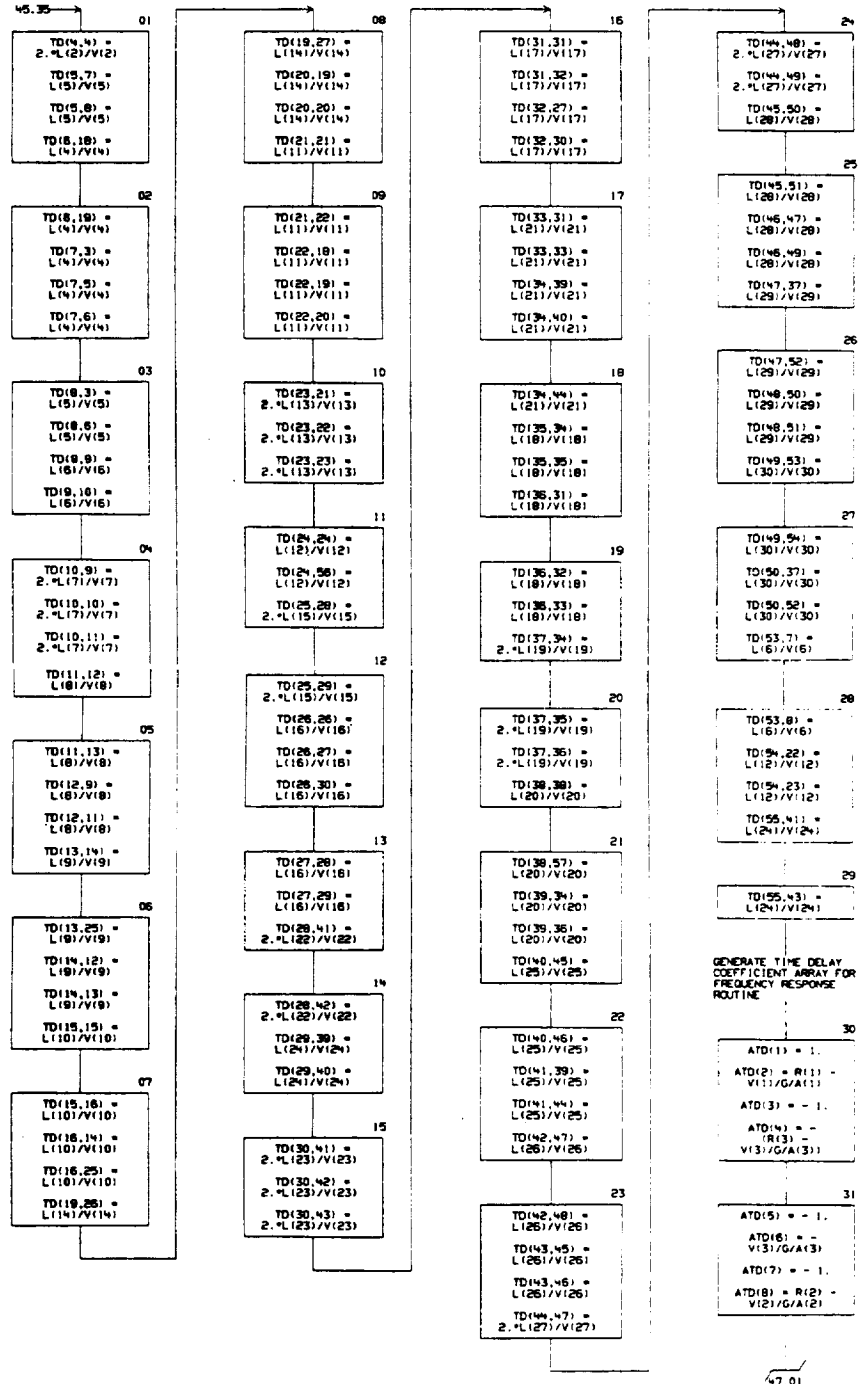




CHART TITLE - SUBROUTINE HYDROY(IR,INPHYD,FRE,GIND,GINF,PCIM,MOIN,FIN)

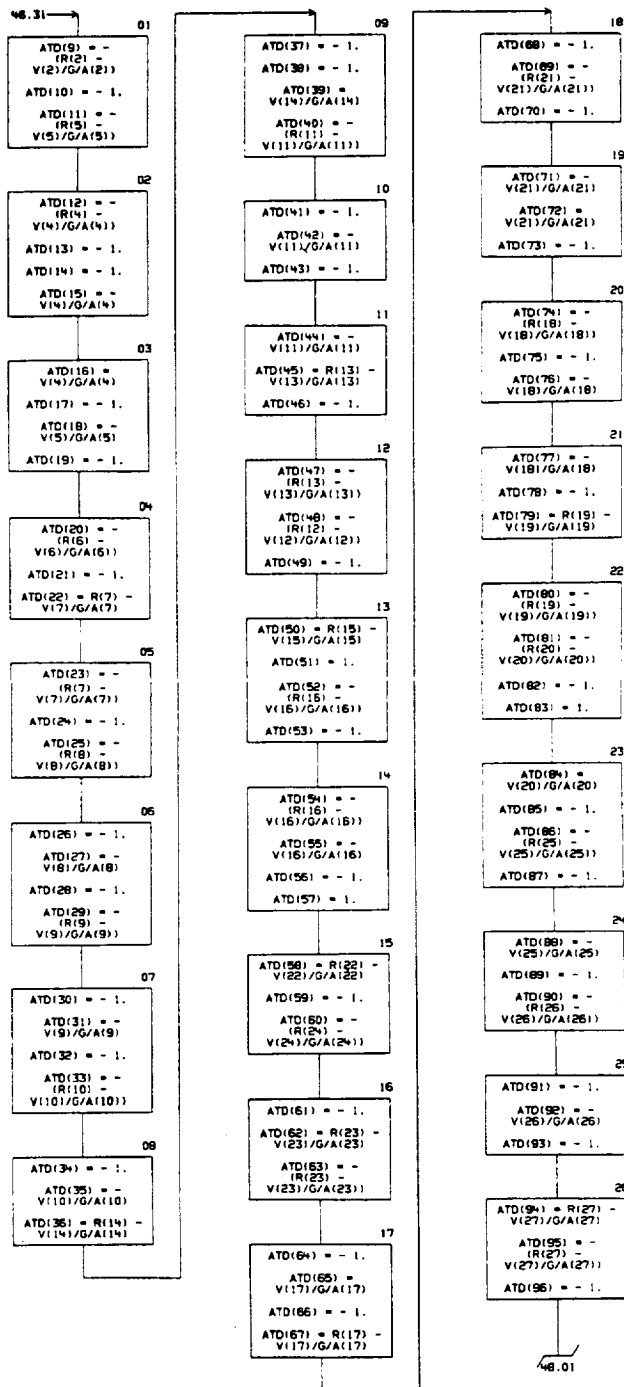


CHART TITLE - SUBROUTINE HYDRY(1R,INPHYD,FRE,GIND,GINF,PCIN,NDIN,MFIN)

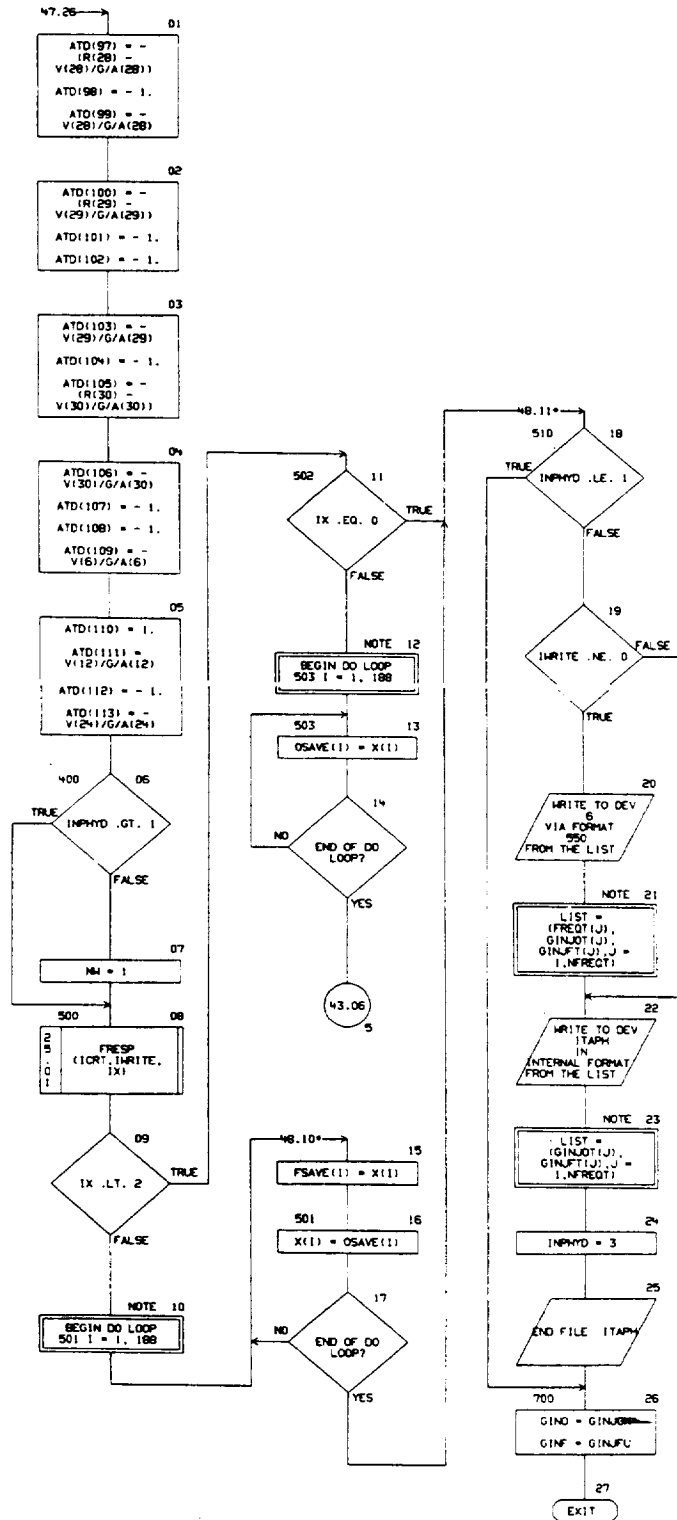


CHART TITLE - NON-PROCEDURAL STATEMENTS

```

COMMON/HY/ICRT,IRFLAG,ITERH,ITYPE,IWRITE,IX,A,CH,KF,KO,L,R,RHOL,V,
VF,VO,VOLF,VOLO,ZF,ZO,OSAVE(188),FSAVE(188)
REAL X(188)
EQUIVALENCE (X(1),A(1))
COMMON/F/NH,NCTD,NR,KED,TRIG,TRPS,OH1,OHFL,M,IM,C,TD,ATD,
FREQ,GINJOX,GINJFU,PC,MO,WF
COMPLEX NOZAT(100),GINJOT(100),GINJFT(100),GINJOX,GINJFU,
GINO,GINF
COMMON/COMTAP/NFREQ,FREQ(100),NOZAT,GINJOT,GINJFT,ITAPN,ITAPC,
ITAPH
REAL L(30),R(30),A(30),V(30),C(62,126),TD(62,63),ATD(126),M(101),
CHI(30),RHOL(30),KO,KF
INTEGERM(1)(126)
NAMELIST/HYD/L,R,A,V,OH1,OHFL,FREQ,NFREQ,IWRITE,ICRT,ZF,RF,VOLF,
VF,ZO,RO,VOLO,VO,FREQT,IM,ITERH,IRFLAG,KO,KF,CH,RHOL,IO,ITYPE,IX
50 FORMAT(' INPUT NAMELIST HYD DATA '// VARIABLES ARE, L,R,A,V,CH,RHO
L,NFREQ,FREQT,IWRITE,ITERH '// ICRT,ZF,RF,VO,KOLF,VF,KF,ZO,RO,VOLO
,VO,IO,ITYPE,IRFLAG')
66 FORMAT(' END OF HYD INPUT')
950 FORMAT(1H1,///,18X,'FEED SYSTEM RESPONSE PARAMETERS',///,
5X,'FREQUENCY',4X,'OXIDIZER INJECTION RATE',4X,
'FUEL INJECTION RATE',/,20X,'AMPLITUDE',5X,'PHASE',
6X,'AMPLITUDE',5X,'PHASE',/,15X,DPF9.3,5X,1PE11.4,
DPF9.2,5X,1PE11.4,DPF9.2)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE LOOFAC(JK,X,TX,NK,JK,FX)

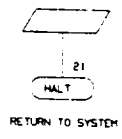
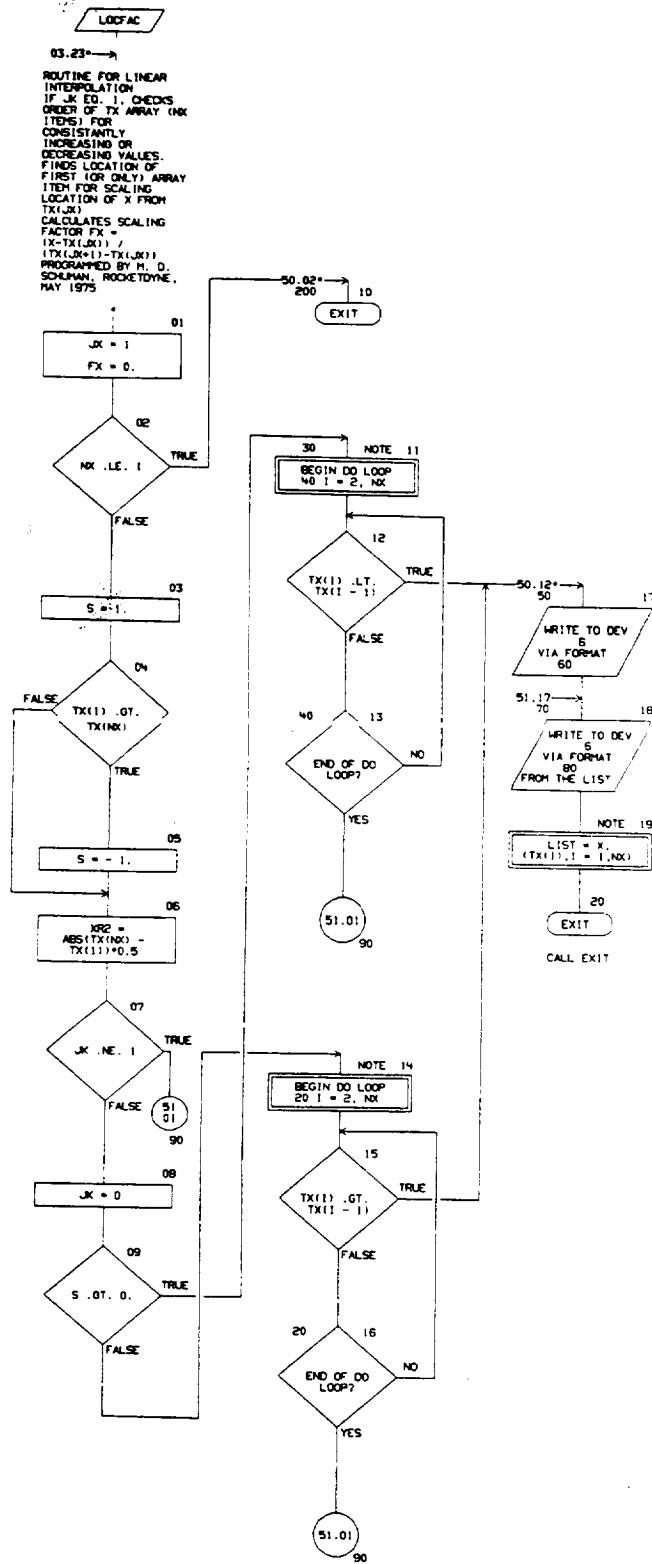


CHART TITLE - SUBROUTINE LOCFAC(JK,X,TX,NX,JX,FX)

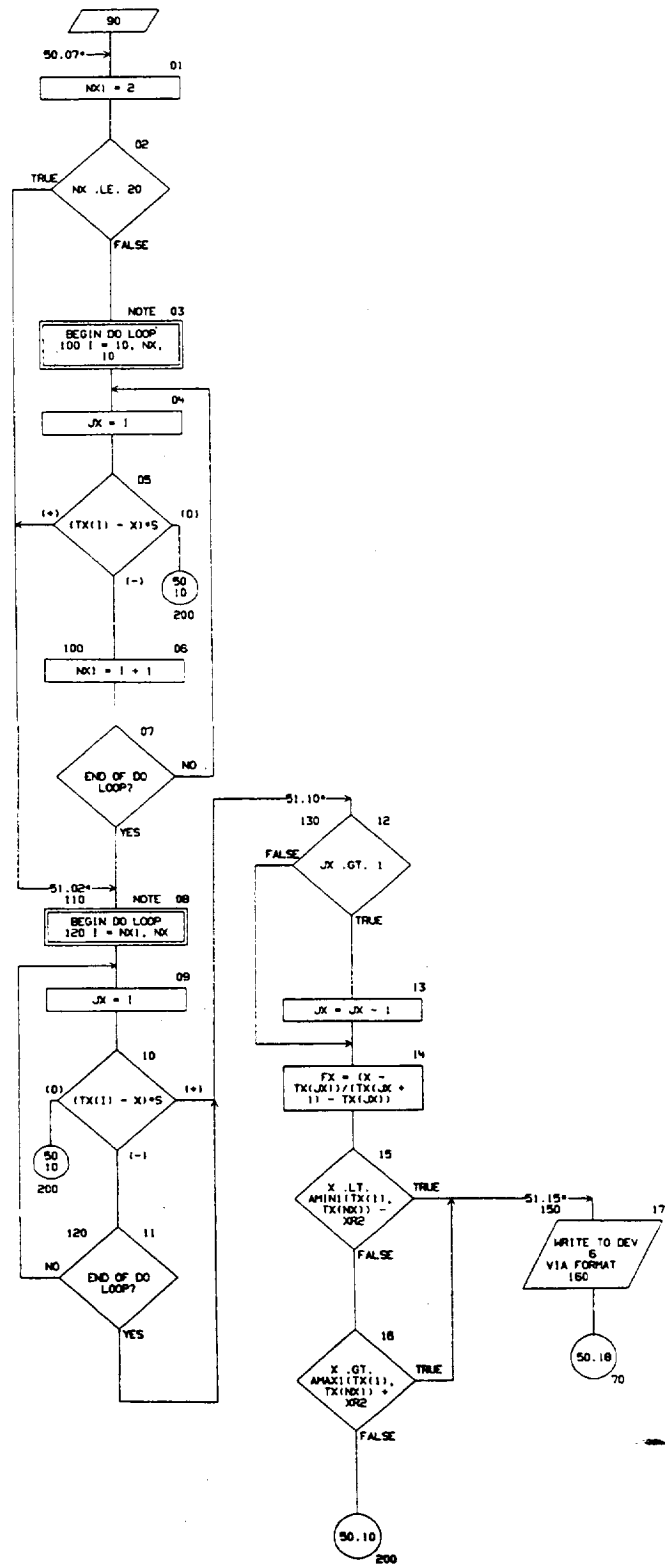


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION TX(1)
60    FORMAT(1M1 4IX 27HE R R O R  I N  T A B L E )
80    FORMAT(1M0 4IX 27HE R R O R  T O  S U B R O U T I N E  L O C F A C  //
      5X 3HX = 1PE15.4 / 4X 4HTX = 6E15.4 / 16X 6E15.4 )
160   FORMAT(1M1 22X 6HE R R O R  -  E X T R A P O L A T I O N  O F  T A B L E  1 5  B E Y O N D  R
      E A S O N A B L E  L I M I T S )
```

CHART TITLE - SUBROUTINE NOZADH (IR,GAMX,CO,FREQ,NOZA)

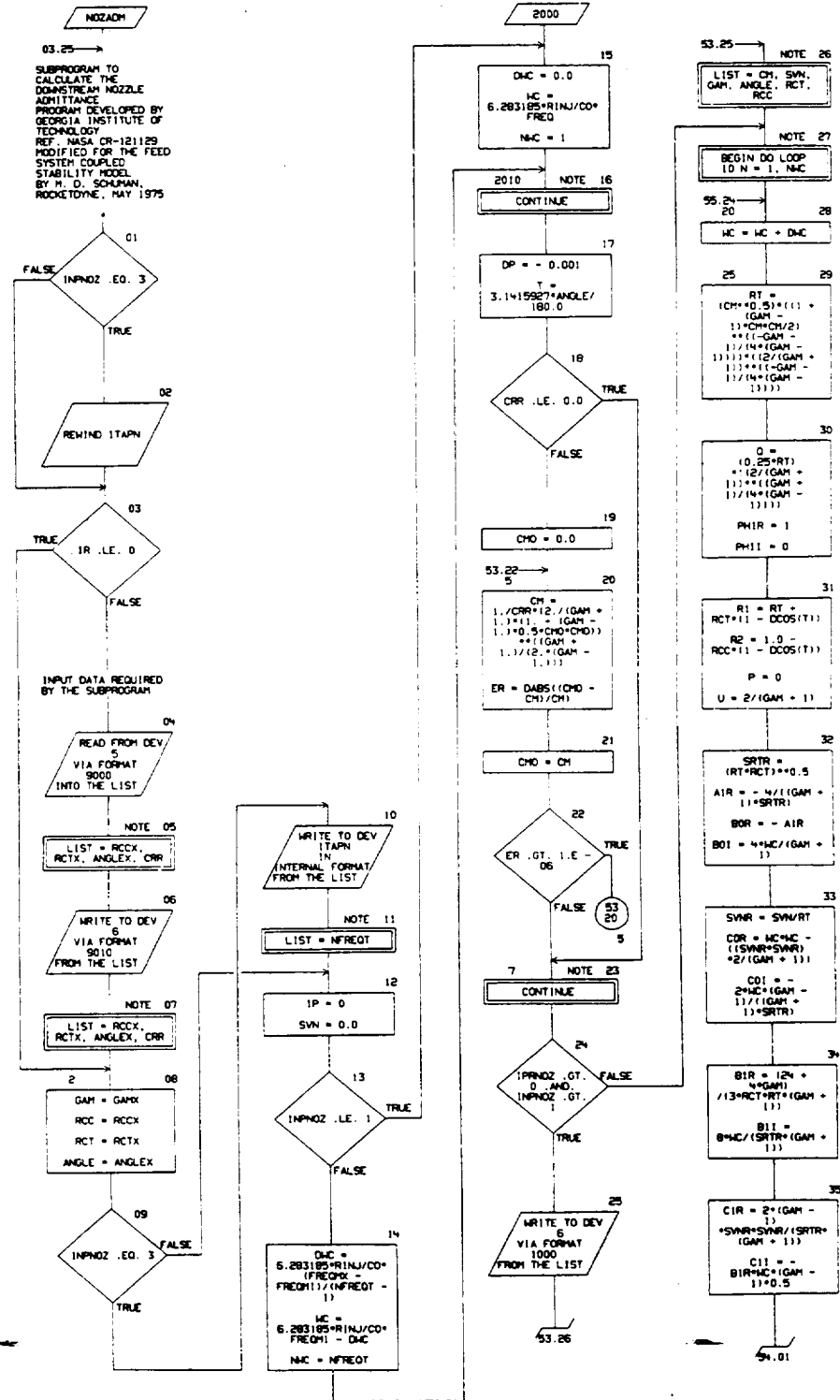


CHART TITLE - SUBROUTINE NOZADM(1R,GAM,CO,FREQ,NOZA)

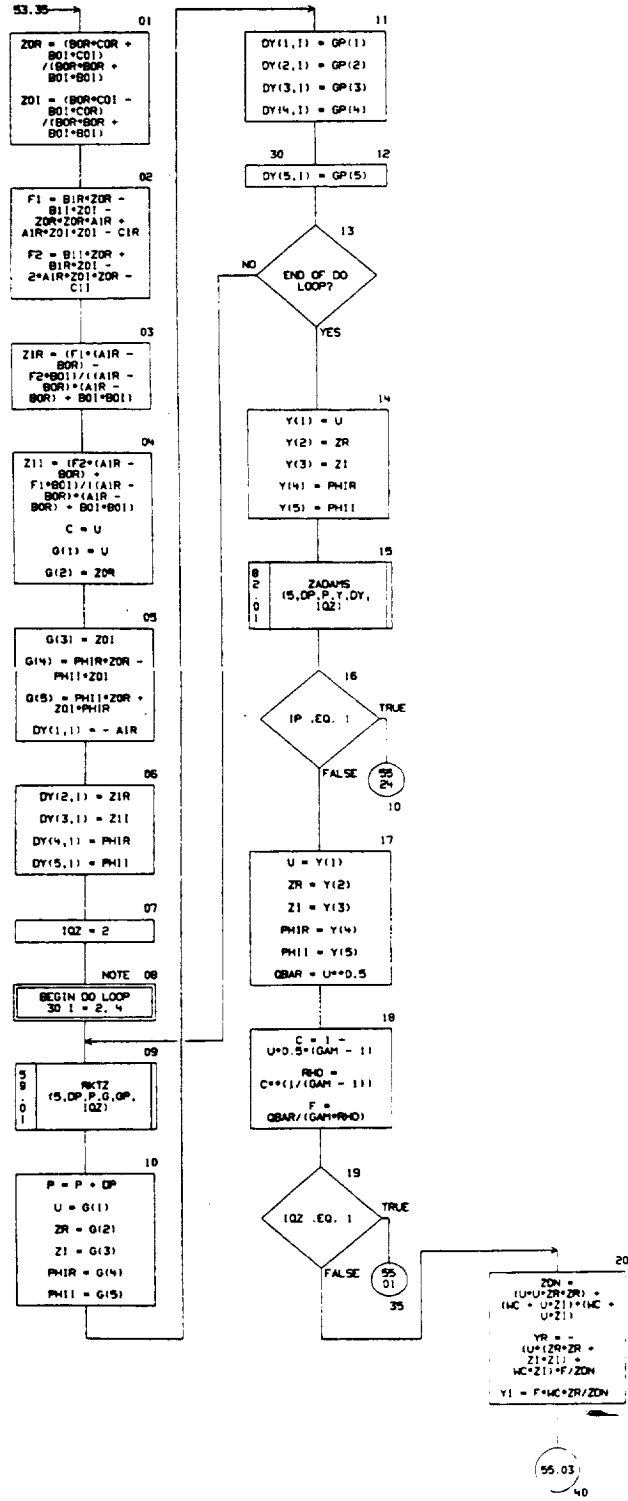
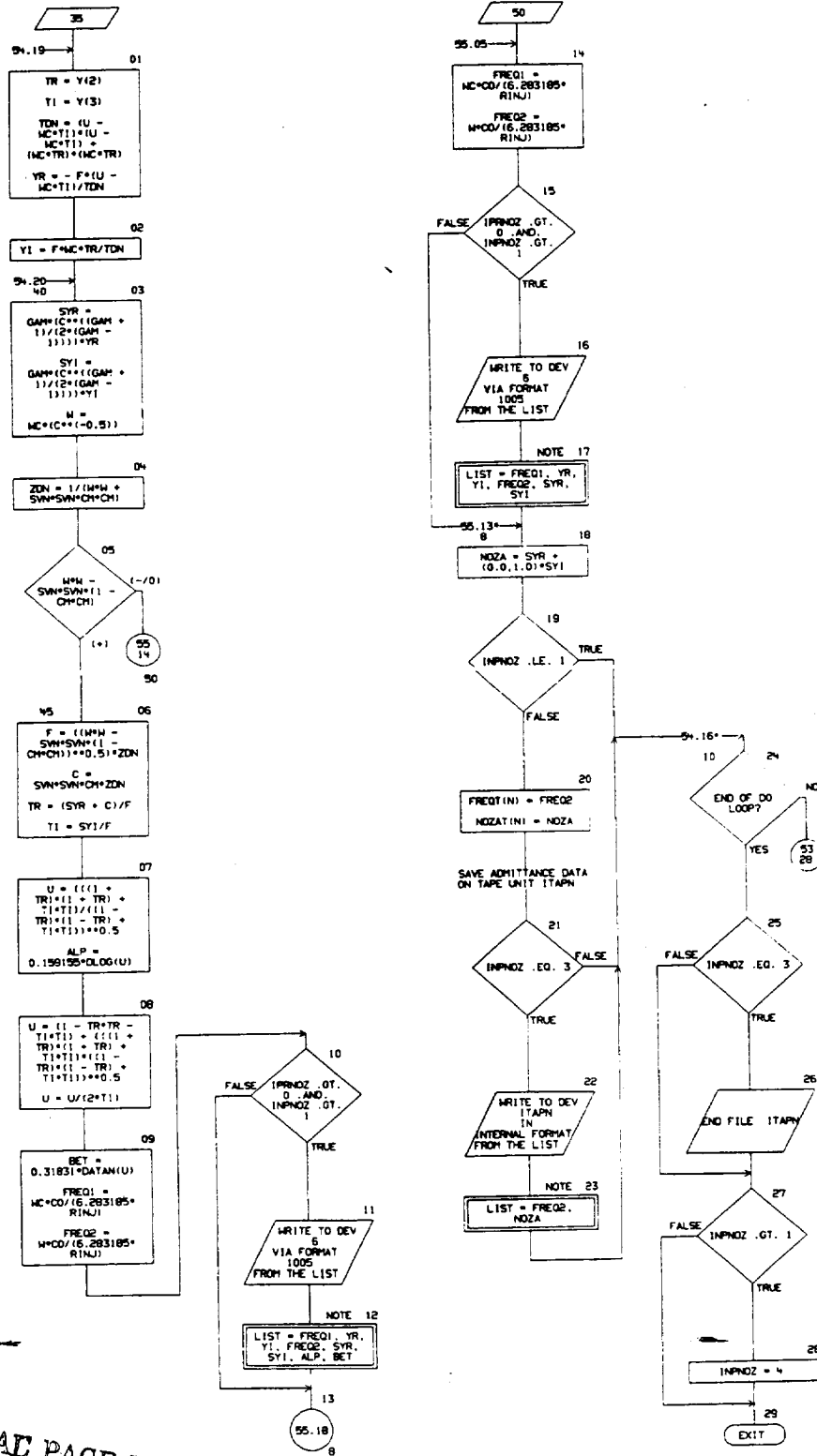




CHART TITLE - SUBROUTINE NOZADN (IR, GAM, CO, FREQ, NOZA)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

```

IMPLICIT REAL*(B1A-M,0-2)
COMPLEX NOZA, NOZAT, GINJOT, GINJFT
REAL FREQ, GANK, RCCX, RCTX, ANGLEX, CRX, RINJ, CO, FREQK,
FREQH, FREQT, FREQI, FREQ2
DIMENSION DY(5,4), G15, GP15, Y15)
COMMON /X1/GAM,SVN,ANGLE,RCC /X2/T,RT,QR,RI,R2,HC,IP
/X3/ZIR,ZII /X4/CM
COMMON /CONTAP/ NFREQT, FREQT(100), NOZAT(100), GINJOT(100),
GINJFT(100), ITAPN, ITAPC, ITAPH
COMMON /CONNOZ/ RCCX, RCTX, ANGLEX, CRX, RINJ, INPNOZ, FREQK,
FREQH, IPRNOZ
9000 FORMAT(6E12.8)
9010 FORMAT(/,5X,'RCCX =',1PE11.4,5X,'RCTX =',E11.4,5X,'ANGLEX =',
E11.4,3X,'CRX =',E11.4)
1000 FORMAT(1H1,////,30X,30M THEORETICAL NOZZLE ADMITTANCES,/,23X,
14M MACH NUMBER = ,F3.2,7M SVN = ,F6.4,9M GAMMA = ,F5.3,/,/,
7X,15M NOZZLE ANGLE = ,F4.1,2X,21M RADII OF CURVATURE,
,9M THROAT = ,F6.4,12M ENTRANCE = ,F6.4,/,/,9X,2M FC,
7X,2M YR,8X,2M YI,8X,1M F,8X,3M SYR,8X,3M SYI,
6X,5M ALPHA,5X,4M BETA,/)
1005 FORMAT(6X,F6.1,2F10.5,F10.2,4F10.5)

```

CHART TITLE - SUBROUTINE RKTDFIP,G,OP1

RKTDFIP

99.071-  
 ROUTINE FOR  
 SUBPROGRAM INOZACH  
 PROGRAM DEVELOPED BY  
 GEORGIA INSTITUTE OF  
 TECHNOLOGY  
 REF. NASA CR-121129

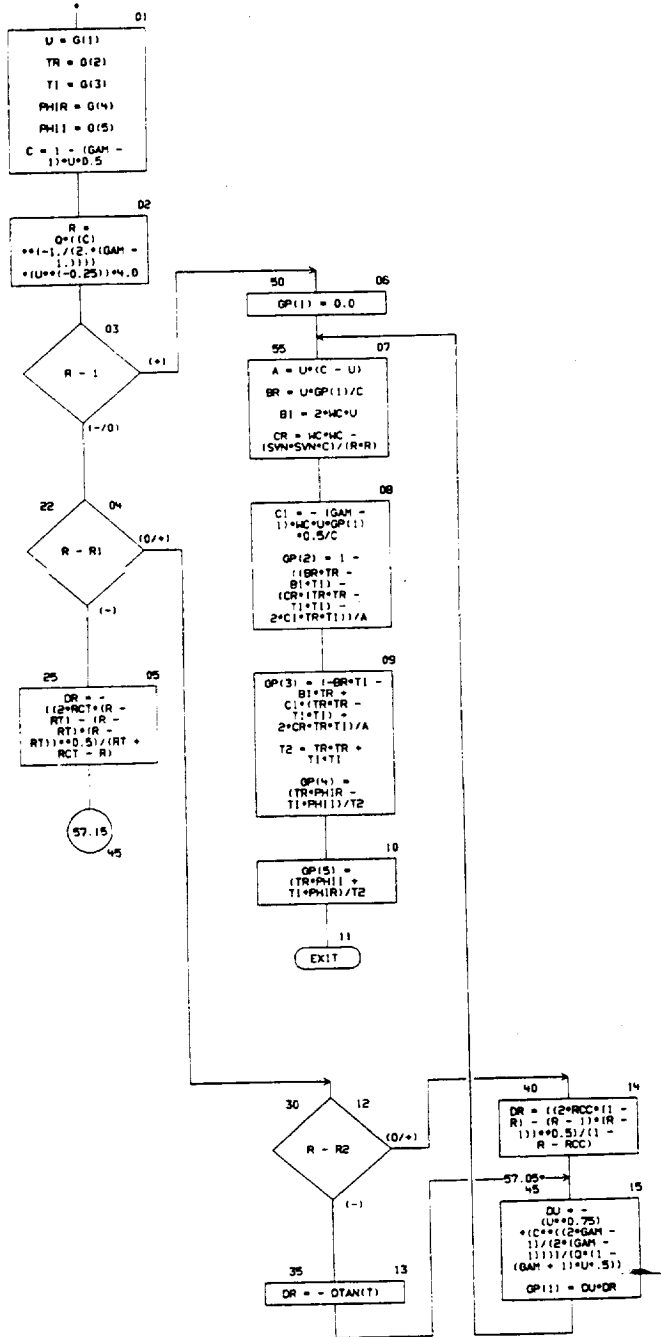
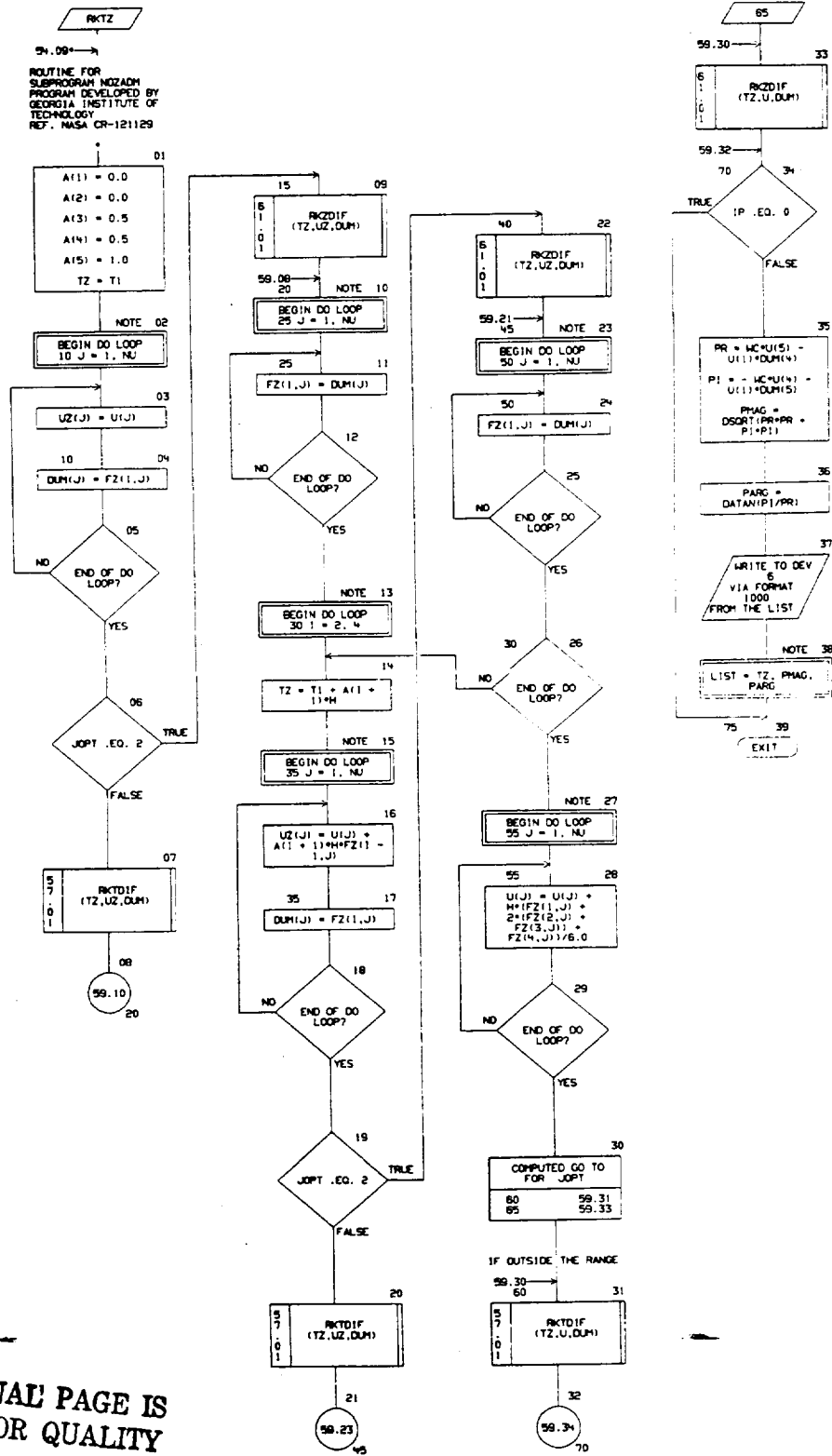


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
IMPLICIT REAL*(B16-H,D-Z)
COMMON /X1/GAM,SYN,ANGLE,RCY,RCX /X2/T,RT,Q,R1,R2,MC,IP
DIMENSION G(15), GP(15)
9  FORMAT(3X,'PRINTING FROM CARD 4570',/,3X,'R=',E15.8,
      3X,'R1=',E15.8,3X,'RT=',E15.8)
```

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE RKTZ(U,H,T1,U,DUM,JOPT)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      IMPLICIT REAL*(A-H,O-Z)
      COMMON /A2/T,RT,Q,R1,RE,MC,IP
      DIMENSION U(5), A(5), UZ(5), FZ(4,5), DUM(5)
1000  FORMAT(4SX,F5.4,1X,F10.5,3X,F10.5)
```

CHART TITLE - SUBROUTINE RKZDIF(P,G,OP)

RKZDIF

59.09 →  
ROUTINE FOR  
SUBPROGRAM NOZADM  
PROGRAM DEVELOPED BY  
GEORGIA INSTITUTE OF  
TECHNOLOGY  
REF. NASA CR-121129

01  
U = G(1)  
ZR = G(2)  
Z1 = G(3)  
PHIR = G(4)  
PHI1 = G(5)

02  
P

61.04  
15

10 03  
OP(1) = W/(GAM +  
1) \* ((RCT\*RT)  
\*\*0.5)  
OP(2) = Z1R  
OP(3) = Z11  
OP(4) = Z1R  
OP(5) = Z11

61.16  
20

61.02  
15 04  
C = 1 - (GAM -  
1) \* U \* 0.5  
R =  
O\*(C)  
\*\*(-1./((2.\*(GAM -  
1.)))  
\*(U\*\*(-0.25)))\*\*0.0

05  
R .LT. RT

FALSE  
TRUE

06  
RT = R

07  
R1 .LT. RT

FALSE  
TRUE

08  
R1 = RT

09  
R - 1

61.03  
20  
16  
EXIT

50 12  
OP(1) = 0.0

55 13  
A = U\*(C - U)  
BR = U\*OP(1)/C  
BI = 2\*MC\*U  
CR = MC\*MC -  
SVN\*SVN/C/(R\*R)

14  
C1 = -(GAM -  
1) \* MC \* U \* OP(1)  
\* 0.5 \* (1/C)  
OP(2) = ((BR\*ZR -  
BI\*Z1 - CR/A) -  
ZR\*ZR + Z1\*Z1

15  
OP(3) = ((BI)\*ZR +  
BR\*Z1 - C1/A) -  
2\*ZR\*Z1  
OP(4) = 2R\*PHIR -  
Z1\*PHI1  
OP(5) = 2R\*PHI1 +  
Z1\*PHIR

61.03  
20  
16  
EXIT

22 10  
R - R1

30 17  
R - R2

25 11  
DR = -  
(12\*RCT\*R -  
RT) - (R -  
RT)\*IR -  
RT)\*\*0.5/(RT +  
RCT - R1

61.20  
45

35 18  
DR = - DTAN(T)

40 19  
DR = ((2\*RCC\*(1 -  
R) - (R - 1)\*(R -  
1))\*\*0.5)/(1 -  
R - RCC)

61.11  
45 20  
DU = -  
(U\*\*0.75)  
\*(C\*\*((2\*GAM -  
1)/(2\*(GAM -  
1))) / (O\*(1 -  
GAM + 1) \* U \* 5))  
OP(1) = DU\*DR

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
IMPLICIT REAL*8(A-H,O-Z)
COMMON /X1/GAM,SYN,ANGLE,RCI,RCO /X2/T,RT,Q,R1,R2,MC,IP
/X3/ZIR,ZII
DIMENSION G(5), GP(5)
16 FORMAT(3X,'PRINTING FROM CARD 5000',/,3X,'R=',E15.8,
3X,'R1=',E15.8,3X,'RT=',E15.8)
```

ORIGINAL PAGE IS  
OF POOR QUALITY



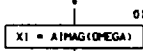
CHART TITLE - SUBROUTINE SOLVH(KWHERE)



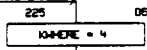
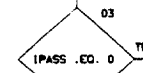
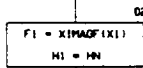
05.30  
SUBPROGRAM SOLVES FOR THE COMPLEX FREQUENCY BASED ON THE UPSTREAM AND DOWNSTREAM NOZZLE ADMITTANCE PROGRAMMED BY K. W. FERTIG, ROCKWELL, MAY 1975

WHEN ISCHT = 1, 4, OR 5 THIS PROGRAM COMPUTES CNOZA-NOZA FOR A GIVEN OMEGA. WHEN ISCHT = 1 OR 4, THIS ROUTINE ALSO FINDS IMAG(OMEGA) THAT MINIMIZES CNOZA-NOZA KEEPING REAL(OMEGA) CONSTANT

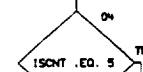
FURTHER, WHEN ISCHT = 1 OR 4, TESTING IS PERFORMED TO DETERMINE IF A ROOT IS NEARBY.



X1MAG COMPUTES FNR=10, 11 \*FNI=CNOZA-NOZA, HNC=CABS(CNOZA-NOZA), WHEN ISCHT = 1 OR 4 AND ISLP = 1, X1MAG=(HNR\*\*2)/D(IMAG(OMEGA))



WHEN ISCHT = 5, BYPASS MINIMIZATION OF CABS(CNOZA-NOZA) W.R.T. IMAG(OMEGA) AND COMPUTATION OF JACOBIAN OF FNR,FNI W.R.T. OMEGA.



IF ISLP=0, THEN (CNOZA-NOZA) AT OMEGA = DELTA REAL(OMEGA) HAS JUST BEEN COMPUTED IN X1MAG. GO TO 146 TO FINISH COMPUTATION OF JACOBIAN AND THEN COMPUTE FTEST2.



CHART TITLE - SUBROUTINE SOLVING(OHRE)

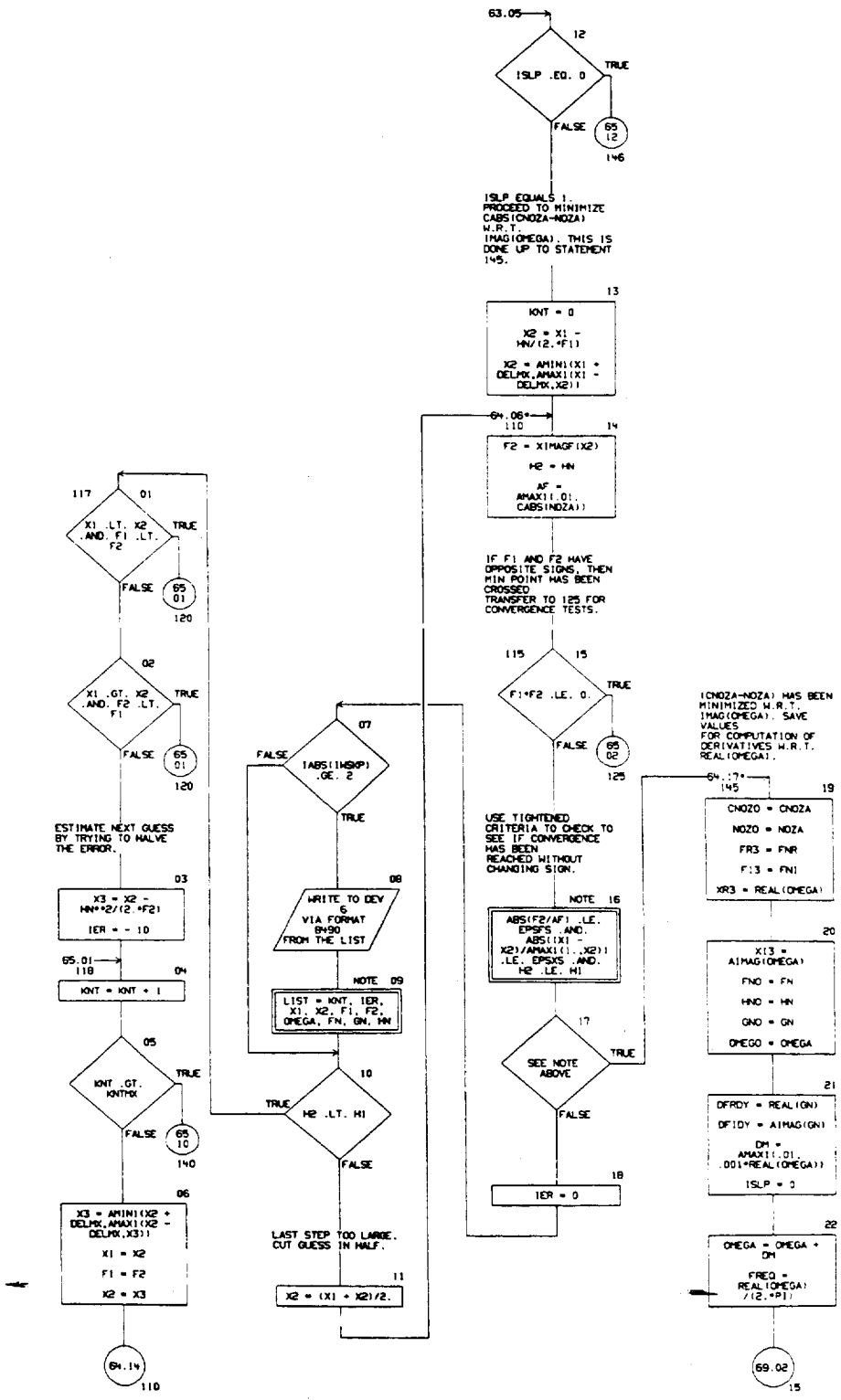
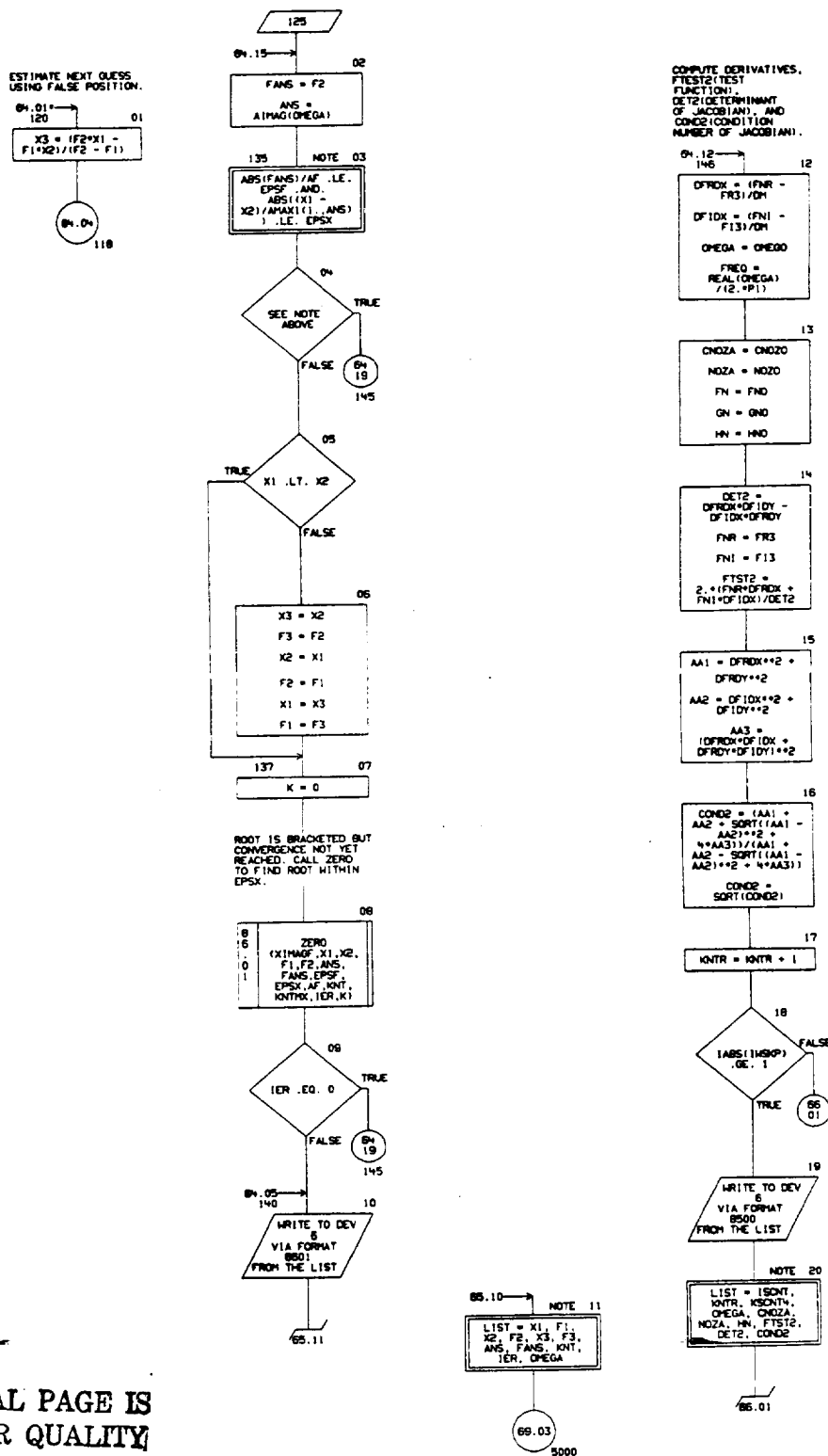


CHART TITLE - SUBROUTINE SOLV(MODE)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - SUBROUTINE SOLVING(HERE)

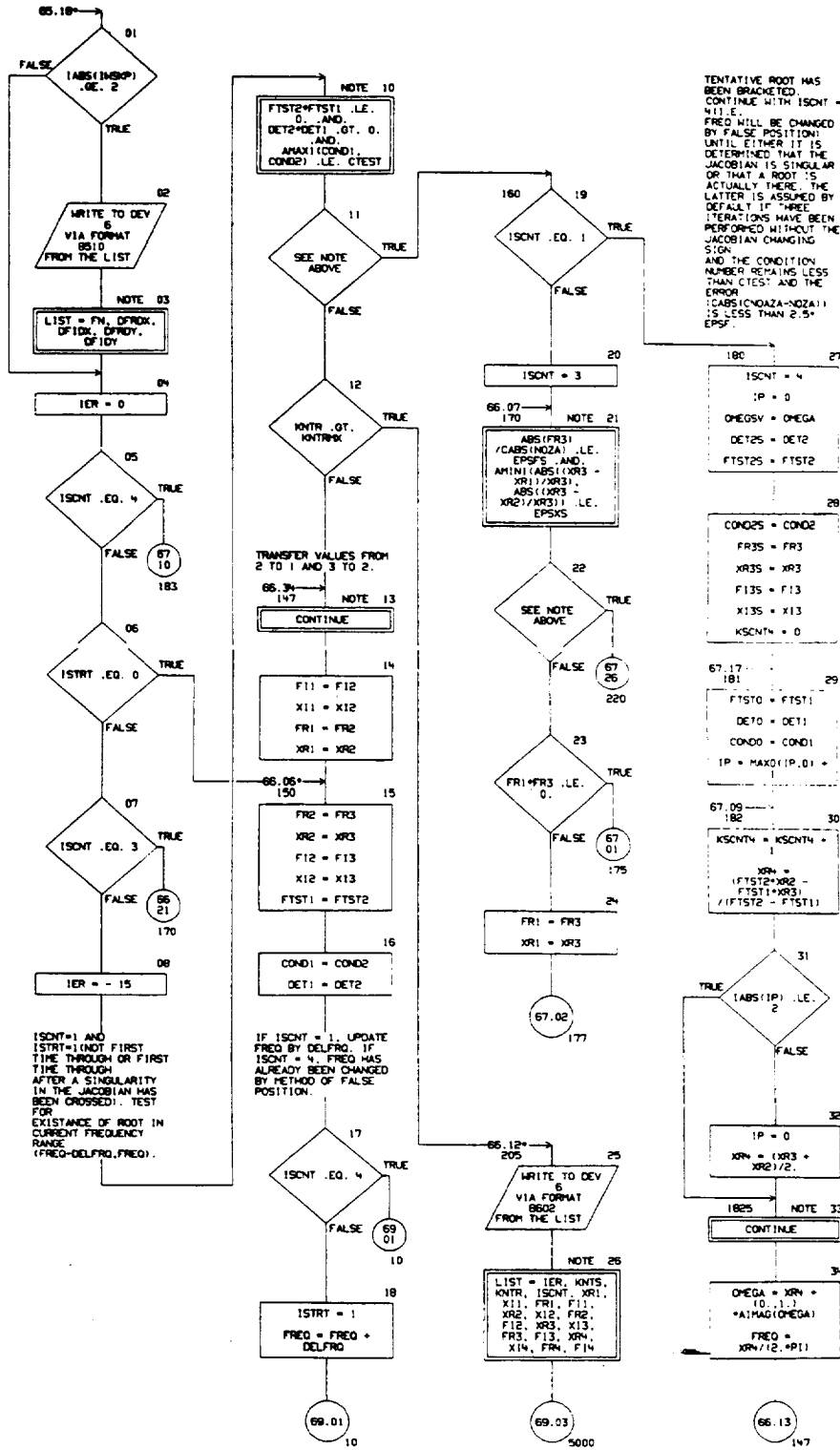
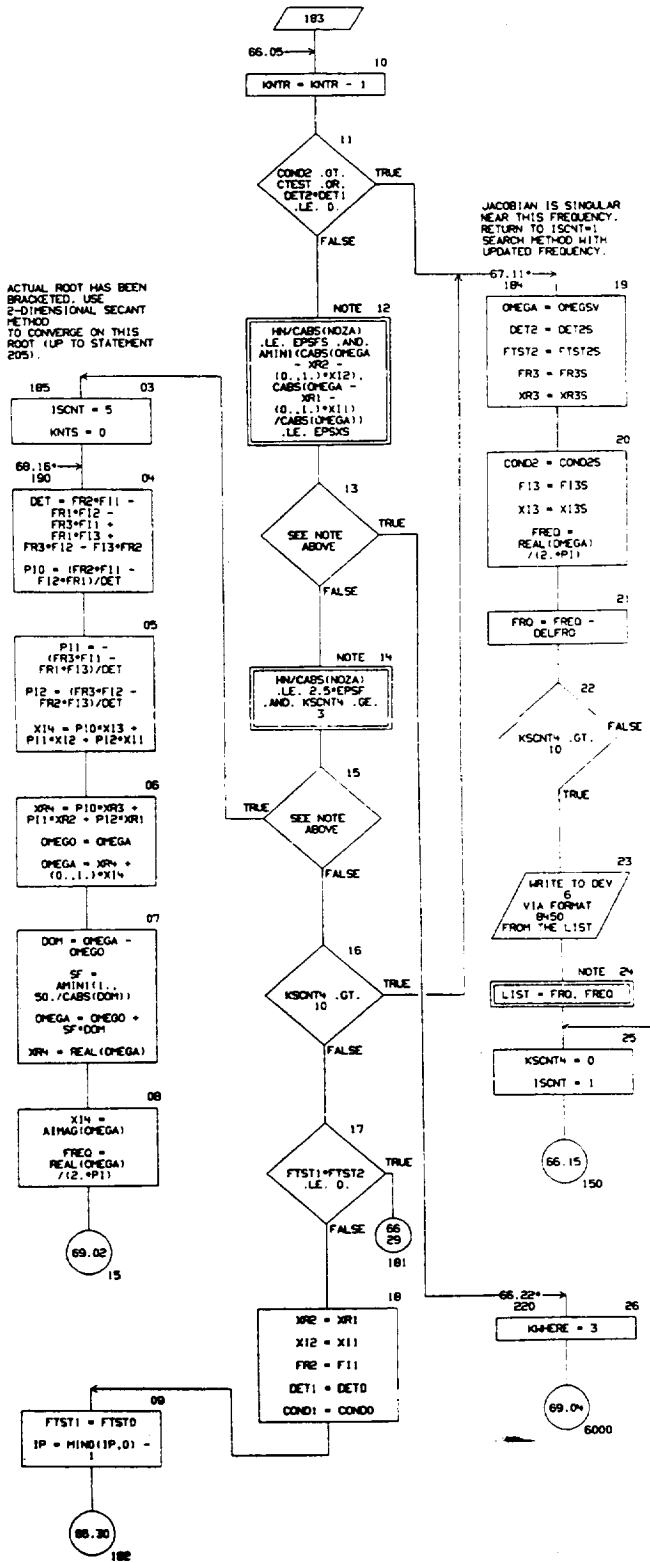
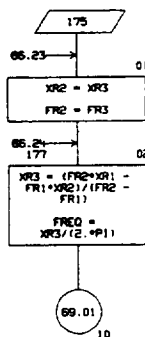


CHART TITLE - SUBROUTINE SOLVW(HIGHERE)



ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE SOLVH(OHHERE)

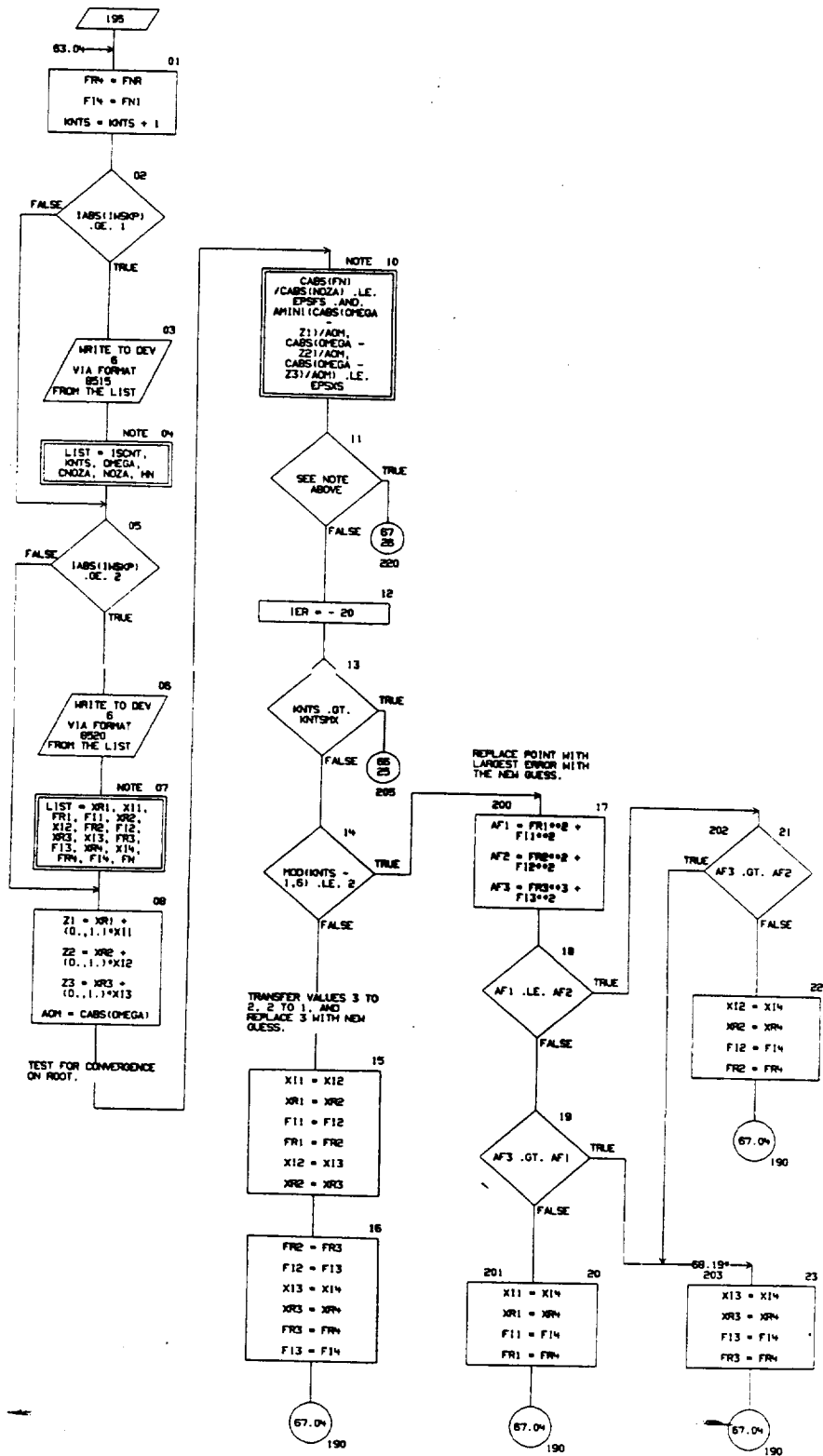


CHART TITLE - SUBROUTINE SOLW(KMERE)

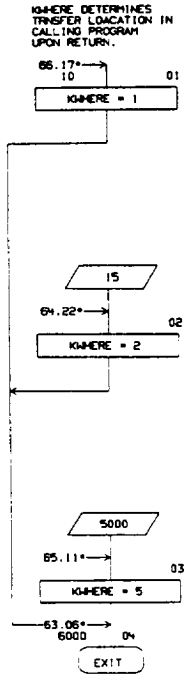


CHART TITLE - NON-PROCEDURAL STATEMENTS

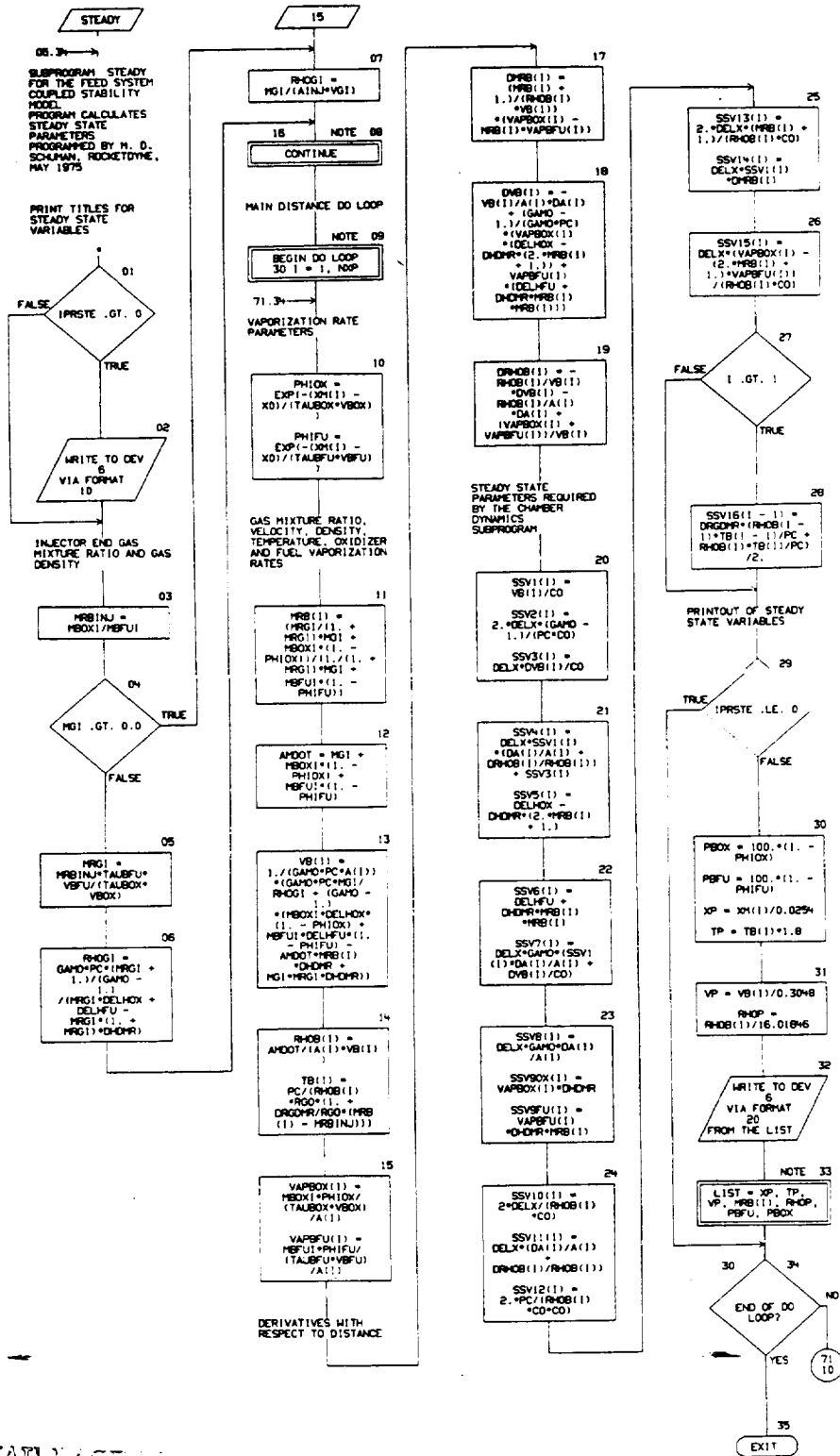
```

COMPLEX OMEGA, P, RHO, V, HR, T, CHOZA, VXO,
NOZA, OMEGO,
DOM, Z1, Z2, Z3, FN, NOZAPR, GN,
CHOZO, NOZO, FNO, GNO, OMEOSV
COMMON /CONCHM/ P(100), RHO(100), V(100), HR(100), T(100),
VXO, OMEGA, CHOZA, DELP
COMMON /FZERO/ NOZA,NOZAPR,GN,FN,FNR,FNI,HN,ISONT,ISLP
EXTERNAL XIMAGF
DATA P1/3.141593/
COMMON /SOLVE/ FREQ, DELFRQ, DELPK, EPSF, EPSK, EPSFS, EPSKS,
FRONAX, CTEST, IPASS, KNTR, ISTRT, KSONT4, I4SKIP,
KNTRX, KNTSPX, KNTRX4
0490 FORMAT(//2I5,1PE13.5/2X,3(1PE13.5,' ',1PE12.5),1PE15.5)
0601 FORMAT(/// **** UNABLE TO FIND ROOT FOR IMAG PART OF F ****
/// X1,F1,X2,F2,X3,F3,ANS,FANS,KNT,IER,OMEGA = //
3X,1PE13.5/3X,2I10/3X,1PE13.5,' ',1PE13.5)
0500 FORMAT(//3I5,3(1PE14.5,' ',1PE12.5)/2X,1PE13.5)
0510 FORMAT(2X,1PE14.5,' ',1PE12.5,2X,1PE13.5)
0450 FORMAT(/// *****//
' WARNING, POSSIBLE ROOT IN FREQUENCY RANGE, '
1PE15.5/// *****//)
0515 FORMAT(//2I5,3(1PE14.5,' ',1PE12.5),1PE14.5)
0520 FORMAT(2X,4(1PE14.5,' ',1PE12.5)/2X,4(1PE14.5,' ',
1PE12.5)/2X,1PE14.5,' ',1PE12.5)
0602 FORMAT(/// **** EXCEED CONVERGENCE LIMIT ****//
' IER,KNTS,KNTR,ISONT = ',4I10// X,F FROM I-N = //
3X,1PE15.5,' ',1PE15.5,1PE18.5,' ',1PE15.5)

```



CHART TITLE - SUBROUTINE STEADY(1PRSTE)



ORIGINAL PAGE IS OF POOR QUALITY

CHART TITLE - NON-PROCEDURAL STATEMENTS

```

COMPLEX COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
        COX10, COX11, COX12, COX13, COX14, COX15, COX16,
        CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
        CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
REAL MRBINU, MBOX1, MBFU1, MRB, MRG1, MG1, NUBOX, NUBFU, MAG
COMMON /COMOBH/ XOXX, XOFU, MBOX1, MBFU1, TAUBOX, TAUBFU, VBOX,
        VBFU, GAMO, RGO, DELHGX, DELHFU, PC, CO,
        COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
        COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
        CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
        CFU12, CFU13, CFU14, CFU15, CFU16, MAG, XIMPFU, XIMPOX,
        CS, DCSOHR, DHGHR, DRGDHR, ADVGX, ADDGX, TORAGX, DELVGX,
        NUBOX, DTGXDH, ADVFU, ADDFU, TORAGF, DELVFU, NUBFU, DTFLDH
COMMON /CONSTS/ MRB(100), TB(100), RHOB(100), VB(100),
        DMRB(100), DRHOB(100), DVB(100), VAPBOX(100), VAPBFU(100),
        SSV1(100), SSV2(100), SSV3(100), SSV4(100),
        SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9OX(100),
        SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),
        SSV14(100), SSV15(100), SSV16(100),
        RHOG1, VG1, MRG1, MG1
COMMON /COMARE/ NOP, X(100), XM(100), A(100), DA(100), DELX,
        XO, XNOZ, AINU
10  FORMAT(1H1,/,/,/,3IX,'STEADY STATE SOLUTION',/,/,
        7X,'DISTANCE',
        4X,'TEMPERATURE',2X,'VELOCITY',3X,'MIXTURE',
        6X,'DENSITY',7X,'PERCENT VAPORIZED',/,7X,'(INCHES)',5X,
        '(RANKINE)',4X,'(FT/S)',5X,'RATIO',5X,
        '(LB/FT**3)',6X,'FUEL',3X,'OXIDIZER',/)
20  FORMAT(6X,F9.4,2X,2F11.2,F11.4,1PE15.5,OP11.2,F9.2)

```

CHART TITLE - SUBROUTINE TADAMS(H,H,X,Y,DY,10Z,10)

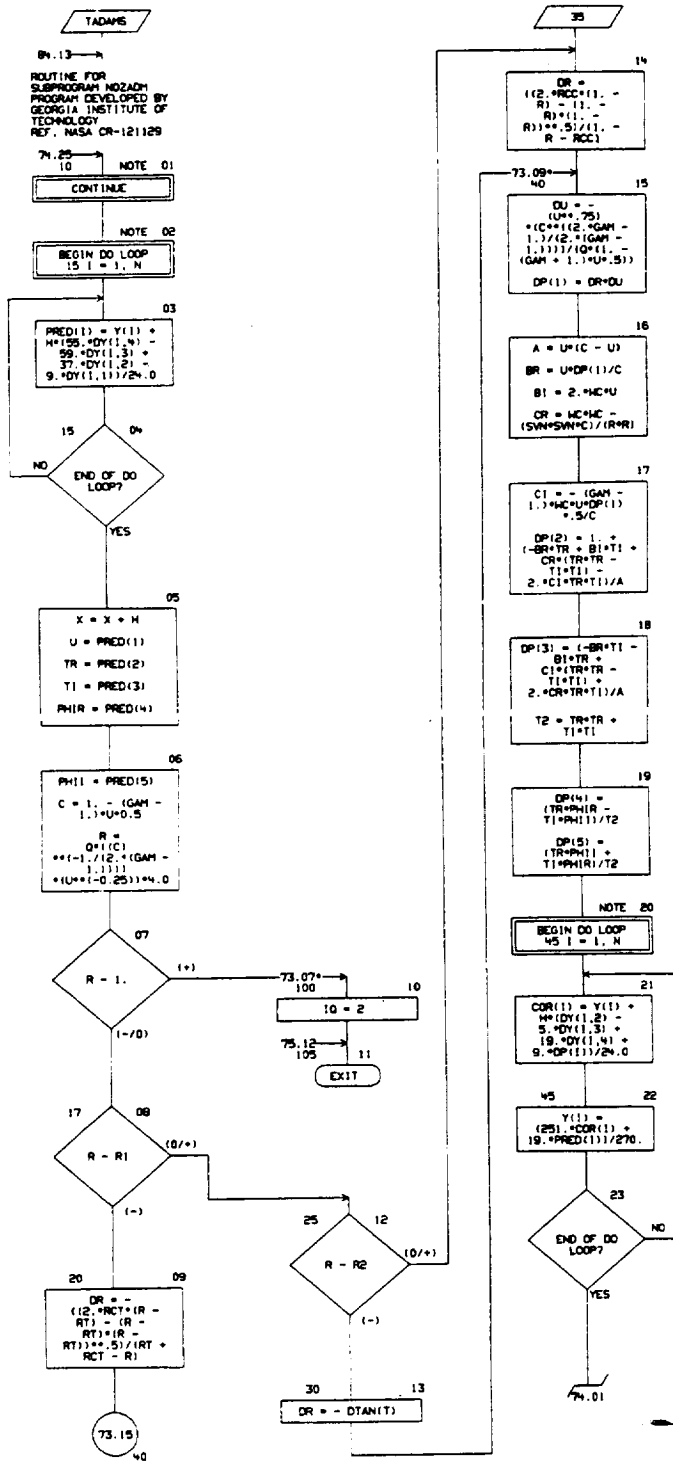


CHART TITLE - SUBROUTINE TADAN(SIN,N,X,Y,DY,IGZ,IG)

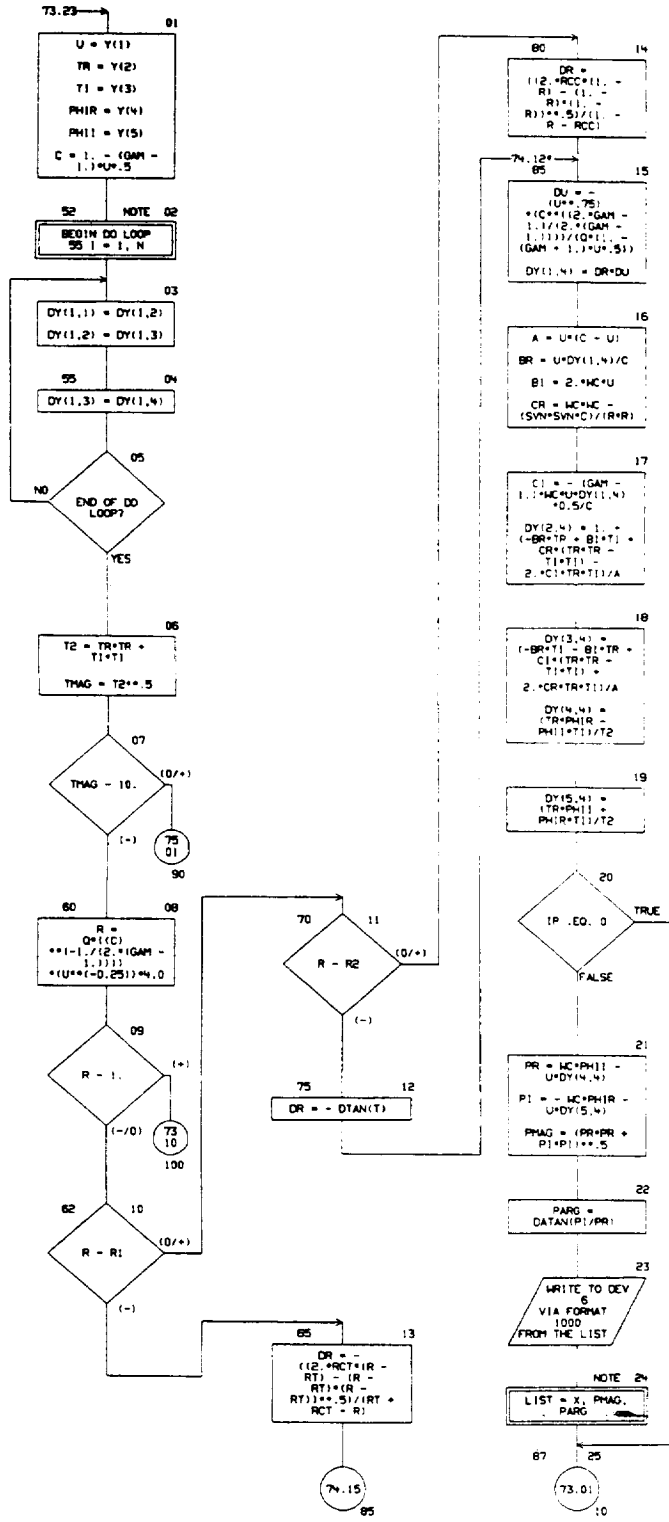
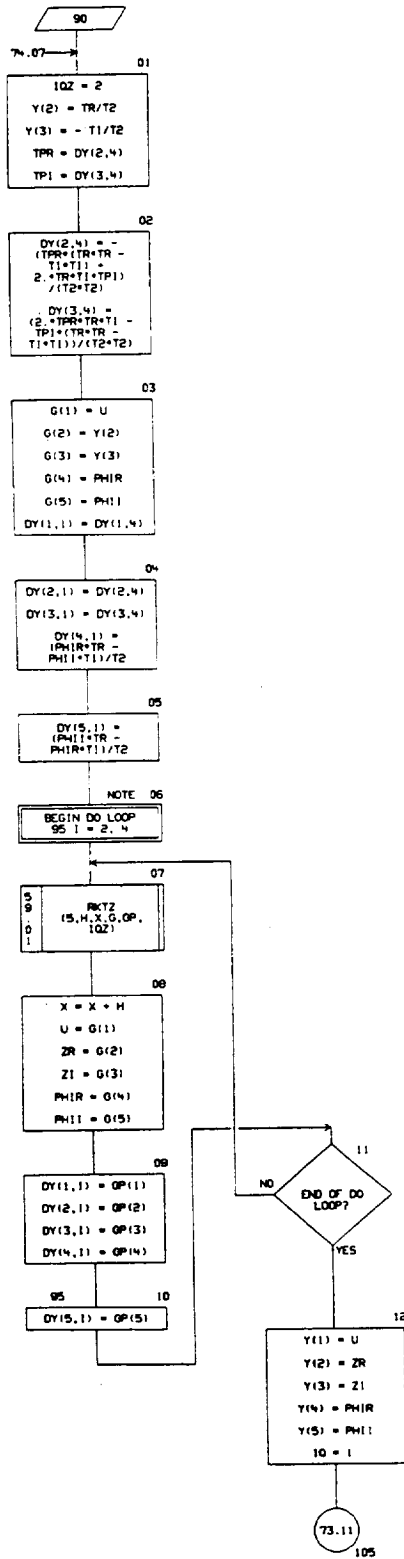


CHART TITLE - SUBROUTINE TADAMS(N,H,X,Y,DY,IOZ,IO)



ORIGINAL PAGE IS  
OF POOR QUALITY

## CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      IMPLICIT REAL*(A-H,O-Z)
      COMMON /X1/GAM,SWN,ANGLE,RCI,RCO /X2/T,RT,Q,R1,R2,MC,IP
      COMMON /X4/CM
      DIMENSION COR(5), DP(5), DY(5,4), PRED(5), Y(5), G(5), GP(5)
16     FORMAT(3X,'PRINTING FROM CARD 2180',/,3X,'R=',E15.8,
           3X,'R1=',E15.8,3X,'RT=',E15.8)
61     FORMAT(3X,'PRINTING FROM CARD 2570',/,3X,'R=',E15.8,
           3X,'R1=',E15.8,3X,'RT=',E15.8)
1000  FORMAT(4X,F6.4,/,X,F10.5,3X,F10.5)
```

CHART TITLE - SUBROUTINE TOPLOT(W,Y,NP,TL,KR,LL)

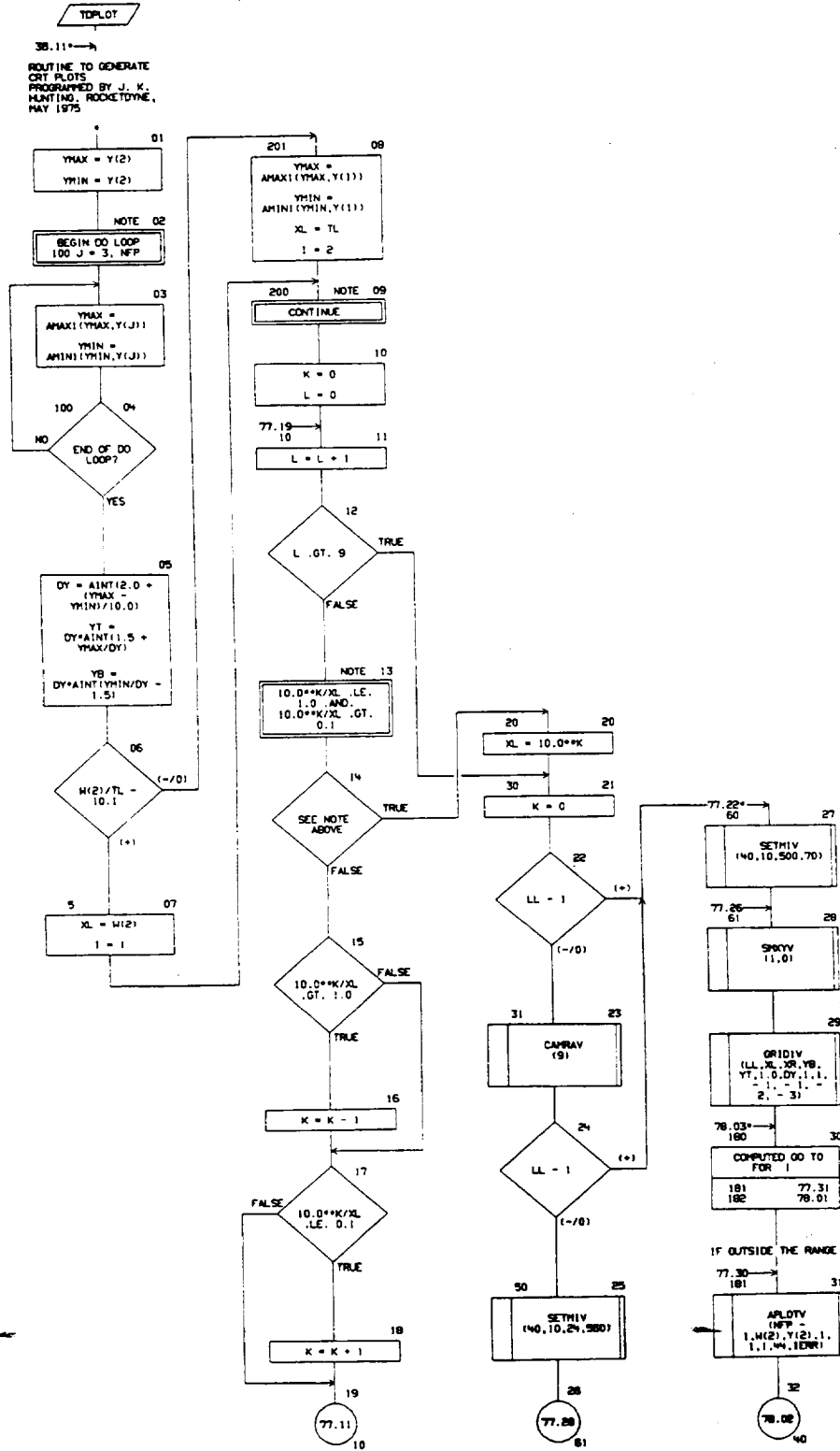


CHART TITLE - SUBROUTINE TOPLOT(W,Y,NP,TL,XR,LL)

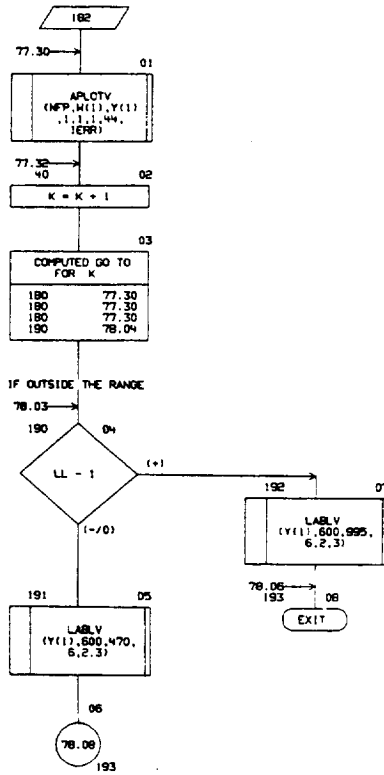




CHART TITLE - NON-PROCEDURAL STATEMENTS

DIMENSION M(101),Y(101)

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - FUNCTION XIMAGF(XIHONG)

ROUTINE TO COMPUTE  
NOZZLE ADM. DIFF AS  
FUNCTION OF  
IMAGINARY PART OF  
OMEGA  
PROGRAMMED BY K. W.  
FERTIG, ROCKWELL,  
MAY 1975

RECOMPUTE OMEGA USING  
NEW IMAGINARY PART.

01  
OMEGA =  
REAL(OMEGA) +  
(0., 1.) \* XIHONG  
IP = 0

NOTE 02  
(ISONT .NE. 1  
.AND. ISONT .NE.  
4) .OR. ISLP .NE.  
1

03  
SEE NOTE  
ABOVE

SAVE OMEGA THEN ALTER  
IMAG PART BY 0.1  
PERCENT TO COMPUTE  
DERIV.

04  
OMEGO = OMEGA  
DH =  
AMAX1(.001, AIMAG  
(OMEGA), 1)  
OMEGA = OMEGA +  
(0., 1.) \* DH

CHANDY COMPUTES  
UPSTREAM NOZZLE  
ADMITTANCE AND  
OSCILLATORY  
PROFILES.

80.03 → 10  
05  
1  
2  
0  
1  
CHANDY

06  
NOZA =  
NOZAMP\*(1. -  
DCSDR\*HR(INSP)  
\*2. \*GAMP/ICS\*P  
(INSP)\*\*GAMP -  
1.1))  
FN = (ONZA -  
NOZA)

NOTE 07  
IP .EQ. 0 .AND.  
(ISONT .EQ. 1  
.OR. ISONT .EQ.  
4) .AND. ISLP  
.EQ. 1

08  
SEE NOTE  
ABOVE

09  
FNR = REAL(FN)  
FNI = AIMAG(FN)  
HN = CABS(FN)  
XIMAGF = FNI

RESTORE OMEGA AND  
SAVE VALUES FOR  
COMPUTATION OF  
DERIVATIVES.

13  
OMEGA = OMEGO  
HND = CABS(FN)  
ON = FN

IF IP EQUALS ZERO AT  
THIS POINT, THEN  
DERIVATIVES WERE NOT  
REQUIRED, HENCE,  
RETURN WITHOUT  
COMPUTING THEM.

10  
IP .EQ. 0

11  
EXIT

COMPUTE  
D(ONZA-NOZA)  
/D( AIMAG(OMEGA) )  
CALLED ON, AND  
D(CABS(ONZA-NOZA))  
/D( AIMAG(OMEGA) )  
CALLED XIMAGF.

80.14 → 15  
30  
ON = (ON - FNI/DH)  
XIMAGF =  
(HND\*\*2 -  
HN\*\*2)/DH

16  
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```

COMPLEX OMEGA,NOZA,CHOZA,P,RHO,V,FR,T,VXO,FN,NOZAPR,GN,
COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
COX10, COX11, COX12, COX13, COX14, COX15, COX16,
CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
,OMEGA
REAL HBOX1, HBFU1, NUBOX, NUBFU, H4O
COMMON /CONCHV/ XKOX, XKFU, HBOX1, HBFU1, TALBOX, TALBFU, YBOX,
YBFU, G4MO, RGO, DELMOX, DELMFU, PC, CO,
COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
CFU12, CFU13, CFU14, CFU15, CFU16, H4O, XHMFU, XHMPX,
CS, DCSMR, DMR, DRGMR, ADVOX, ADDOX, TORAGO, DELVOX,
NUBOX, DTOMX, ADVFU, ADOFU, TORAGF, DELVFU, NUBFU, DTFUM
COMMON /FZERO/ NOZA,NOZAPR,GN,FN,FNR,FNI,HN,ISNT,ISLP
COMMON /CONCHV/ P(100),RHO(100),V(100),FR(100),T(100),
VXO,OMEGA,CHOZA,DELP
COMMON /COMARE/ NXP,X(100),X(100),A(100),DA(100),DELX,
XO,XNOZ,AINJ

```

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE ZADAMS(N,X,Y,DT,102)

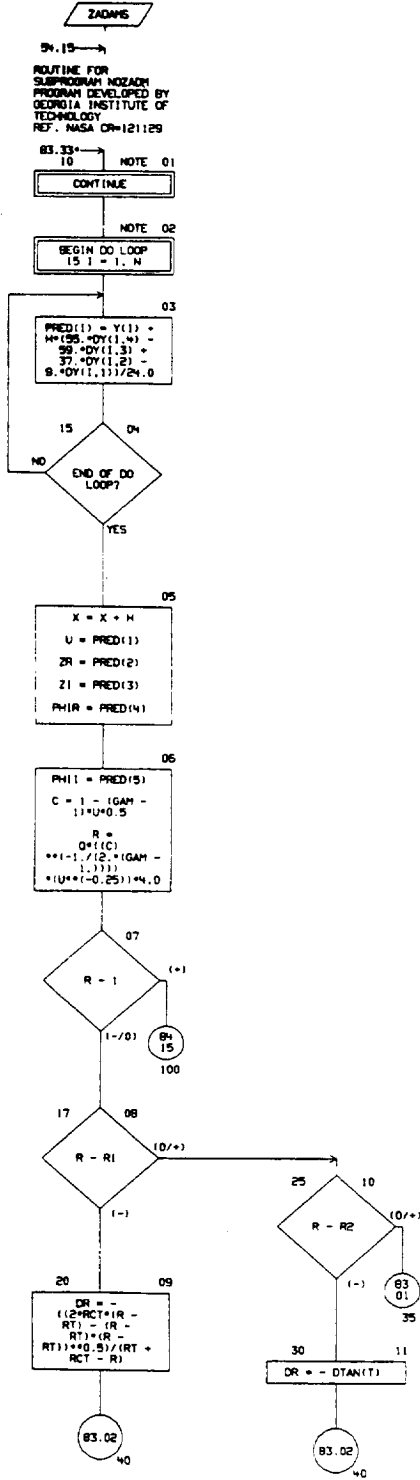


CHART TITLE - SUBROUTINE ZADAMS(N,H,X,Y,DY,IGZ)

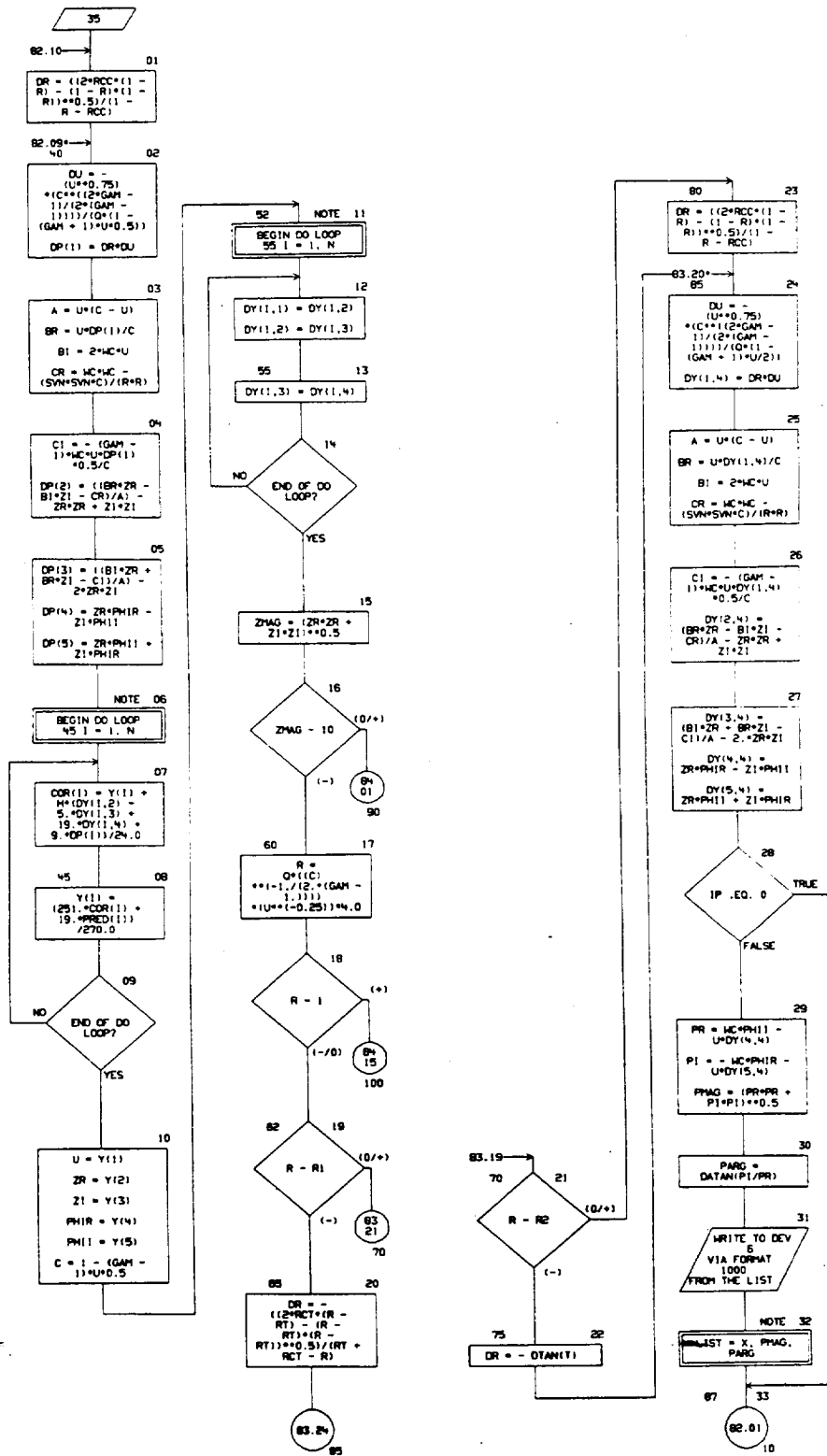


CHART TITLE - SUBROUTINE ZADWIS(N,H,X,Y,DY,IQZ)

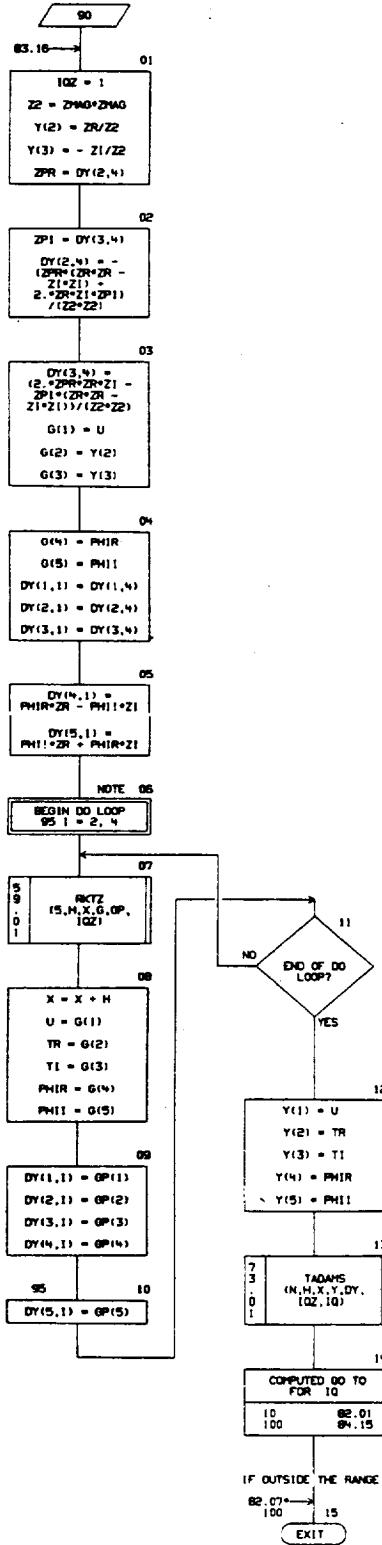


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      IMPLICIT REAL*(81A-H,O-Z)
      COMMON /X1/DAH,SVN,ANGLE,RCY,RCX /X2/T,RT,0,R1,R2,HC,IP
      /X3/CM
      DIMENSION COR(5), OP(5), DY(5,4), PRED(5), Y(5), G(5), GP(5)
16     FORMAT(3X,'PRINTING FROM CARD 3430',/,3X,'R=',E15.8,
           3X,'RT=',E15.8,3X,'RT=',E15.8)
81     FORMAT(3X,'PRINTING FROM CARD 3790',/,3X,'R=',E15.8,
           3X,'RT=',E15.8,3X,'RT=',E15.8)
1000    FORMAT(46X,F6.4,1X,F10.5,3X,F10.5)
```

ORIGINAL PAGE IS  
OF POOR QUALITY

CHART TITLE - SUBROUTINE ZERO(F1,T11,T21,F11,F21,ANS,FANS,EPSPF,EPSPX,AF,ICNT,NON

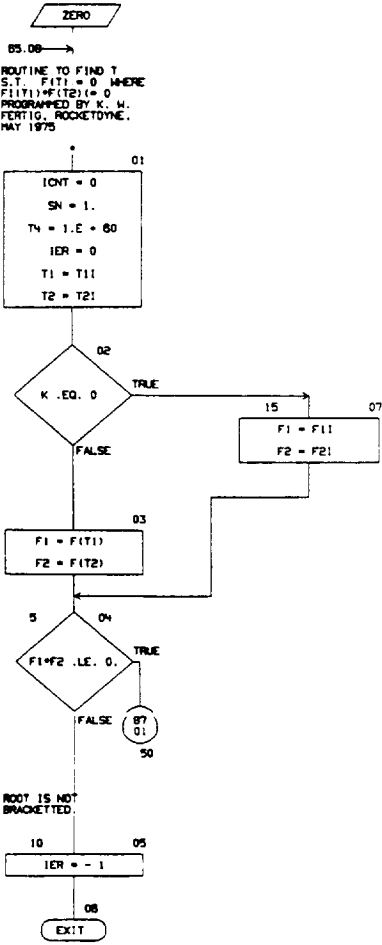




CHART TITLE - SUBROUTINE ZERD(F1,T1,T2,F1,F2,ANS,FANS,EPFX,EPFX,AF,ICNT,NON

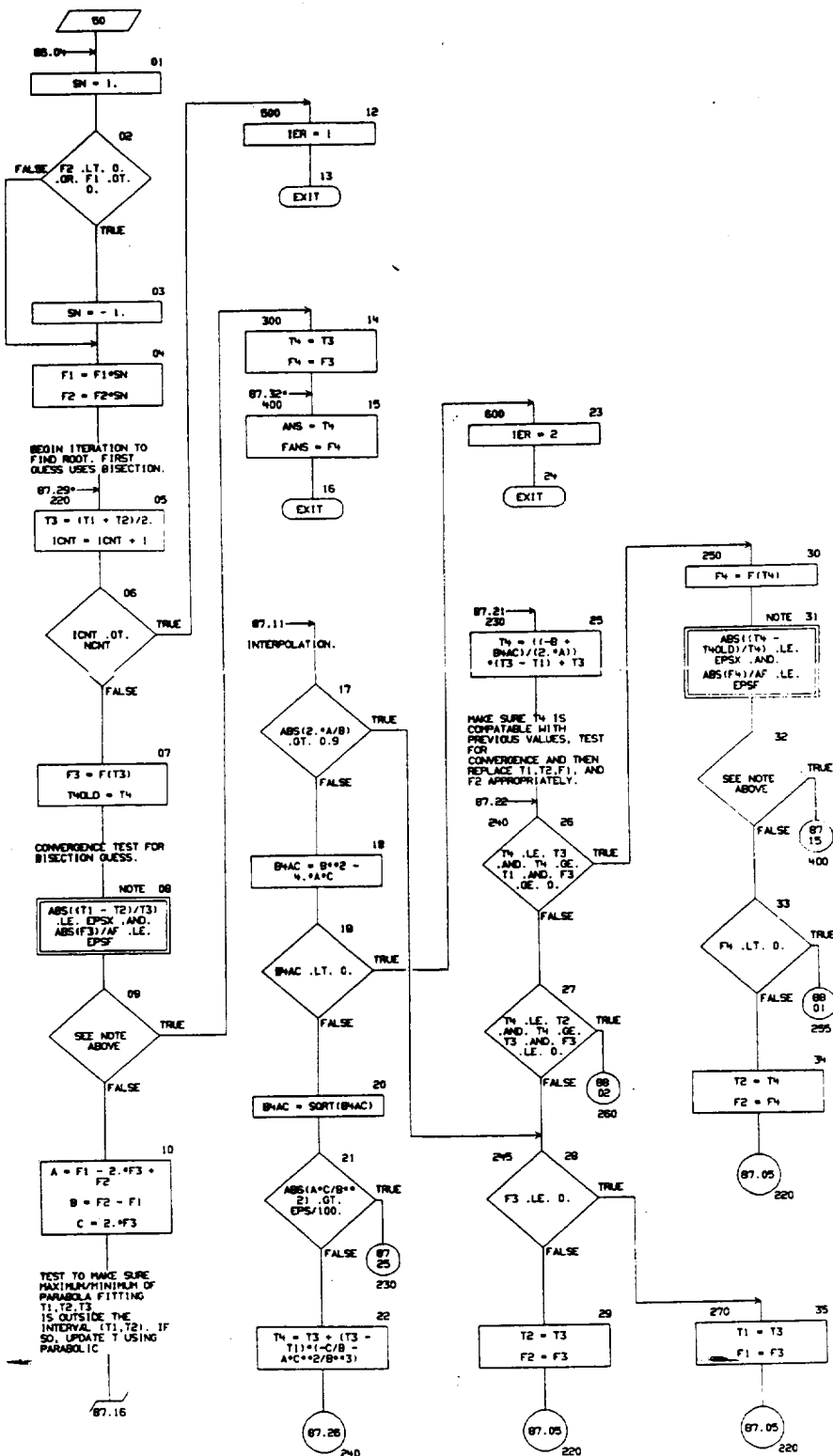
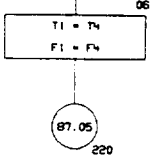
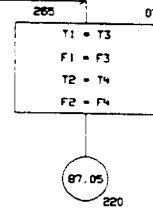
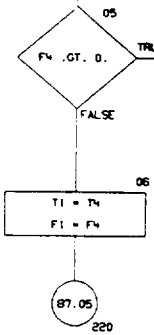
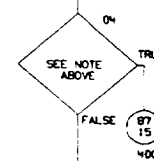
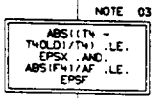
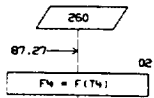
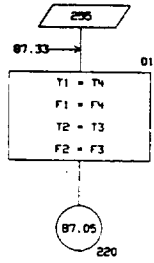


CHART TITLE - SUBROUTINE ZEND(F1,T11,T21,F11,F21,MS,FMS,EPF,EPX,AF,ICNT,NCN



06/25/75

AUTOLOM CHART SET - F3C3H COMPUTER PROGRAM

PAGE 89

CHART TITLE - NON-PROCEDURAL STATEMENTS

STATEMENT FUNCTION DEFINITION, F(X) = SINPI(X)

ORIGINAL PAGE IS  
OF POOR QUALITY

R-9808/8-101

END OF AUTOFLON CHART SET

3,387 INPUT STATEMENTS PROCESSED

EXECUTION TIME -

4 MIN 21 SEC

ORIGINAL PAGE IS  
OF POOR QUALITY

APPENDIX C

COMPUTER CODE LISTINGS

<u>INDEX</u>	<u>PAGE</u>
MAIN	C-1
AREA	C-12
CHAMDY	C-13
CHMCØN	C-20
COØAEL	C-24
CØMBDY	C-27
CØMMAT	C-32
FRESP	C-34
HEAD	C-48
HY (Block Data)	C-49
HYDRDY	C-50
LØCFAC	C-67
NØZADM	C-68
RKTDIF	C-75
RKTZ	C-77
RKZDIF	C-79
SØLVW	C-81
STEADY	C-94
TADAMS	C-98
TDPLØT	C-103
XIMAGF	C-105
ZADAMS	C-108
ZERØ	C-112

F S C S M MAIN PROGRAM

MAIN CONTROL PROGRAM FOR THE FEED SYSTEM COUPLED STABILITY MODEL  
 PROGRAM DEVELOPED BY ROCKETDYNE, A DIVISION OF ROCKWELL  
 INTEKNATIONAL, CANOGA PARK, CALIF 91304  
 PROGRAMMED BY M. D. SCHUMAN, ROCKETDYNE, MAY 1975

DIMENSION TITLE(18,2)

COMPLEX OMEGA, P, RHO, V, MR, T, CNOZA, VXO,  
 1 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,  
 2 COX10, COX11, COX12, COX13, COX14, COX15, COX16,  
 3 CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,  
 4 CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16,  
 5 NOZA, GINJOX, GINJFU,  
 6 NOZAT, GINJOT, GINJFI,  
 7 FN, NOZAMR, GN,  
 8 COXT(16), CFUT(16),  
 9 COXTT(16,2), CFUTT(16,2)

REAL MBOXI, MBFUI, MRB, MRGI, MGI, NUBOX, NUHFU, KF, KO, L, MWG

COMMON /COMCBM/ XKOX, XKFU, MBOXI, MBFUI, TAUBOX, TAUBFU, VBOX,  
 1 VBFU, GAMU, XGO, DELHOX, DELHFU, PC, CO,  
 2 COX1, COX2, COX3, COX4, COX5, CUX6, COX7, COX8, COX9, COX10,  
 3 COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,  
 4 CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,  
 5 CFU12, CFU13, CFU14, CFU15, CFU16, MWG, XIMPFU, XIMPOX,  
 6 CS, DCSOMR, DHDMR, DRGDMR, ADVOX, ADDOX, TDRAGO, DELVOX,  
 7 NUBOX, DTUXDM, ADVFU, ADDFU, TDRAGF, DELVDFU, NUBFU, DTFUDM

EQUIVALENCE (COXT(1),COX1), (CFUT(1),CFU1)

CCCCCG1G  
 CCCCCC2G  
 00000030  
 0000004G  
 00000050  
 0000006G  
 00000070  
 0000008G  
 0000009G  
 0000010G  
 0000011G  
 0000012G  
 0000013G  
 0000014G  
 0000015G  
 0000016G  
 0000017G  
 0000018G  
 0000019G  
 0000020G  
 0000021G  
 0000022G  
 0000023G  
 0000024G  
 0000025G  
 0000026G  
 0000027G  
 0000028G  
 0000029G  
 0000030G  
 0000031G  
 0000032G  
 0000033G

F S C S M MAIN PROGRAM

```

C      COMMON /CONSTS/ MRB(100), TB(100), RHDH(100), VB(100),
1      DRHB(100), DRHDB(100), DVB(100), VAPBOX(100), VAPBFU(100),
2      SSV1(100), SSV2(100), SSV3(100), SSV4(100),
3      SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9OX(100),
4      SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),
5      SSV14(100), SSV15(100), SSV16(100),
6      RHUGI, VGI, MRGI, MGI
C
C      COMMON /COMCHM/ P(100), RHD(100), V(100), MR(100), T(100),
1      VXU, OMEGA, CNOZA, DELP
C
C      COMMON /COMARE/ NXP, X(100), XM(100), A(100), DA(100), DELX,
1      XO, XNOZ, AINJ
C
C      COMMON /COMNOZ/ RLCX, RCTX, ANGLEX, CRR, RINJ, INPNZ, FREQMX,
1      FREQMI, IPRNOZ
C
C      COMMON /FZERO/ NOZA,NOZAMR,GN,FN,FNR,FNI,HN,ISCNT,ISLP
C
C      COMMON /DUMP/ IWRT
C
C      DATA JK/1/, JKK/1/, PI/3.141593/
C
C      COMMON /SOLVE/ FREQ, DELFRQ, DELMX, EPSF, EPSX, EPSFS, EPSXS,
1      FRQMAX, CTEST, IPASS, KNTR, ISTRT, KSCNT4, IWSKP,
2      KNTMX, KNTSMX, KNTRMX
C
C      COMMON /COMTAP/ NFREQT, FREQT(100), NOZAT(100), GINJOT(100),
1      GINJFT(100), ITAPN, ITAPC, ITAPH
C
C      COMMON /HY/ ICT,IRFLAG,ITERM,ITYPE,IPRHYD,IX,AA(30),CW(30),KF,KD,00000650
1      L(30),R(30),RHOL(30),VV(30),VF,VO,VOLF,VOLO,ZF,ZO,OSAVE(188),00000660

```

F S C S M MAIN PROGRAM

```

C1 2 FSAVE(166)
C 1 CONTINUE
C CALL HEAD
C READ AND WRITE INPUT DATA
C READ(5,490,END=9000) ((TITLE(I,J),I=1,18),J=1,2)
C 490 FORMAT(16A4)
C WRITE(6,900) ((TITLE(I,J),I=1,18),J=1,2)
C 900 FORMAT(1H1,/,/,26X,'FEED SYSTEM COUPLED STABILITY MODEL',/,
C 1 9X,18A4,/,9X,16A4)
C READ(5,500,END=4000) INPHYD, INPCOM, INPNOZ, ITAPH, ITAPC, ITAPN,
C 1 IPRHYD, IPRCOM, IPRNUZ, IPRCHM, IPRSTE, NXP
C 500 FORMAT(12I6)
C WRITE(6,600) INPHYD, INPCOM, INPNOZ, ITAPH, ITAPC, ITAPN,
C 1 IPRHYD, IPRCOM, IPRNOZ, IPRCHM, IPRSTE, NXP
C 600 FORMAT(/,5X,INPHYD =,12,5X,INPCOM =,12,5X,INPNOZ =,12,
C 1 5X,ITAPH =,13,5X,ITAPC =,13,5X,ITAPN =,13,/,5X,
C 2 IPRHYD =,12,5X,IPRCOM =,12,5X,IPRNOZ =,12,5X,
C 3 IPRCHM =,12,5X,IPRSTE =,12,5X,IXP =,14)
C READ(5,510) X0, XNOZ, RINJ, GAMO, CU, DELP
C 510 FORMAT(6E12.8)
C WRITE(6,610) X0, XNOZ, RINJ, GAMO, CO, DELP
C 610 FORMAT(/,5X,X0 =,1PE11.4,7X,XNOZ =,E11.4,5X,RINJ =,
C 1 E11.4,5X,GAMO =,E11.4,/,27X,CO =,E11.4,7X,DELP =,
C 2 E11.4)
C X0 = X0*0.0254
C XNOZ = XNOZ*0.0254
C RINJ = RINJ*0.0254
C CO = CO*0.3048
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080

```



F S C S M MAIN PROGRAM

```

C / READ(5,500) NROOT, IWRT, IWSKP, KNTMX, KNTRMX, KNTSMX
WRITE(6,620) NROOT, IWRT, IWSKP, KNTMX, KNTRMX, KNTSMX
620 FORMAT(/,5X,'NROOT =',I3,5X,'IWRT =',I2,7X,'IWSKP =',I2,6X,
      'KNTRMX =',I4,4X,'KNTRMX =',I4,3X,'KNTSMX =',I4)
C / READ(5,510) OMEGA, FRQMAX, DELFRQ, DELMX, CTEST,
      EPSF, EPSX, EPSFS, EPSXS
WRITE(6,630) OMEGA, FRQMAX, DELFRQ, DELMX, CTEST, EPSF, EPSX,
      EPSFS, EPSXS
630 FORMAT(/,5X,'OMEGA(R) =',1PE11.4,2X,'OMEGA(I) =',E11.4,2X,
      'FRQMAX =',E11.4,3X,'DELFRO =',E11.4,/,27X,'DELMX =',
      E11.4,4X,'CTEST =',E11.4,/,5X,'EPSF =',E11.4,5X,
      'EPSX =',E11.4,5X,'EPSFS =',E11.4,4X,'EPSXS =',E11.4)
C / READ(5,510) PC, MBOXI, MBFUI
WRITE(6,650) PC, MBOXI, MBFUI
650 FORMAT(/,5X,'PC =',1PE11.4,7X,'MBOXI =',E11.4,4X,'MBFUI =',
      E11.4)
      PCIN = PC
      WGIN = MBOXI
      WFIN = MBFUI
      PC = PC*6894.76
      MBOXI = MBOXI*0.453592
      MBFUI = MBFUI*0.453592
C / IF(NROOT.LT.0) OMEGA = 2.*PI*REAL(OMEGA) + (0.,-1.)*
      REAL(OMEGA)*AIMAG(OMEGA)
      NROOT = IABS(NROOT)
      VGI = 0.0
      MGI = 0.0
C /
      C

```

```

00001090
00001130
00001140
00001150
00001160
00001170
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520

```

F S C S M MAIN PROGRAM

```

C
C
C COMPUTE AREA AND DISTANCE PROFILES.
C
C AINJ = PI*RIJ*RIJ
C CALL AREA
C
C READ IN AND COMPUTE FREQUENCY TABLE IF INPNOZ .LE. 3
C
C IF(INPNOZ.GT.3) GO TO 1060
C READ(5,520) NFREQT, FREQMI, FREQMX
C 520 FORMAT(112,5E12.8)
C WRITE(6,640) NFREQT, FREQMI, FREQMX
C 640 FORMAT(/,5X,'NFREQT =',14,10X,'FREQMI =',1PE11.4,3X,
C 1 'FREQMX =',E11.4)
C DELF = (FREQMX-FREQMI)/(NFREQT-1)
C DO 1050 I=1,NFREQT
C FREQ(I) = FREQMI+(I-1)*DELF
C 1050 CONTINUE
C
C SET FLAG (IR=1) TO INDICATE FIRST PASS.
C
C 1060 IR = 1
C
C ISCVT = 1 : FREQUENCY BEING INCREMENTED TO ZERO IMAG F
C 2 : FREQUENCY STEP TO START TO START LOOP WITHIN LOOP
C 3 : LOOP WITHIN LOOP
C 4 : STARTING SECANT METHOD
C 5 : USING SECANT METHOD
C
C OMEGA = UMEGA - 31.4
C
C BEGIN DO LOOP TO SEARCH FOR OMEGAS THAT SATISFY BOUNDARY
C CONDITION: UPSTREAM NOZZLE ADMITTANCE = DOWNSTREAM NOZZLE
C ADMITTANCE.

```

```

00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880

```

F S C S M MAIN PROGRAM

```

C      A      DO 5000 NRT = 1,NROOT
C
C      CHANGE OMEGA BY 5 HERTZ TO START SEARCH FOR NEXT ROOT.
C
C      OMEGA = OMEGA + 31.4
C
C      INITIALIZE FLAGS AND COUNTERS.
C
C      ISCNT = 1
C      IPASS = 1
C      KNTK = 0
C      ISTRT = 0
C      KSCNT4 = 0
C      FREQ = REAL(OMEGA)/(2.*PI)
C
C      THIS IS RETURN POINT FOR ITERATION TO FIND FREQUENCIES THAT
C      BRACKET A ROOT.
C
C      10 OMEGA = 2.*PI*FREQ + (0.,1.)*AIMAG(OMEGA)
C      IF(FREQ.GT.FRQMAX) GO TO 1
C
C      ISLP = 1 SETS FLAG TO COMPUTE DERIVATIVES IF ISCNT = 1 OR 4
C
C      ISLP = 1
C
C      15 IF((INPHYD.GE.3.OR.INPCOM.GE.3.OR.INPNOZ.GE.4)
C      1 .AND.(IR.LE.6.OR.INPNOZ.LE.3))
C      2 CALL LOCFAC(JKK,FREQ,FREQ,NFREQ,I1,F1)
C
C      COMPUTE DOWNSTREAM NOZZLE ADMITTANCE PARAMETER, NOZAMR.
C      (THROUGH STATEMENT 40)

```

```

00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210

```

F S C S M MAIN PROGRAM

```

IF(INPNOZ.GT.3) GO TO 20
CALL NOZADM(IR,CAMO,CO,FREQ,NOZAMR)
IF(INPNUZ.LE.1) GO TO 40
IF(IPRNOZ.GT.0) WRITE(6,900) ((TITLE(I,J),I=1,18),J=1,2)
GO TO 25
20 IF(IR.LE.0) GO TO 30
REWIND ITAPN
READ(ITAPN) NFREQ
DO 22 I=1,NFREQ
READ(ITAPN) FREQ(I), NOZAT(I)
22 CONTINUE
25 CALL LOGFAC(JK,FREQ,FREQ,NFREQ,I1,F1)
30 NOZAMR = NOZAT(I1)+F1*(NOZAT(I1+1)-NOZAT(I1))
40 CONTINUE

C
C COMPUTE FEED SYSTEM RESPONSE PARAMETERS, GINJOX AND GINJFU.
C (THROUGH STATEMENT 70)
C
IF(INPHYD.GT.2) GO TO 50
CALL HYDROY(IR,INPHYD,FREQ,GINJOX,GINJFU,PCIN,WGIN,WFIN)
IF(INPHYD.GT.1) GO TO 45
GINJOX = REAL(GINJOX)*CEXP((0.0,-1.0)*AIMAG(GINJOX)*PI/180.)
GINJFU = REAL(GINJFU)*CEXP((0.0,-1.0)*AIMAG(GINJFU)*PI/180.)
GO TO 70
45 IF(IPRHYD.GT.0) WRITE(6,900) ((TITLE(I,J),I=1,18),J=1,2)
50 IF(IR.LE.0) GO TO 60
REWIND ITAPH
READ(ITAPH) (GINJOT(I),GINJFT(I),I=1,NFREQ)
DO 55 I=1,NFREQ
GINJOT(I) = REAL(GINJOT(I))*CEXP((0.0,-1.0)*
1 AIMAG(GINJOT(I))*PI/180.)
GINJFT(I) = REAL(GINJFT(I))*CEXP((0.0,-1.0)*
1 AIMAG(GINJFT(I))*PI/180.)

```

F S C S M MAIN PROGRAM

```

55 CONTINUE
60 GINJOX = GINJOT(I1)+F1*(GINJOT(I1+1))-GINJOT(I1)
GINJFU = GINJFT(I1)+F1*(GINJFT(I1+1))-GINJFT(I1)
70 CONTINUE
C
C      COMPUTE COMBUSTION COEFFICIENTS. (THROUGH STATEMENT 110)
C
      IF(INPCOM.GT.2) GO TO 80
      CALL CUMBDY(IR,FREQ,GINJOX,GINJFU,IIPRCOM,INPCOM)
      IF(INPCOM.LE.1) GO TO 110
80 IF(IR.LE.0) GO TO 82
      REWIND ITAPC
      READ(ITAPC) XKOX, TAUBOX, VROX, DELHOX, TLRAGO, ADVOX,
1      ADDOX, UELVOX, NUBOX, DTUXDM, XIMPOX, XKFU, TAUBFU,
2      VBFU, DELHFU, TORAGF, ADVFU, ADDFU, DELVFU, NURFU,
3      DTFUUM, XIMPFU, MMG, CS, DRGDMR, DCSDMR, DHDMR, RGO
      110 = -1
82 NI = 1
      IF(I1-110) 84, 96, 92
84 NI = 110-I1+1
86 DO 88 I=1,NI
88 BACKSPACE ITAPC
      READ(ITAPC) (COXTT(J,1),CFUTT(J,1),J=1,16)
90 READ(ITAPC) (COXTT(J,2),CFUTT(J,2),J=1,16)
      GO TO 96
92 IF(I1.EQ.110+1) GO TO 86
      NI = 110+2
      DO 94 I=NI,I1
94 READ(ITAPC) (COXTT(J,1),CFUTT(J,1),J=1,16)
      GO TO 90
96 110 = I1
      DO 100 J=1,16
      COXT(J) = COXTT(J,1)+F1*(COXTT(J,2))-COXTT(J,1)

```

```

00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820

```

F S C S M MAIN PROGRAM

```

100 CFUT(J) = CFUTT(J,1)+F1*(CFUTT(J,2)-CFUTT(J,1))
110 CONTINUE
C
C COMPUTE STEADY STATE PROFILES IF THIS IS FIRST PASS.
C
IF(IR.GT.0) CALL STEADY(IPRSTE)
IR = 0
VXC = 0.0
C
CALL SULVW TO FINISH COMPUTATIONS FOR THIS OMEGA/FREQUENCY.
UPDATE OMEGA TO NEW GUESS IF ISCNT = 4 OR 5 AND CONVERGENCE HAS
NOT BEEN REACHED. WHEN ISCNT=1 OR 4, TWO PASSES WILL BE MADE PAST
THIS POINT(ONE WITH ISLP=1 AND ONE WITH ISLP=0) IN ORDER TO
COMPUTE JACOBIAN.
C
CALL SULVW(KWHERE)
GO TO (10,15,225,225,5000), KWHERE
C
CONVERGENCE HAS BEEN REACHED. PRINT RESULTS
(THROUGH STATEMENT 300).
C
225 CONTINUE
C
IF(NRT.LE.1.OR.IPRCHM.GT.0.OR.IPRCOM.GT.0)
1 WRITE(6,900) (TITLE(I,J),I=1,18),J=1,2)
IF(IPRCOM.LE.0) GO TO 1090
WRITE(6,1080)
1080 FORMAT(///,2GX,'COMBUSTION DYNAMIC COEFFICIENTS',/)
DO 1085 I=1,16
WRITE(6,1061) I, COXT(I), I, CFUT(I)
1081 FORMAT(5X,'COX(',I2,') =',1PE11.4,';',E11.4,5X,'CFU(',I2,
I ' ) =',E11.4,';',E11.4)

```

```

00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950
00002960
00002970
00002980
00002990
00003000
00003010
00003020
00003030
00003040
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150

```

F S C S M MAIN PROGRAM

```

1085 CONTINUE
C
1090 CONTINUE
DEC = -AIMAG(OMEGA)/REAL(OMEGA)
WRITE(6,2000) FREQ, DEC, NOZA, GINJUX, GINJFU
2000 FORMAT(//,10X,'FREQUENCY =',F8.2,' HZ,',/,10X,
1 'DECREMENT =',F8.5,/,
2 10X,'NOZZLE ADMITTANCE =',F9.5,':',2X,F9.5,/,10X,
3 'FEED SYSTEM RESPONSE',/,20X,'OXIDIZER =',F9.5,':',2X,F9.5,
4 /,24X,'FUEL =',F9.5,':',2X,F9.5,///)
C
IF(IPRCHM.LE.0) GO TO 5000
IF(IPRCOM.GT.0) WRITE(6,90C) ((TITLE(I),J),I=1,18),J=1,2)
WRITE(6,2010)
2010 FORMAT(10X,4(11X,'OSCILLATORY'),/,5X,'DISTANCE',6X,
1 'PRESSURE RATIO',7X,'VELOCITY RATIO',6X,
2 'TEMPERATURE RATIO',7X,'MIXTURE RATIO',/,5X,'(INCHES)',
3 4(5X,'AMPLITUDE PHASE'),/)
DD 300 I=1,NXP
AP = CABS(P(I))
IF(AP.LE.0.0.AND.I.LT.NXP) P(I)=P(I+1)*1.E-10
IF(AP.LE.0.0.AND.I.GE.NXP) P(I)=P(I-1)*1.E-10
PV = ATAND(AIMAG(P(I)),REAL(P(I)))
AV = CABS(V(I))
IF(AV.LE.0.0.AND.I.LT.NXP) V(I)=V(I+1)*1.E-10
IF(AV.LE.0.0.AND.I.GE.NXP) V(I)=V(I-1)*1.E-10
PV = ATAND(AIMAG(V(I)),REAL(V(I)))
AT = CABS(T(I))
IF(AT.LE.0.0.AND.I.LT.NXP) T(I)=T(I+1)*1.E-10
IF(AT.LE.0.0.AND.I.GE.NXP) T(I)=T(I-1)*1.E-10
PT = ATAND(AIMAG(T(I)),REAL(T(I)))
AMR = CABS(MR(I))
IF(AMR.LE.0.0.AND.I.LT.NXP) MR(I)=MR(I+1)*1.E-10

```

F S C S M MAIN PROGRAM

```

1  IF (AMR.LE.0.0.AND.I.GE.NXP) MR(I)=MR(I-1)*1.E-10
    PMR = ATAND(AIMAG(MR(I)),REAL(MR(I)))
    XP = X(I)/0.0254
    WRITE(6,2020) XP, AP, PP, AV, PV, AT, PT, AMR, PMR
2020 FORMAT(5X,F8.4,4(5X,F8.5,2X,F7.2))
    300 CONTINUE
C
C      END OF DO LOOP TO FIND OMEGAS SATISFYING BOUNDARY CONDITIONS.
C
C      5000 CONTINUE
C      GO TO 1
C
C      9000 CALL EXIT
C      STOP
C      END
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620

```





F S C S M SUBROUTINES

```

SUBROUTINE CHAMDY
SUBPROGRAM CALCULATES THE OSCILLATORY PRESSURE, TEMPERATURE,
VELOCITY, MIXTURE RATIO, AND DENSITY PROFILES PLUS THE
UPSTREAM NOZZLE ADMITTANCE FROM THE CHAMBER DYNAMIC EQUATIONS
PROGRAMMED BY K. W. FERTIG AND M. D. SCHUMAN, RCKETDYNE, MAY 1975
COMPLEX*16 AMA(4,5), CMA(4)
COMPLEX OMEGA, P, RHO, V, MR, T, CNOZA, VXO,
1 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
2 COX10, COX11, COX12, COX13, COX14, COX15, COX16,
3 CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
4 CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
COMPLEX DM1, DM3, DM4, DM5, DM6, DM7OX, DM8OX, DM8OX,
1 DM8FU, G1OX, G1FU, G2OX, G2FU, G3OX, G3FU, G4OX, G4FU,
2 RBSOX, RBSFU, PINTOX, PINTFU, RHNTOX, RHNTFU, MRNTOX,
3 MRN1FU, VINTOX, VINTFU, II
4 ,DM9OX,DM9FU
REAL MEXI, MBFUI, MRB, MRGI, MGI, NUBOX, NUBFU, MWG
COMMON /COMCBM/ XKOX, XKFU, MBOXI, MBFUI, TAUBOX, TAUBFU, VBOX,
1 VbFU, GAMO, KGO, DELHOX, DELHFU, PC, CU,
2 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
3 COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
4 CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
5 CFU12, CFU13, CFU14, CFU15, CFU16, MWG, XIMPFU, XIMPOX,
6 CS, UCSLMR, DHDMR, DRGDMR, ADVOX, ADDOX, TDRAGO, DELVOX,
7 NUBOX, D1OXDM, ADVFU, ADDFU, TDRAGF, DELVFU, NUBFU, DTFUDM
COMMON /CONSTS/ MRB(100), TB(100), RHOB(100), VB(100),

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330

```

F S C S M SUBROUTINES

```

1  DMRB(100), DRHOB(100), DVB(100), VAPBOX(100), VAPBFU(100),      00000340
2  SSV1(100), SSV2(100), SSV3(100), SSV4(100),                    00000350
3  SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9CX(100),      00000360
4  SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),  00000370
5  SSV14(100), SSV15(100), SSV16(100),                            00000380
6  RHOGI, VGI, MRGI, MGI                                          00000390
C
COMMON /COMCHM/ P(100), RHO(100), V(100), MK(100), T(100),      00000400
1  VX0, OMEGA, CNOZA, DELP                                        00000410
C
COMMON /COMARE/ NXP, X(100), XM(100), A(100), DA(100), DELX,    00000420
1  XG, XNOZ, AINJ                                              00000430
C
COMMON /ADARND/ G10X,G20X,G30X,G40X,G1FU,G2FU,G3FU,G4FU,PINTOX,  00000440
1  RHNTOX,MRNTOX,VINTOX,PINTOX,RHNTFU,RHNTFU,VRNNTFU,VINTFU,RBSOX,  00000450
2  RBSFU,DM1,DM3,DM4,DM5,DM6,DM7OX,DM8OX,DM9OX,DM7FU,          00000460
3  DM8FU,DM9FU,II,DM2,DM22,RHOINJ                               00000470
C
COMMON /DUMP/ IWRT                                             00000480
C
EQUIVALENCE (AMA(1,5),CMA(1))                                  00000490
C
NXPM1 = NXP-1                                                  00000500
C
COMPUTE BOUNDARY CONDITIONS AT X=X0.                             00000510
C
RHOINJ = DELP/GAMO                                             00000520
DM2 = CU**2*RHOGI/(GAMO*PC)                                    00000530
DM22 = SQR(DM2)                                                00000540
DM1 = CEXP((0.,-1.)*OMEGA/CO*DM2*VGI/CO*XG)                   00000550
DM3 = DM22*OMEGA*X0/CO                                         00000560
P(1) = DM1*(DELP*CCOS(DM3)+ (0.,1.)*GAMO*DM22*VX0*CSIN(DM3))  00000570
RHO(1) = P(1)/GAMO                                             00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660

```

F S C S M SUBROUTINES

```

V(1) = DM1*((0.,1.)*DEL P/(GAMU*DM22)*CSIN(DM3)+VX0*CCOS(DM3))
DM1 = CEXP((0.,1.)*OMEGA*X0/VBOX)
DM3 = CEXP((0.,1.)*OMEGA*X0/VBFU)
DM4 = DELP*(COX1*DM1-CFU1*DM3) + P(1)*(COX2-CFU2)
      + DELP/GAMU*(COX3*DM1-CFU3*DM3) + RHO(1)*(COX4-CFU4)
      + VX0*(COX7*DM1-CFU7*DM3) + V(1)*(COX8-CFU8)
DM2 = (MRGI+1.)*MRGI*MBFUI/(A(1)*RHOGI*CO)
DM4 = DM2*DM4 - V(1)*MRGI*(MRGI+1.)*(1.-MRGI*MBFUI/MBOXI)/
      (MRGI+2.)
DM5 = DM2*(COX5*DM1-CFU5*DM3 + COX6-CFU6)
DM5 = MBFUI*(MRGI+1.)/(A(1)*RHOGI*CO) - (0.,1.)*OMEGA*TAUBFU
      *VBFU/CO - DM5
MR(1) = DM4/DM5
T(1) = P(1) - RHO(1) - MR(1)*DRGDMR/(RGO*(1.
      + DRGDMR*(MRGI-MBOXI/MBFUI)/RGO))
1
C
C
C
INITIALIZE INTEGRALS USED IN VAPORIZATION EXPRESSION.
PINTOX = (0.,0.)
RHNTOX = (0.,0.)
MRNTOX = (0.,0.)
VINTOX = (0.,0.)
PINTFU = (0.,0.)
RHNTFU = (0.,0.)
MRNTFU = (0.,0.)
VINTFU = (0.,0.)
II = (0.,1.)
C
C
I = 1
IF(IWRT.GT.0) WRITE(6,9000) OMEGA,I,P(1),RHO(1),MR(1),V(1),T(1)
9000 FORMAT(///' OMEGA = ',1P2E13.5//3X,'I',11X,'P',19X,'RHO',
1 2LX,'MR',2LX,'V',21X,'T'//1X,13,1P1E11.4)
C
C

```

```

CCCC0670
CC0C06b0
CCCC0690
0C000700
CCCC0710
C00C0720
CCCC0730
CC0C0740
CCCC0750
CCCC0760
000C0770
000C0780
0CC00790
0CC0080C
000C0810
000C0820
0CCC0830
CC0C0840
000C0850
0CC00860
0CC00870
0CC00880
0CC00890
0CC00900
0CC00910
0CC00920
0CC00930
0CC00940
CC0C0950
0C0C0960
CCCC0970
0C000980
0C0C0990

```

F S C S M SUBROUTINES

```

C      10 CONTINUE
C      CHMCON COMPUTES MOST OF THE VARIABLES IN /ADARNDF/
C      CALL CHMCON(I)
C      DM1 = -11*OMEGA*DELX/CO
C      ENERGY EQUATION
C      AMA(1,1) = DM1 + 2.*SSV1(I)+SSV7(I)-SSV2(I)*(SSV5(I))*G10X
C      +SSV6(I)*G1FU)
C      AMA(1,2) = -SSV2(I)*(SSV5(I))*G20X+SSV6(I)*G2FU)
C      AMA(1,3) = 2.*GAMO + SSV8(I) - SSV2(I)*(SSV5(I))*G40X
C      +SSV6(I)*G4FU)
C      AMA(1,4) = -SSV2(I)*(SSV5(I))*G30X+SSV6(I)*G3FU - SSV90X(I)
C      + SSV9FU(I))
C      CMA(1) = SSV2(I)*(SSV5(I))*RBSUX + SSV6(I)*RESFU - SSV90X(I))*
C      MK(I) + SSV9FU(I))*MR(I))
C      1 - P(I)*(DM1-2.*SSV1(I)+SSV7(I))
C      2 - V(I)*(-2.*GAMO + SSV8(I))
C      3
C      CONTINUITY EQUATION
C      AMA(2,1) = -SSV10(I)*(G10X+G1FU)
C      AMA(2,2) = DM1 + 2.*SSV1(I) + SSV4(I) - SSV10(I)*(G20X+G2FU)
C      AMA(2,3) = 2. + SSV11(I) - SSV10(I)*(G40X+G4FU)
C      AMA(2,4) = -SSV10(I)*(G30X+G3FU)
C      CMA(2) = SSV10(I)*(RBSUX+RBSFU) - DM1*RHO(I) + 2.*V(I)
C      - V(I)*SSV11(I) + 2.*SSV1(I)*RHO(I) - RHO(I)*SSV4(I)
C      MOMENTUM EQUATION

```

00001000  
00001010  
00001020  
00001030  
00001040  
00001050  
00001060  
00001070  
00001080  
00001090  
00001100  
00001110  
00001120  
00001130  
00001140  
00001150  
00001160  
00001170  
00001180  
00001190  
00001200  
00001210  
00001220  
00001230  
00001240  
00001250  
00001260  
00001270  
00001280  
00001290  
00001300  
00001310  
00001320

F S C S M SUBROUTINES

```

C      I
C      AMA(3,1) = SSV12(I)
C      AMA(3,2) = (0.,0.)
C      AMA(3,3) = DM1 + 2.*SSV1(I) + SSV3(I)
C      AMA(3,4) = (0.,0.)
C      CMA(3) = V(I)*(-DM1-SSV3(I)+2.*SSV1(I) + SSV12(I))*P(I)
C
C      MIXTURE RATIO EQUATION
C
C      AMA(4,1) = -SSV13(I)*(G1OX-MRB(I))*G1FU)
C      AMA(4,2) = SSV14(I) - SSV13(I)*(G2OX-MRB(I))*G2FU)
C      AMA(4,3) = DELX*DMRB(I) - SSV13(I)*(G4OX-MRB(I))*G4FU)
C      AMA(4,4) = DM1 + 2.*SSV1(I) - SSV13(I)*(G3OX-MRB(I))*G3FU)
C      - SSV15(I)
C      CMA(4) = SSV15(I)*(RBSOX-MRB(I))*RBSFU) + MR(I)*(SSV15(I)
C      -DM1 + 2.*SSV1(I)) - DELX*DMRB(I))*(SSV1(I))*RHO(I)
C      + V(I))
C
C      SOLVE 4 BY 4 SYSTEM OF COMPLEX EQUATIONS
C      CALL COMMAT(AMA,4,4)
C
C      IP1 = I+1
C
C      P FROM ENERGY EQUATION.
C      RHO FROM CONTINUITY EQUATION.
C      RHO(IP1) = CMA(1)
C
C      V FROM MOMENTUM EQUATION.

```

```

00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650

```

F S C S M SUBROUTINES

```

C          V(IPI) = CMA(3)
C /
C          MR FROM MIXTURE RATIO EQUATION.
C
C          MR(IPI) = CMA(4)
C
C          T FROM EQUATION OF STATE.
C
C          T(IPI) = P(IPI)-RHO(IPI)-SSV16(I)*MR(IPI)
C          IF(IWRT.GT.0) WRITE(6,9100) IPI,P(IPI),RHO(IPI),MR(IPI),
C          1 V(IPI),T(IPI)
C          9100 FORMAT(IX,I3,I10E11.4)
C
C          UPDATE INTEGRALS FOR NEXT STEP.
C
C          DM70X = 2.*DM70X*CEXP(-II*OMEGA*X(I)/V60X)
C          PINTUX = PINTUX + DM70X*(P(I)*DM80X+P(IPI)*DM90X)
C          RHNTOX = RHNTOX + DM70X*(RHO(I)*DM80X+RHO(IPI)*DM90X)
C          MRNTUX = MRNTUX + DM70X*(MR(I)*DM80X+MR(IPI)*DM90X)
C          VINTUX = VINTUX + DM70X*(V(I)*DM80X+V(IPI)*DM90X)
C          DM7FU = 2.*DM7FU*CEXP(-II*OMEGA*X(I)/V6FU)
C          PINTFU = PINTFU + DM7FU*(P(I)*DM8FU+P(IPI)*DM9FU)
C          RHNTFU = RHNTFU + DM7FU*(RHO(I)*DM8FU+RHO(IPI)*DM9FU)
C          MRNTFU = MRNTFU + DM7FU*(MR(I)*DM8FU+MR(IPI)*DM9FU)
C          VINTFU = VINTFU + DM7FU*(V(I)*DM8FU+V(IPI)*DM9FU)
C /
C          I = I+1
C          IF(I.LE.NXPM1) GO TO 10
C
C          COMPUTE UPSTREAM NOZZLE ADMITTANCE.
C
C          CNOZA = GAMO*V(NXP)/P(NXP)

```

```

00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980

```

F S C S M SUBROUTINES

C

RETURN  
END

00001990  
00002000  
00002010



F S C S M SUBROUTINES

```

SUBROUTINE CHMCON(I)
ROUTINE TO CALCULATE PARAMETERS FOR SUBPROGRAM CHAMDY
PROGRAMMED BY K. W. FERTIG, ROCKETDYNE, MAY 1975

COMPLEX OMEGA, P, RHO, V, MR, T, CNOZA, VXO,
1 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
2 COX10, COX11, COX12, COX13, COX14, COX15, COX16,
3 CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
4 CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
COMPLEX DM1, DM3, DM4, DM5, DM6, DM7OX, DM8OX,
1 DM6FU, G1OX, G1FU, G2OX, G2FU, G3OX, G3FU, G4OX, G4FU,
2 RBSOX, RBSFU, PINTOX, PINTFU, RHNTOX, RHNTFU, MRNTOX,
3 MRNTFU, VINTOX, VINTFU, II
4 ,DM9OX,DM9FU

REAL M6OXI, MBFUI, MR8, MRGI, MGI, NU8OX, NUBFU, MWG

COMMON /COMCBM/ XKOX, XKFU, MBOXI, MBFUI, TAUBOX, TAUBFU, VBOX,
1 VBFU, GAMO, RGO, DELHDX, DELHFU, PC, CO,
2 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
3 COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
4 CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
5 CFU12, CFU13, CFU14, CFU15, CFU16, MWG, XIMPFU, XIMPOX,
6 CS, DCSDMR, DHDMR, DRGDMR, ADVGX, ADDOX, TDRAGO, DELVOX,
7 NUBOX, DTOXDM, ADVFU, ADDFU, TURAGF, DELVVFU, NUBFU, DTFUDM

COMMON /CONSTS/ MRB(100), TB(100), RHOB(100), VB(100),
1 DMRB(100), DRHOB(100), DVB(100), VAPBOX(100), VAPBFU(100),
2 SSV1(100), SSV2(100), SSV3(100), SSV4(100),
3 SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9OX(100),
4 SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),
5 SSV14(100), SSV15(100), SSV16(100),

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350

```

F S C S M SUBROUTINES

```

C I
6   RHGGI, VGI, MRGI, MGI
COMMON /COMCHM/ P(100), RHO(100), V(100), MR(100), T(100),
1   VXC, OMEGA, CNOZA, DELP
COMMON /COMARE/ NXP, X(100), XM(100), A(100), DA(100), DELX,
1   XG, XNDZ, AINJ
COMMON /ADARND/ G10X,G20X,G30X,G40X,G1FU,G2FU,G3FU,G4FU,PINTOX,
1   RHNTOX,MRNTOX,VINTOX,PINTFU,RHNTFU,MRNTOFU,VINTFU,RBSOX,
2   KSFU,DM1,DM3,DM4,DM5,DM6,DM7OX,UM8OX,DM9OX,DM7FU,
3   DM8FU,DM9FU,II,DM2,DM22,RHOINJ
10  DM1 = CEXP(II*OMEGA*(XM(II))/VBOX)
    DM2 = (XM(II)-X0)/(TAUBOX*VBOX)
    DM6 = CEXP(II*OMEGA*DELX/VBOX)
    DM3 = CEXP(II*OMEGA*X(II)/VBOX)*(1.+DM6)/2.
C
C   CALCULATE OXIDIZER VAPORIZATION PARAMETERS.
C
DM7OX = VBOX/(2.*OMEGA**2*DELX*TAUBOX)
DM8OX = 1. - II*OMEGA*DELX/VBOX - 1./DM6
DM9OX = 1. + II*OMEGA*DELX/VBOX - DM6
DM4 = DM7OX*DM8OX*DM6
DM5 = DM7OX*DM9OX
DM9UX = DM9OX/DM6
RBSOX = DELP*DM1*(COX1+COX9*DM2)
1   +RHUINJ*DM1*(COX3+COX11*DM2)+MR(1)*DM1*(COX5+COX13*DM2)
2   +VXO*DM1*(COX7+COX15*DM2)
1   RBSUX = RBSOX+DM3*(COX10*PINTOX+COX12*RHNTOX+COX14*MRNTOX
    +COX16*VINTOX)
RBSOX = RBSOX+P(II)*(COX2/2.+COX10*DM4) + RHO(II)*(COX4/2.
1   +COX12*(M4) + MR(II)*(COX6/2.+COX14*DM4)
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700

```

F S C S M SUBROUTINES

```

2      + V(I)*(COX8/2.+COX16*DM4)
      RBSOX = VAPBOX(I)*RBSOX
      G1OX = VAPBOX(I)*(COX2/2.+COX10*DM5)
      G2OX = VAPBOX(I)*(COX4/2.+COX12*DM5)
      G3OX = VAPBOX(I)*(COX6/2.+COX14*DM5)
      G4OX = VAPBOX(I)*(COX8/2.+COX16*DM5)

      DM1 = CEXP(II*OMEGA*(XM(I))/VBFU)
      DM2 = (XM(I)-XU)/(TAUBFU*VBFU)
      DM6 = CEXP(II*OMEGA*DELX/VBFU)
      DM3 = CEXP(II*OMEGA*X(I)/VBFU)*(1.+DM6)/2.

      CALCULATE FUEL VAPORIZATION PARAMETERS.

      DM7FU = VBFU/12.*OMEGA**2*DELX*TAUBFU)
      DM8FU = 1. - II*OMEGA*DELX/VBFU - 1./DM6
      DM9FU = 1. + II*OMEGA*DELX/VBFU - DM6
      DM4 = DM7FU*DM8FU*DM6
      DM5 = DM7FU*DM9FU
      DM9FU = DM9FU/DM6
      RBSFU = DELP*DM1*(CFU1+CFU9*DM2)
      1      +RHOINJ*DM1*(CFU3+CFU11*DM2)+MR(1)*DM1*(CFU5+CFU13*DM2)
      2      +VX0*DM1*(CFU7+CFU15*DM2)
      RBSFU = RBSFU+DM3*(CFU10*PINTFU+CFU12*RHNTFU+CFU14*MRNTFU
      1      +CFU16*VINTFU)
      RBSFU = RBSFU+P(I)*(CFU2/2.+CFU10*DM4) + RHO(I)*(CFU4/2.
      1      +CFU12*DM4) + MR(I)*(CFU6/2.+CFU14*DM4)
      2      + V(I)*(CFU8/2.+CFU16*DM4)
      RBSFU = VAPBFU(I)*RBSFU
      G1FU = VAPBFU(I)*(CFU2/2.+CFU10*DM5)
      G2FU = VAPBFU(I)*(CFU4/2.+CFU12*DM5)
      G3FU = VAPBFU(I)*(CFU6/2.+CFU14*DM5)
      G4FU = VAPBFU(I)*(CFU8/2.+CFU16*DM5)

```

```

00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00010000
00010010
00010020
00010030

```

F S C S M SUBROUTINES

RETURN  
END

00001040  
00001050

F S C S M SUBROUTINES

```

SUBROUTINE COGAEL(A,N)
DIMENSION A(62,126)
C
C GAUSSIAN ELIMINATION OF COMPLEX MATRIX.
C PROGRAMMED BY J. K. HUNTING, ROCKETDYNE, MAY 1975
C
NMI = N - 1
NPI = N + 1
ZZ = 10.0E-30
C
C J IS THE COLUMN BEING ELIMINATED BELOW THE DIAGONAL.
C
DO 27 J = 1,N
C
C FIND MAXIMUM AMPLITUDE IN COLUMN AND BELOW THE DIAGONAL.
C
JMAX = J
JJ = J + NPI
CMAX = A(J,J)**2 + A(J,JJ)**2
JPI = J + 1
IF (J - N) 25,5,6
25 DO 2 K = JPI,N
CCC = A(K,J)**2 + A(K,JJ)**2
IF (CCC - CMAX) 2,2,3
3 JMAX = K
CMAX = CCC
2 CONTINUE
C
C INTERCHANGE ROWS IF REQUIRED TO OBTAIN MAXIMUM PIVOTAL ELEMENT.
C
IF (J - JMAX) 4,5,6
6 WRITE (6,12) J,JMAX
12 FORMAT (6I1ERKOR IN COGAEL SUBROUTINE, J AND JMAX EQUAL, RESPECTI
COGAEL(A,N)

```

F S C S M SUBROUTINES

```

1 IVELY, 2I12)
  CALL EXIT
4 DC 7 K = J,NP1
  K1 = K + NP1
  HOLD = A(J,K)
  A(J,K) = A(JMAX,K)
  A(JMAX,K) = HOLD
  HOLD = A(J,KN)
  A(J,KN) = A(JMAX,KN)
  A(JMAX,KN) = HOLD
7 CONTINUE
C
C DIVIDE PIVOT ROW BY PIVOT ELEMENT. PIVOT ELEMENT BECOMES 1.0.
C
5 CONTINUE
  DU 8 K = JP1,NP1
  KN = K + NP1
  IF (ABS(A(J,J)) - ZZ) 14,14,13
14 IF (ABS(A(J,JJ)) - ZZ) 15,15,13
15 DUM1 = A(J,J)
  DUM2 = A(J,JJ)
  WRITE (6,16) DUM1,DUM2
16 FORMAT (6I16)
  ITS ARE,2E14.6)
  CALL EXIT
13 REALA = (A(J,K)*A(J,J) + A(J,KN)*A(J,JJ))/CMAX
  VIMAG = (A(J,KN)*A(J,J) - A(J,K)*A(J,JJ))/CMAX
  A(J,K) = REALA
  A(J,KN) = VIMAG
8 CONTINUE
C
C ELIMINATE ELEMENTS BELOW DIAGONAL. THE ELEMENTS ELIMINATED ARE NOT
C THEMSELVES CALCULATED.
C
00012320
00012330
00012340
00012350
00012360
00012370
00012380
00012390
00012400
00012410
00012420
00012430
00012440
00012450
00012460
00012470
00012480
00012490
00012500
00012510
00012520
00012530
00012540
00012550
00012560
00012570
00012580
00012590
00012600
00012610
00012620
00012630
00012640

```

F S C S M SUBROUTINES

```

C
  26 IF (J - N) 26,27,6
    JPI = J + 1
    DO 9 I = JPI,N
      DO 9 K = JPI,NPI
        KN = K + NPI
        REALA = A(I,K) - A(J,K)*A(I,J) + A(J,KN)*A(I,JJ)
        VIMAG = A(I,KN) - A(J,KN)*A(I,J) - A(I,JJ)*A(J,K)
        A(I,K) = REALA
        A(I,KN) = VIMAG
      9 CONTINUE
    27 RETURN
      END

```

00012650  
 00012660  
 00012670  
 00012680  
 00012690  
 00012700  
 00012710  
 00012720  
 00012730  
 00012740  
 00012750  
 00012760  
 00012770  
 00012780

F S C S M SUBROUTINES

```

SUBROUTINE COMBDY(IR,FREQ,GINJOX,GINJFU,IPRCOM,INPCOM)
SUBPROGRAM TO CALCULATE THE COMBUSTION COEFFICIENTS.
PROGRAMMED BY M. D. SCHUMAN, ROCKETUZYNE, MAY 1975

COMPLEX GINJOX, GINJFU, DUM, NOZAT, GINJOT, GINJFT,
1 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
2 COX10, COX11, COX12, COX13, COX14, COX15, COX16,
3 CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
4 CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16,
5 COXT(16), CFUT(16), RM, RU, RD, RFU

REAL MBOXI, MBFUI, MWG, NUBOX, NUBFU

COMMON /COMTAP/ NFREQI, FREQI(100), NOZAT(100), GINJOT(100),
1 GINJFT(100), ITAPN, ITAPC, ITAPH

COMMON /COMCBM/ XKOX, XKFU, MBOXI, MBFUI, TAUBOX, TAUBFU, VBOX,
1 VBFU, GAMD, RGO, DELHOX, DELHFU, PC, CO,
2 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
3 COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
4 CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
5 CFU12, CFU13, CFU14, CFU15, CFU16, MWG, XIMPFU, XIMPOX,
6 CS, DCSUMR, DHDMR, DRGDMR, ADVUX, ADDOX, TDRAGO, DELVOX,
7 NUBOX, UTOXDM, ADVFU, ADDFU, TDRAGF, DELVFU, NUBFU, DTFUDM

EQUIVALENCE (COXT(1),COX1), (CFUT(1),CFU1)

DATA PI/3.14159/

IF(IR.LE.0) GO TO 100

```

9  
C  
C  
C  
C  
C

C  
C  
C

d  
C  
C  
C

00000010  
00000020  
00000030  
00000040  
00000050  
00000060  
00000070  
00000080  
00000090  
00000100  
00000110  
00000120  
00000130  
00000140  
00000150  
00000160  
00000170  
00000180  
00000190  
00000200  
00000210  
00000220  
00000230  
00000240  
00000250  
00000260  
00000270  
00000280  
00000290  
00000300  
00000310  
00000320  
00000330



F S C S M SUBROUTINES

```

C
C
C
C
      THIS IS FIRST PASS FOR THIS CASE. READ INPUT DATA.

      READ(5,1) XKOX, TAUBOX, VBOX, DELHOX, TDRAGO, ADVOX,
1      ADDOX, DELVOX, NUBOX, DTOXDM, XIMPOX
1  FORMAT(6E12.8)
      WRITE(6,2) XKOX, TAUBOX, VBOX, DELHOX, TDRAGO,
1      ADVOX, ADDOX, DELVOX, NUBOX, DTOXDM, XIMPOX
2  FORMAT(/5X, 'XKOX =', 1PE11.4, 5X, 'TAUBOX =',
1      E11.4, 3X, 'VBOX =', E11.4, 5X, 'DELHOX =', E11.4, /, 27X,
2      'TDRAGO =', E11.4, 3X, 'ADVOX =', E11.4, /, 5X, 'ADDOX =', E11.4,
3      4X, 'DELVOX =', E11.4, 3X, 'NUBOX =', E11.4, 4X, 'DTOXDM =',
4      E11.4, /, 27X, 'XIMPOX =', E11.4)
      XKOX = XKOX*0.0254
      VBOX = VBOX*0.3048
      DELHOX = DELHOX*1054.35/0.453592
      XIMPOX = XIMPOX*0.0254

      READ(5,1) XKFU, TAUBFU, VBFU, DELHFU, TDRAGF, ADVFU,
1      ADDFU, DELVDFU, NUBDFU, DTFUDM, XIMPFU
      WRITE(6,3) XKFU, TAUBFU, VBFU, DELHFU, TDRAGF,
1      ADVFU, ADDFU, DELVDFU, NUBDFU, DTFUDM, XIMPFU
3  FORMAT(/5X, 'XKFU =', 1PE11.4, 5X, 'TAUBFU =',
1      E11.4, 3X, 'VBFU =', E11.4, 5X, 'DELHFU =', E11.4, /, 27X,
2      'TDRAGF =', E11.4, 3X, 'ADVFU =', E11.4, /, 5X, 'ADDFU =', E11.4,
3      4X, 'DELVDFU =', E11.4, 3X, 'NUBDFU =', E11.4, 4X, 'DTFUDM =',
4      E11.4, /, 27X, 'XIMPFU =', E11.4)
      XKFU = XKFU*0.0254
      VBFU = VBFU*0.3048
      DELHFU = DELHFU*1054.35/0.453592
      XIMPFU = XIMPFU*0.0254
      READ(5,1) MWG, CS, DRGDMR, UCSDMR, DHDMR

```

```

CC000340
CC000350
CC000360
CC000420
CC000430
CC000440
CC000450
CC000460
CC000470
CC000480
CC000490
CC000500
CC000510
CC000520
CC000530
CC000540
CC000550
CC000560
CC000620
CC000630
CC000640
CC000650
CC000660
CC000670
CC000680
CC000690
CC000700
CC000710
CC000720
CC000730
CC000740
CC000750
CC000790

```

F S C S M SUBROUTINES

```

WRITE(6,4) MWG, CS, DRGDMR, DCSDMR, DHDMR
4 FORMAT(/,5X,'MWG =',1PE11.4,6X,'CS =',E11.4,
1  /X,'DRGDMR =',E11.4,3X,'DCSDMR =',F11.4,/,27X,'DHDMR =',
2  E11.4)
CS = CS*C.3048
DRGDMR = DRGDMR*1.8*1.35582/0.453592
DCSDMR = DCSDMR*0.3048
DHDMR = DHDMR*1054.35/0.453592
RGO = 8314.34/MWG
REWIND ITAPC
C 100 CONTINUE
C
C IF INPCOM .LE. 1, COMPUTE COMBUSTION COEFFICIENTS FOR INPUT
C FREQUENCY(FREQ). IF INPCOM .GT. 1, COMPUTE COMBUSTION
C COEFFICIENTS FOR ENTIRE FREQUENCY TABLE AND SAVE ON ITAPC.
C
FRE = FREQ
I = 0
IF(INPCOM.LE.1) GO TO 200
FRE = FREQ(1)
WRITE(ITAPC) XKOX, TAUBOX, VBOX, DELHOX, TDRAGO, ADVOX,
1  ADDUX, DELVOX, NUBOX, DTOXDM, XIMPOX, XKFU, TAUBFU,
2  VBFU, DELHFU, TDRAGF, ADVFU, DELVFU, NUBFU,
3  DTFUDM, XIMPFU, MWG, CS, DRGDMR, DCSDMR, DHDMR, RGO
200 I = I+1
DUM = (0.0,1.0)*2.*PI*FRE*XKOX/VBOX
RM = (1.-DUM)*GINJOX
PJ = GINJUX
DUM = DUM*XIMPOX/XKOX
RD = (ADVFX-0.5*ADDOX*DUM)*GINJOX
RFR = 0.5
U1 = 2.*PI*FRE*TURAGO
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120

```

F S C S M SUBROUTINES

```

RFU = 0.5/DELVOX*(D1*D1-(0.0,1.0)*D1)/(1.+D1*D1)
COX1 = RM-RU-2.*RD+(NUBOX-2.)*0.5*RD/NUBOX
COX2 = 0.0
COX3 = 0.0
COX4 = RFR*(NUBOX-2.)/NUBOX
COX5 = 0.0
COX6 = -DTOXDM/TAUBOX
COX7 = 0.0
COX8 = RFU*(NUBOX-2.)/NUBOX
COX9 = 2.*KD-(NUBOX-2.)*0.5*RD/NUBOX+RU
COX10 = 0.0
COX11 = 0.0
COX12 = -COX4
COX13 = 0.0
COX14 = -COX6
COX15 = 0.0
COX16 = -COX8
DUM = (0.0,1.0)*2.*PI*FRE*XKFU/VBFU
RM = (1.-DUM)*GINJFU
RU = GINJFU
DUM = DUM*XIMPFU/XKFU
RD = (ADVFU-0.5*ADDFU*DUM)*GINJFU
RFR = 0.5
D1 = 2.*PI*FRL*TDGRAG
RFU = 0.5/DELVFU*(D1*D1-(0.0,1.0)*D1)/(1.+D1*D1)
CFU1 = KM-RU-2.*RD+(NUBFU-2.)*0.5*RD/NUBFU
CFU2 = 0.0
CFU3 = 0.0
CFU4 = RFR*(NUBFU-2.)/NUBFU
CFU5 = 0.0
CFU6 = -DTFUDM/TAUBFU
CFU7 = 0.0
CFU8 = KFU*(NUBFU-2.)/NUBFU

```

```

00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450

```

F S C S M SUBROUTINES

```

CFU9 = 2.*RD-(NUBFU-2.)*0.5*RD/NUBFU+RU
CFU10 = 0.0
CFU11 = 0.0
CFU12 = -CFU4
CFU13 = 0.0
CFU14 = -CFU6
CFU15 = 0.0
CFU16 = -CFU8
C
IF(INPCOM.LE.1) RETURN
C
WRITE(ITAPC) (COXT(J),CFUT(J),J=1,16)
IF(I.GE.NFREQ) GO TO 900
FRE = FREQ(I+1)
GO TO 200
900 END FILE ITAPC
C
RETURN
END
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640

```

F S C S M SUBROUTINES

```

C
C SUBROUTINE COMMAT(A,NRA,N)
C
C ROUTINE SOLVES: AX = B, WHERE A,X, AND B ARE COMPLEX.
C A(I,J) IS N BY N COMPLEX MATRIX
C B(I) IS N BY 1 COMPLEX VECTOR. IT IS STORED IN A(I,N+1) FOR
C I = 1,N
C ANSWER IS RETURNED IN A(I,N+1) I=1,N
C NRA IS THE NUMBER OF ROWS IN THE DIMENSION STATEMENT FOR
C MATRIX A IN THE CALLING PROGRAM.
C
C A,X, AND B ARE ALL COMPLEX*16
C
C PROGRAMMED BY K. W. FERTIG, ROCKETDYNE, MAY 1975
C
C COMPLEX*16 A(NRA,1),W
C
C NMI = N-1
C NPI = N+1
C
C MATRIX TRIANGULARIZATION IS PERFORMED THROUGH STATEMENT 20
C
C DO 20 IR = 1,NMI
C   IRL = IR+1
C   IF(CDABS(A(IR,IR)).LE.0.00) GO TO 100
C
C   DO 10 I = IRL,N
C     W = -A(I,IR)/A(IR,IR)
C     DO 10 J = IRL,NPI
C       10 A(I,J) = A(I,J) + W*A(IR,J)
C
C   20 CONTINUE
C
C
C

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330

```

F S C S M SUBROUTINES

```

C      BACKWARDS SUBSTITUTION IS PERFORMED THROUGH STATEMENT 50
C      C
C      A(N,NP1) = A(N,NP1)/A(N,N)
C      DO 50 IR = 1,NM1
C      I = N-IR
C      II = I+1
C      DO 40 J = II,N
C      40 A(I,NP1) = A(I,NP1) - A(I,J)*A(J,NP1)
C      50 A(I,NP1) = A(I,NP1)/A(I,I)
C      RETURN
C
C      ALGORITHM HAS FAILED DUE TO ZERO DIAGONAL ELEMENT
C
C      100 WRITE(6,9000) IR,((I,J,A(I,J),I=1,N),J=1,NP1)
C      9000 FORMAT(// , **** DIVIDE CHECK IN COMMENT **** //
C      1 , IR = ,I10// MATRIX A(I,J) = ,/
C      2 (5X,2I6,1PE15.6, , : ,1PE13.6))
C      CALL EXIT
C      RETURN
C      END
C
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550

```

F S C S M SUBROUTINES

```

C
C
C
C
SUBROUTINE FRESP(ICRT, IWRITE, IX)
FREQUENCY RESPONSE ROUTINE WITH TIME DELAY
PROGRAMMED BY J. K. HUNTING, ROCKETDYNE, MAY 1975

DIMENSION OM(101), OMSCPS(101), W(101), OMS(101), ID(126), CP(62, 126)
DIMENSION LXI(3), JI(3), JI(3), C(62, 126), TD(62, 63), D(126)
DIMENSION XMAG(62), NO(62), XREAL(62), XIMAG(62), GAIN(101), FAZE(101)
DIMENSION VARR(62), VARI(62), PHAN(5, 101), DEC8(5, 101)
DIMENSION ATD(126), IDF(18)
COMMON/F/NW, NCTD, NR, KEQ, TRIG, IRPS, OMI, OMFL, W, ID, C, TU, ATD,
IFREQ, GINJDX, GINJFU, PC, WO, WF
COMPLEX NOZAT(100), GINJDT(100), GINJFT(100), GINJDX, GINJFU
COMMON/CUMTAP/NFREQT, FREQT(100), NCZAT, GINJDT, GINJFT, ITAPN, ITAPC,
ITAPH
DELTA(Q001FL) = 1.0 - ABS(Q001FL)/AMAX1(ABS(Q001FL), .146936794E-38)
UNIT(Q002FL) = 0.5 + SIGN( 0.5, DELTA(Q002FL) ) + Q002FL*(1.0-DELTA(Q0000006130
102FL))
THETA(Q003FL, Q004FL) = (180.0*(1.0+UNIT(Q003FL))*(1.0-2.0*UNIT(Q004FL)+UNIT(Q004FL*Q0006150
1L)))+ SIGN(1.0, Q003FL*Q004FL)* ATAN((ABS(Q004FL))*(1.0-DELTA(Q0030006616C
2FL)))/(ABS(Q003FL)+DELTA(Q003FL)))*(180.0/3.141593)) * (1.0-DELTA(Q0006170
3(Q0003FL))+ 180.0*DELTA(Q0003FL)*(1.0-DELTA(Q004FL)))*(0.5 +UNIT(-Q00400006618G
4FL))
C
10 DO 2001 KK = 1, 62
DO 2001 LL = 1, 126
POC1 CP(KK, LL) = 0.0
IF (TRIG.NE.1.) GO TO 2006
DO 2002 KK = 1, 62
DO 2002 LL = 1, 126
2002 C(KK, LL) = 0.0
DO 2003 KK = 1, 62
DO 2003 LL = 1, 63
C
00006000
00006001
00006010
00006011
00006012
00006020
00006030
00006040
00006050
00006060
00006070
00006080
00006090
00006100
00006110
00006120
00006130
00006140
00006150
00006160
00006170
00006180
00006190
00006200
00006210
00006220
00006230
00006240
00006250
00006260
00006270
00006280
00006290

```

F S C S M SUBROUTINES

```

2003 TD(KK,LL)=0.0
2006 DO 2004 KK=1,5
DO 2004 LL=1,101
PHAN(KK,LL)=0.0
2004 DECB(KK,LL)=0.0
IF(TRIG.NE.1.) GO TO 444
IF(TRIG.EQ.1.)READ(5,120)NW,NCID,NR,ICRT,KEQ,TRIG,IRPS,OMI,OMFL
120 FORMAT(I12,I3,I9,I3,I9,F2.0,I10,2F12.0)
IF(NW)1216,1215,1216
1216 IF(NW-100)1217,1217,1218
1218 WRITE (6,122)
122 FORMAT(46H-NUMBER OF SHUFFLED IN FREQUENCIES EXCEEDS 100)
GO TO 9
1217 IF(TRIG.EQ.1.)READ (5,121)(W(I),I=1,NW)
121 FORMAT(6F12.0)
1215 IF(NR-126)1219,1219,1220
1220 WRITE (6,123)
123 FORMAT(36H-NUMBER OF COLUMNS EXCEEDS 126)
GO TO 9
1219 IF(TRIG.EQ.1.)READ (5,51)(ID(I),I=1,NR)
51 FORMAT(6I12)
IF(KEQ-62)1221,1221,1222
1222 WRITE (6,124)
124 FORMAT(34H-NUMBER OF EQUATIONS EXCEEDS 62)
GO TO 9
1221 IF(OMI)1671,3,1667
3 WRITE (6,115)
GO TO 9
1667 OM(I)=OMI
J=2
RATIO=(OMFL/OMI)**(1./AMAX1(1.,FLOAT(NFREQT-2)))
1668 OM(J)=OM(J-1)*RATIO
IF(OM(J).GE..9999*OMFL.OR..J.EQ.100) GO TO 1670

```

```

00006300
00006310
00006320
00006330
00006340
00006350
00006360
00006370
00006380
00006390
00006400
00006410
00006420
00006430
00006440
00006450
00006460
00006470
00006480
00006490
00006500
00006510
00006520
00006530
00006540
00006550
00006560
00006570
00006580
00006590
00006600
00006610
00006620

```



F S C S M SUBROUTINES

```

1669 J=J+1
      GO TO 1668
1670 OM(J)=OMFL
      NOM=J
      GO TO 17
1671 OMI=ABS(OMI)
1675 J=1
      N=0
      M=-1
      T=-1.0
      R=0.1
115 FORMAT(26H INITIAL FREQUENCY IS ZERO)
      4 IF(OMI-K)5,8,8
      5 M=M-1
      R=R/10.0
      GO TO 4
      8 K=1
      NEW CARD---INITIALIZATION OF L
      L=0
12 OM(J)=UNIT(T)*((2.0**L)*3.0*10.0**M)/2.0)+UNIT(-T)*(2.0**L)*10.0*
1*M
      IF(T)24,24,23
23 L=L+1
24 T=-T
      IF(L-3)26,27,25
27 IF(T)26,26,25
25 L=0
      T=-1.0
      M=M+1
26 GO TO(11,16),K
11 IF(OM(1)-OMI)12,15,13
13 OM(2)=OM(1)
      OM(1)=OMI

```

F S C S M SUBROUTINES

```

J=3
K=2
GO TO 12
15 OM(1)=OMI
K=2
16 J=J+1
IF(OM(J-1)-OMFL)12,18,13
18 OM(J-1)=OMFL
NUM=J-1
17 J=1
IF(NW)28,28,29
28 OMS(J)=OM(J)
IF(OMS(J)-OMFL)30,32,32
30 J=J+1
IF(J-100)28,28,45
29 N=0
I=1
35 K=J+N
IF(OM(J)-W(I))20,34,34
20 IF(J-NOM)33,33,34
33 OMS(K)=OM(J)
J=J+1
GO TO 35
34 OMS(K)=W(I)
N=N+1
IF(I-NW)36,37,37
37 IF(J-NOM)31,31,380
31 W(I)=OMFL
GO TO 38
36 I=I+1
IF(J-NOM)38,38,35
38 IF(OMS(K)-OMFL)35,32,32
380 OMFL=OMS(K)

```

```

00006960
00006970
00006980
00006990
00007000
00007010
00007020
00007030
00007040
00007050
00007060
00007070
00007080
00007090
00007100
00007110
00007120
00007130
00007140
00007150
00007160
00007170
00007180
00007190
00007200
00007210
00007220
00007230
00007240
00007250
00007260
00007270
00007280

```

F S C S M SUBROUTINES

```

444 IF(K-100)32,32,45
      IF(NFREQ.GT.100) GO TO 45
      OMS(1)=.001
      NP1=NFREQ+1
      DO 445 I=2,NP1
445  OMS(I)=FREQ(I-1)
      IF(NW.EQ.1) OMS(1)=FREQ
      GO TO 32
45  WRITE (6,117)
117  FORMAT(36H1THE NUMBER OF FREQUENCIES EXCEEDS 100)
      GO TO 9
32  J=1
431  IF(IRPS)432,432,43
432  OMFLC=OMFL
      UMFL=2.0*3.1415927*OMFL
43  IF(IRPS)39,39,40
39  OMSCPS(J)=OMS(J)
      OMS(J)=2.0*3.1415927*OMSCPS(J)
      GO TO 41
40  OMSCPS(J)=OMS(J)/(2.0*3.1415927)
41  IF(OMS(J)-OMFL)42,44,44
42  J=J+1
      GO TO 43
44  IF(TRIG.EQ.1)READ (5,50)(LXI(I),I1(I),J1(I),C7(I),I=1,3)
50  FORMAT(3(I2,I4,I6,F12.0))
      IF(TRIG.NE.1.) GO TO 60
      DO47 I=1,3
      IF(LXI(I))300,301,301
301  IF(I1(I))47,47,48
48  I9=I1(I)
      J9=J1(I)
      C(I9,J9)=C7(I)
      IF(LXI(I))47,47,60

```

```

00007290
00007300
00007310
00007320
00007330
00007340
00007350
00007360
00007370
00007380
00007390
00007400
00007410
00007420
00007430
00007440
00007450
00007460
00007470
00007480
00007490
00007500
00007510
00007520
00007530
00007540
00007550
00007560
00007570
00007580
00007590
00007600
00007610

```

F S C S M SUBROUTINES

```

300 IF(I1(I))47,47,302
302 ITD=I1(I)
    JTD=J1(I)
    TD(ITD,JTD)=C7(I)
    IF(LXI(I)+1)47,60,60
47 CONTINUE
49 GO TO 44
60 IW=1
    KEJ=KEQ+1
    IF(TRIG.EQ.1.)READ (5,113)
113 FORMAT(72H1
1
    IF(IWRITE.GT.1)WRITE (6,113)
    IF(IWRITE.GT.1)WRITE (6,59)(ID(I),I=1,NR)
59 FORMAT(16H I.D. VECTOR /(1H0,24I4))
    DO 111 I73=1,KEQ
111 IF(IWRITE.GT.1)WRITE (6,112)I73,(C(I73,J),J=1,NR)
112 FORMAT(13H0 EQUATION ,I3 / (4H0 ,1P7E14.6))
310 FORMAT(27H1MATRIX OF TIME DELAY TERMS)
    DO311 I=1,KEQ
311 IF(IWRITE.GT.1)WRITE (6,112)I,(TD(I,J),J=1,KEJ )
    IF(NCTD) 75,75,309
309 IF(TRIG.EQ.1.)READ (5,121)(ATD(K),K=1,NCTD)
75 DO 662 J=1,KEQ
    IJ=J+KEQ
    LS=0
    LK=0
    LF=0
    I=0
76 I=I+1
    IF(LK)7,62,610
62 IF(KEQ-LF)64,64,65
00007620
00007630
00007640
00007650
00007660
00007670
00007680
00007690
00007700
00007710
00007720
00007730
00007740
00007750
00007760
00007770
00007780
00007790
00007800
00007810
00007820
00007830
00007840
00007850
00007860
00007870
00007880
00007890
00007900
00007910
00007920
00007930
00007940

```

F S C S M SUBROUTINES

```

65 LF=LF+1
IM=KEQ+LF
67 IR=I+LS
68 IF(ID(IR))69,68,69
68 LS=LS+1
69 GO TO 67
69 II=IR-1
LK=LS
CREAL=0.0
CIMAG=0.0
R=1.0
S=0.0
LS=LS+1
55 CREAL=CREAL + R*C(J,IR)*(OMS(IW)**S)
LS=LS-1
IF(LS)7,57,56
56 CIMAG=CIMAG+R*C(J,II)*(DMS(IW)**(S+1.0))
S=S+2.0
R=(-1.0)*R
IR=IR-2
II=II-2
LS=LS-1
IF(LS)7,57,55
57 CP(J,LF)=CREAL
CP(J,IM)=CIMAG
GO TO 61
64 IR=I+LS
71 IF(ID(IR))70,71,70
71 LS=LS+1
GO TO 64
70 II=IR-1
LK=LS
DREAL=0.0

```

```

00007950
00007960
00007970
00007980
00007990
00008000
00008010
00008020
00008030
00008040
00008050
00008060
00008070
00008080
00008090
00008100
00008110
00008120
00008130
00008140
00008150
00008160
00008170
00008180
00008190
00008200
00008210
00008220
00008230
00008240
00008250
00008260
00008270

```

F S C S M SUBROUTINES

```

DIMAG=0.0
R=1.0
S=0.0
LS=LS+1
772 DREAL=DREAL+R*C(J,IR)*(OMS(IW)**S)
LS=LS-1
I=I+1
IF(LS)7,63,773
773 DIMAG=DIMAG+R*C(J,II)*(OMS(IW)**(S+1.0))
S=S+2.0
R=(-1.0)*R
IR=IR-2
II=II-2
LS=LS-1
IF(LS)7,63,772
63 D(IJ)=DREAL
D(IJ)=DIMAG
GO TO 61
7 WRITE (6,116)
116 FORMAT(24H LS HAS A NEGATIVE VALUE)
GO TO 9
610 LK=LK-1
61 CONTINUE
320 IF(I.LT.NR) GO TO 76
662 CONTINUE
L=1
DO 313 I=1,KEQ
DO 313 J=1,KEQ
CTD=1.0
KTD=KEQ+J
IF(TD(I,J))312,313,316
316 CTD=ATD(L)
L=L+1

```

```

00008280
00008290
00008300
00008310
00008320
00008330
00008340
00008350
00008360
00008370
00008380
00008390
00008400
00008410
00008420
00008430
00008440
00008450
00008460
00008470
00008480
00008490
00008500
00008510
00008520
00008530
00008540
00008550
00008560
00008570
00008580
00008590
00008600

```

F S C S M SUBROUTINES

```

312 ANGLE=ABS(TD(I,J)*OMS(IW))-6.283185*IFIX(ABS(TD(I,J)*OMS(IW)/6.28300008610
    1185))
    CP(I,J)=CP(I,J)+CTD*COS(ANGLE)
    CP(I,KTD)=CP(I,KTD)-CTD*SIN(ANGLE)
313 CONTINUE
    K=KEQ+1
    DO 73 I=1,KEQ
    L=KEQ+I
    IF(TD(I,K))72,73,72
72 ANGLE=ABS(TD(I,K)*OMS(IW))-6.28318*IFIX(ABS(TD(I,K)*OMS(IW)/6.283100008700
    185))
    D(I)=D(I)+COS(ANGLE)
    D(L)=D(L)+SIN(ANGLE)*SIGN(1.0,TD(I,K))
73 CONTINUE
    NPH=2*KEQ
    IF(IW-1)303,303,305
303 IF(IWRITE.GT.1)WRITE(6,323)
    DO 304 I=1,KEQ
    IJ=I+KEQ
304 IF(IWRITE.GT.1)WRITE(6,321)I,D(I),D(IJ),(CP(I,J),J=1,NPH)
305 CONTINUE
C
C ADD U-VECTUK TO CP-MATRIX. SHIFT COLUMNS OF IMAG CP BY 1 TO RIGHT.
C KEAL D INTO COL. KEQ+1. IMAG D INTO COL. 2KEQ+2.
C
    DO 202 I = 1,KEQ
    IN = 2*KEQ + 1 - I
    INP1 = IN + 1
    DO 202 J = 1,KEQ
    CP(J,INP1) = CP(J,IN)
202 CONTINUE
    IN = KEQ + 1
    INN = 2*KEQ + 2
    
```

F S C S M SUBROUTINES

```

DO 203 J = 1,KEQ
CP(J,IN) = D(J)
JN = J + KEQ
CP(J,INN) = D(JN)
203 CONTINUE
C
C CALL COGAEI TO TRIANGULARIZE THE COMPLEX MATRIX BY GAUSS. ELIMIN.
C
C CALL COGAEI(CP,KEQ)
C
C BACK SUBSTITUTE TO DETERMINE REAL AND IMAGINARY PART OF VARIABLES,
C THEN DETERMINE MAGNITUDE AND PHASE.
C
NPI = KEQ + 1
NNPI = 2*NNPI
XREAL(KEQ) = CP(KEQ,NNPI)
XIMAG(KEQ) = CP(KEQ,NNPI)
XMAQ(KEQ) = SQRT(XREAL(KEQ)**2 + XIMAG(KEQ)**2)
FAZE(KEQ) = THETA(XREAL(KEQ),XIMAG(KEQ))
DO 205 J = 2,KEQ
JJ = KEQ - J + 1
JJPI = JJ + 1
SUMR = 0.0
SUMI = 0.0
DO 206 K = JJPI,KEQ
KN = K + KEQ + 1
SUMR = SUMR + CP(JJ,K)*XREAL(K) - CP(JJ,KN)*XIMAG(K)
SUMI = SUMI + CP(JJ,KN)*XREAL(K) + CP(JJ,K)*XIMAG(K)
206 CONTINUE
XREAL(JJ) = CP(JJ,NPI) - SUMR
XIMAG(JJ) = CP(JJ,NNPI) - SUMI
XMAQ(JJ) = SQRT(XREAL(JJ)**2 + XIMAG(JJ)**2)
205 FAZE(JJ) = THETA(XREAL(JJ),XIMAG(JJ))

```



F S C S M SUBROUTINES

```

IF (IW - 1) 82,82,83
82 CONTINUE
IF(IWRITE.GT.1)WRITE (6,52)OMS(1),(I,XMAG(I),FAZE(I),I=1,KEU)
52 FORMAT(67H1 VARIABLE MAGNITUDES USING A FRECC009300
    QUENCY OF ,1PE14.6,5H RPS./ 1H0 ,84H VARIAB00009310
    2LE NUMBER MAGNITUDE PHASE /(1H0 ,00009320
    3I27,1P2E30.6))
770 IF(IWRITE.GT.1)WRITE (6,1001)
1001 FORMAT(70 H1 THESE ARE INTERMEDIATE RESULTS PRODUCED AFTER EACH MAG00009350
    1TRIX INVERSION./53H0INPUT FREQUENCY IN RADIAN5/SECOND AND CYCLES/S00009360
    2SECOND./59H0 VARIABLE,GAIN(D8),PHASE(DEGREES) VARIABLE,ECC0009380
    3TC. )
323 FORMAT(37H1INITIAL VALUES OF COEFFICIENT MATRIX)
321 FORMAT(/13,5X,1P2E20.6/(4H0 ,1P7E14.6))
771 IV=1
    J=3
    NO(1)=17
    NO(2)=55
    DO 78 I=1,NR
    IF(ID(I))79,80,61
79 IV=IV+1
80 GO TO 78
81 IF(IV.NE.17.AND.IV.NE.55) NO(J)=IV
    IF(IV.NE.17.AND.IV.NE.55) J=J+1
    IV=IV+1
78 CONTINUE
    NOPRT=MINO(5,J-1)
83 DO 84 K=1,NOPRT
    JX=NO(K)
    VARR(K) = XREAL(JX)
    VARI(K) = XIMAG(JX)
84 CONTINUE
86 DO 87 I=1,NOPRT

```

F S C S M SUBROUTINES

```

90 PHAN(I,IW)=THETA(VARR(I),VARI(I))
196 DECB(I,IW)=20.0*ALOG10(AMAX1(1.0E-30,SQRT(VARR(I)**2+VARI(I)**2))
87 CONTINUE
    IF(TRIG)960,960,961
960 IF(IWRITE.GT.1)WRITE (6,1000)OMS(IW),OMSCPS(IW),(NO(NOM),DECB(NOM
1,IW),PHAN(NOM,IW),NOM=1,NOPRT)
1000 FORMAT(/F20.4,F20.5/(I4,F8.2,F7.1,I4,F8.2,F7.1,I4,F8.2,F7.1,I4,F
18.2,F7.1,I4,F8.2,F7.1)
961 IF(OMS(IW).GE.OMFL.OR.IW.EQ.101) GO TO 95
85 IW=IW+1
    GO TO 75
95 CONTINUE
53 FORMAT(21H1 FREQUENCY ,5(13H VARIABLE ,I2,3H )/21H0
1 RPS CPS ,5(18H DECIBELS PHASE )/)
54 FORMAT(1P2E10.2,5(2H /, OP2F8.2)
    NZ=5
    NY=1
    NX=NOPRT
97 IF(NX)100,100,98
98 IF(IWRITE.GT.1)WRITE (6,53)(NO(I),I=NY,NZ)
    DO 99 J=1,IW
    IF(IWRITE.GT.1)WRITE (6,54)(OMS(J),OMSCPS(J),(DECB(K,J),PHAN(K,J)
1,K=NY,NZ))
99 CONTINUE
    NY=NY+5
    NZ=NZ+5
    NX=NX-5
    GO TO 97
100 CONTINUE
    DO 400 J=1,NOPRT
    DO 102 K=1,IW
    GAIN(K)=DECB(J,K)
    FAZE(K)=PHAN(J,K)

```

F S C S M SUBROUTINES

```

IF(K.GT.101.OR.IW.LE.2.OR.K.EQ.1) GO TO 102
IF(J.EQ.1.AND.IX.LE.1)GINJOT(K-1)=CMPLX(PC/WO*(10.**GAIN(K)/20.))00009950
&,(FAZE(K)))00009960
IF(J.EQ.2.AND.IX.NE.1)GINJFT(K-1)=CMPLX(PC/WF*(10.**GAIN(K)/20.))00009970
&,(FAZE(K)))00009980
102 CONTINUE00009990
IF(J.EQ.1.AND.IX.LE.1)GINJDX=CMPLX(PC/WO*(10.**GAIN(1)/20.)),(FAZ00010000
&E(1)))00010001
IF(J.EQ.2.AND.IX.NE.1)GINJFU=CMPLX(PC/WF*(10.**GAIN(1)/20.)),(FAZ00010010
&E(1)))00010011
IF(IW.LE.2.OR.ICRT.LT.1) GO TO 40000010020
IF(IRPS)104,104,10300010030
103 L=100010040
CALL TDPL0T(OMS,GAIN,IW,OMI,OMFL,L)00010050
CALL PRINTV(-21,21H GAIN IN DECIBELS , 400, 470)00010060
CALL PRINTV(-21,21H FREQUENCY IN RADIAN,400,005)00010070
L=200010080
CALL TDPL0T(OMS,FAZE,IW,OMI,OMFL,L)00010090
CALL PRINTV(-21,21H PHASE IN DEGREE,400,995)00010100
GO TO 10500010110
104 L=100010120
CALL TDPL0T(OMSCPS,GAIN,IW,OMI,OMFLC,L)00010130
CALL PRINTV(-21,21H GAIN IN DECIBELS , 400, 470)00010140
CALL PRINTV(-21,21H FREQUENCY IN CPS ,400,005)00010150
L=200010160
CALL TDPL0T(OMSCPS,FAZE,IW,OMI,OMFLC,L)00010170
CALL PRINTV(-21,21H PHASE IN DEGREE,400,995)00010180
105 CALL PRINTV(-21,20H VARIABLE NUMBER ,400,1010)00010190
PND=ND(J)00010200
CALL LABLV (PND,560,1010,4,2,3)00010210
400 CONTINUE00010220
999 RETURN00010230
9 CALL EXIT00010240

```

F S C S M SUBROUTINES

STOP  
END

00010250  
00010260

F S C S M SUBROUTINES

```

SUBROUTINE HEAD
ROUTINE TO PRINT HEADING
PROGRAMMED BY M. D. SCHUMAN, ROCKETDYNE, MAY 1975
WRITE(6,10)
10 FORMAT(1H1,////,43X,'ANALYTICAL DESCRIPTION',//,
1 44X,'FEED SYSTEM COUPLED',//,47X,'STABILITY MODEL',/////,
2 47X,'COMPUTER MODEL',//,25X,'PROGRAM NAME: FSCSM, FIV VER',/////,
3 'SION, MAY 1975',//,25X,'DEVELOPED BY: M. D. SCHUMAN,',
4 ' J. K. HUNTING, AND K. W. FERTIG',//,42X,'ADVANCED PROGRAMS,',
5 ' ROCKEIDYNE',//,42X,'DIVISION OF ROCKWELL INTERNATIONAL',//,
6 42X,'CANAOGA PARK, CALIF 91304',//,
7 25X,'SPONSERED BY: NASA/LYNDON B. JOHNSON SPACE CENTER',//,
8 42X,'HOUSTON, TEXAS 77058',//,42X,'UNDER CONTRACT NAS9-14315')
C
RETURN
END
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000150
00000140
00000150
00000160
00000170
00000180

```

F S C S M SUBROUTINES

```
BLOCK DATA
COMMON/HY/ICRT,IRFLAG,ITERM,ITYPE,IMWRITE,IX,A,CH,KF,KO,L,R,RHOL,V,00013010
&VF,VO,VCLF,VOLO,ZF,ZO,OSAVE(188),FSAVE(188)
REAL A(30),CH(30),KF,KO,L(30),R(30),RHOL(30),V(30)
DATA ICRT/0/,IRFLAG/0/,ITERM/0/,ITYPE/1/,IMWRITE/0/
END
00013040
00013050
```

F S C S M SUBROUTINES

```

C  SUBROUTINE HYDRDY(IR,INPHYD,FKE,GINO,GINF,PCIN,WGIN,WFIN)
C  SUBROUTINE HYDROY(IR,INPHYD,FREQ,GINJOX,GINJFU,PC,WO,WF)
C
C  SUBPROGRAM TO CALCULATE FEED SYSTEM RESPONSE
C  PRGRAMMED BY J. K. HUNTING, ROCKETDYNE, MAY 1975
C  ARGUMENT LIST VARIABLES:
C  IR - DATA FLAG: =0-NO DATA READ; =1-READ NEW DATA CASE;
C  =2-MODIFY EXISTING DATA CASE
C  INPHYD - MODE FLAG: =1-CALCULATE INJ. FLOW GAIN & PHASE AT 1 FREQ.
C  =2-CALC. & SAVE INJ. FLOW GAIN & PHASE FOR FREQ. RANGE
C  FREQ - INPUT FREQUENCY FOR CASE WITH INPHYD=1
C  GINJOX - OUTPUT COMPLEX NUMBER WITH AMPLITUDE AND PHASE ANGLE OF
C  OXID. INJECTOR FLOW OSCILLATIONS AT FREQUENCY, FREQ
C  GINJFU - OUTPUT COMPLEX NUMBER WITH AMPLITUDE AND PHASE ANGLE OF
C  FUEL INJECTOR FLOW OSCILLATIONS AT FREQUENCY, FREQ
C  PC - STEADY STATE OPERATING CHAMBER PRESSURE (LB/IN**2)
C  WU - STEADY STATE OXIDIZER INJECTOR FLOW (LB/SEC)
C  WF - STEADY STATE FUEL INJECTOR FLOW (LB/SEC)
C
C  LABELED COMMON BLOCK /COMTAP/ VARIABLES:
C  FREQT - ARRAY OF FREQUENCIES (100 MAX.) FOR INPHYD=2 CASE
C  NFREQT - NUMBER OF FREQUENCIES IN ARRAY, FREQT
C  GINJOT - TABLE OF COMPLEX NUMBERS WITH GAINS AND PHASE ANGLES OF
C  OXID. INJECTOR FLOW OSCILLATIONS AT FREQT FREQUENCIES
C  GINJFT - TABLE OF COMPLEX NUMBERS WITH GAINS AND PHASE ANGLES OF
C  FUEL INJECTOR FLOW OSCILLATIONS AT FREQT FREQUENCIES
C
C  NAMELIST /HYD/ VARIABLES:
C  ICRT - PLOT FLAG: =0-NO PLOTS; =1-PLOT GAIN&PHASE VS FREQUENCY
C  IWRITE - PRINT FLAG: =-2-NO PRINT; =-1-PRINT OUTPUT ONLY;
C  =0-PRINT INPUT ONLY; =1-PRINT INPUT & OUTPUT;
C  =2-PRINT INPUT, OUTPUT & INTERMEDIATE CALCULATIONS
C  IRFLAG - READ FLAG: =0-CARD INPUT; =1-TERMINAL INPUT FROM ITEM

```

```

00000005
00000010
00000020
00000021
00000022
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300

```





F S C S M SUBROUTINES

```

C | NOTE: MAXIMUM OF 100 VALUES IN FREQT ARRAY
C | MAXIMUM OF 30 VALUES IN A, CW, L, R, RHOL & V ARRAYS
C
COMMON/HY/ICRT,IRFLAG,I TERM,ITYPE,ITWRITE,IX,A,CW,KF,KO,L,R,RHOL,V,CCCC1005
&VF,VO,VOLF,VOLO,ZF,ZO,OSAVE(188),FSAVE(188)
REAL X(188)
EQUIVALENCE (X(1),A(1))
COMMON/F/NW,NCTD,NR,KEQ,TRIG,IRPS,OMI,OMFL,W,IH,C,TD,ATD,
1FREQ,GINJOX,GINJFU,PC,W0,WF
COMPLEX NOZAT(100),GINJDT(100),GINJFT(100),GINJOX,GINJFU,
&GIND,GINF
COMMON/COMTAP/NFREQT,FREQT(100),NOZAT,GINJDT,GINJFT,ITAPN,ITAPC,
1ITAPH
REAL L(30),R(30),A(30),V(30),C(62,126),TD(62,63),ATD(126),W(101)00001070
6,CW(30),RHOL(30),KO,KF
INTEGER*4 IH(126)
NAMELIST/HYD/L,R,A,V,OMI,OMFL,FREQ,NFREQT,IWRITE,ICRT,ZF,RF,VOLF,
&VF,ZO,RO,VOLO,VO,FREQT,IH,ITERM,IRFLAG,KO,KF,CW,RHOL,ID,ITYPE,IX
IX=0
IF(INPHYD.EQ.2) REWIND ITAPH
G=386.4
FREQ=FRE
PC=PCIN
W0=WOIN
WF=WFIN
5 DO 10 I=1,62
DO 10 J=1,126
10 C(I,J)=0.
DO 20 I=1,62
DO 20 J=1,63
20 TU(I,J)=0.
00000640
00000650
00000660
00000670
00000680
00001007
00001008
00001010
00001020
00001020
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001115
00001111
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220

```

F S C S M SUBROUTINES

```

30 DO 30 I=1,126
    IH(1)=-1
    IH(15)=0
    IH(18)=0
    IH(19)=1
    IH(55)=0
    IH(58)=0
    IH(59)=1
    IH(62)=0
    NW=0
    NCTD=113
    NR=63
    KEQ=57
    TRIG=0.
    IRPS=0
    OMI=FREQT(1)
    OMFL=FREQT(NFKQNT)
35 IF(IR.LE.0) GO TO 100
    IF(IR.EQ.2) GO TO 45
    DO 40 I=1,30
        R(I)=1000000.
        V(I)=40000.
        A(I)=1.
        CW(I)=0.
        RHOL(I)=.04
40 L(I)=.001
        ZF=.00004
        ZU=.00004
        RF=.1
        RU=.1
        VF=40000.
        VU=40000.
        KF=0.

```

```

00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560

```

F S C S M SUBROUTINES

```

KD=0.
VOLF=.01
VOLO=.01
45 IF(IRFLAG.EQ.0) READ(5,HYD,END=60)
   IF(ITERM.GT.0) IRFLAG=1
50 FORMAT(' INPUT NAMELIST &HYD DATA',* VARIABLES ARE: L,R,A,V,CW,RHOL,
&L,NFREQ,FREQT,IWRITE,ITERM',* ICRT,ZF,RF,VO,KOLF,VF,KF,ZO,RO,VOL,
&,VO,IO,ITYPE,IRFLAG')
60 IF(ITERM.GT.0) READ(ITERM,HYD,END=999)
66 IF(ITERM.GT.0) WRITE(ITERM,66)
   FORMAT(' END OF &HYD INPUT')
   IF(IWRITE.GE.0) WRITE(6,HYD)
100 IF(ITYPE.EQ.2) IX=MIN0(IX+1,2)
C GENERATE COEFFICIENT MATRIX FOR FREQUENCY RESPONSE ROUTINE
  DO I=1,30
101 V(I)=SQRT(1./(1.+V(I))/V(I)+RHOL(I)*CW(I)/G)
    C(1,1)=1.
    C(1,2)=R(1)+V(1)/G/A(1)
    C(2,1)=1.
    C(2,4)=-V(3)/G/A(3)
    C(3,3)=1.
    C(3,5)=R(3)+V(3)/G/A(3)
    C(4,1)=1.
    C(4,2)=-R(2)+V(2)/G/A(2)
    C(4,4)=R(2)+V(2)/G/A(2)
    C(5,3)=1.
    C(5,6)=-V(5)/G/A(5)
    C(6,3)=1.
    C(6,5)=-V(4)/G/A(4)
    C(6,6)=V(4)/G/A(4)
    C(7,21)=1.
    C(7,20)=R(4)+V(4)/G/A(4)

```

```

00001570
00001580
00001590
00001600
00001610
00001620
00000000
00000000
00000000
00001640
00001645
00001650
00001660
00001670
00001680
00001685
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870

```

F S C S M SUBROUTINES

C(8,7)=1.	00001880
C(8,8)=R(5)+V(5)/G/A(5)	00001890
C(9,7)=1.	00001900
C(9,8)=-V(6)/G/A(6)	00001910
C(10,9)=1.	00001920
C(10,10)=-R(7)+V(7)/G/A(7)	00001930
C(10,11)=R(7)+V(7)/G/A(7)	00001940
C(11,9)=1.	00001950
C(11,11)=-V(8)/G/A(8)	00001960
C(12,12)=1.	00001970
C(12,13)=R(8)+V(8)/G/A(8)	00001980
C(13,12)=1.	00001990
C(13,13)=-V(9)/G/A(9)	00002000
C(14,14)=1.	00002010
C(14,27)=R(9)+V(9)/G/A(9)	00002020
C(15,14)=1.	00002030
C(15,27)=-V(10)/G/A(10)	00002040
C(16,16)=1.	00002050
C(16,17)=R(10)+V(10)/G/A(10)	00002060
C(17,16)=1.	00002070
C(17,18)=-Z0	00002080
C(17,19)=-RC	00002090
C(18,15)=1.+V0*V0*K0/G/V0LO	00002100
C(18,17)=-V0*V0/G/V0LO	00002110
C(18,19)=V0*V0/G/V0LO	00002120
C(18,62)=V0*V0*K0/G/V0LO	00002130
C(19,21)=1.	00002140
C(19,22)=V(14)/G/A(14)	00002150
C(20,29)=1.	00002160
C(20,28)=-K(14)+V(14)/G/A(14)	00002170
C(21,21)=1.	00002180
C(21,20)=-V(11)/G/A(11)	00002190
C(21,22)=-V(11)/G/A(11)	00002200

F S C S M SUBROUTINES

```

C(22,24)=1.
C(22,23)=R(11)+V(11)/G/A(11)
C(23,24)=1.
C(23,23)=-R(13)+V(13)/G/A(13)
C(23,25)=R(13)+V(13)/G/A(13)
C(24,24)=1.
C(24,25)=-V(12)/G/A(12)
C(25,21)=1.
C(25,30)=R(15)+V(15)/G/A(15)
C(26,31)=1.
C(26,30)=-V(16)/G/A(16)
C(27,29)=1.
C(27,28)=R(16)+V(16)/G/A(16)
C(27,32)=R(16)+V(16)/G/A(16)
C(28,43)=1.
C(28,44)=R(22)+V(22)/G/A(22)
C(29,43)=1.
C(29,45)=-V(24)/G/A(24)
C(30,43)=1.
C(30,44)=-R(23)+V(23)/G/A(23)
C(30,45)=R(23)+V(23)/G/A(23)
C(31,29)=1.
C(31,32)=-R(17)+V(17)/G/A(17)
C(32,33)=1.
C(32,34)=V(17)/G/A(17)
C(33,41)=1.
C(33,42)=-V(21)/G/A(21)
C(33,46)=V(21)/G/A(21)
C(34,33)=1.
C(34,35)=R(21)+V(21)/G/A(21)
C(35,33)=1.
C(35,34)=-V(18)/G/A(18)
C(35,35)=-V(18)/G/A(18)
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300
00002310
00002320
00002330
00002340
00002350
00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530

```

F S C S M SUBROUTINES

```

C(36,36)=1.
C(36,37)=R(18)+V(18)/G/A(18)
C(37,36)=1.
C(37,37)=-R(19)+V(19)/G/A(19)
C(37,38)=R(19)+V(19)/G/A(19)
C(38,36)=1.
C(38,38)=-V(20)/G/A(20)
C(39,40)=-R(20)+V(20)/G/A(20)
C(40,41)=1.
C(40,46)=-V(25)/G/A(25)
C(41,47)=1.
C(41,48)=R(25)+V(25)/G/A(25)
C(42,47)=1.
C(42,48)=-V(26)/G/A(26)
C(43,49)=1.
C(43,50)=R(26)+V(26)/G/A(26)
C(44,49)=1.
C(44,50)=-R(27)+V(27)/G/A(27)
C(44,51)=R(27)+V(27)/G/A(27)
C(45,49)=1.
C(45,51)=-V(28)/G/A(28)
C(46,52)=1.
C(46,53)=R(28)+V(28)/G/A(28)
C(47,52)=1.
C(47,53)=-V(29)/G/A(29)
C(48,54)=1.
C(48,59)=R(29)+V(29)/G/A(29)
C(49,54)=1.
C(49,59)=-V(30)/G/A(30)
C(50,56)=1.
C(50,57)=R(30)+V(30)/G/A(30)
C(51,56)=1.
C(51,58)=-ZF

```

```

00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860

```

F S C S M SUBROUTINES

```

C(51,59)=-RF
C(52,55)=1.+VF*VF*KF/G/VOLF
C(52,57)=- (VF*VF/G/VOLF)
C(52,59)=VF*VF/G/VOLF
C(52,62)=VF*VF*KF/G/VOLF
C(53,9)=1.
C(53,10)=R(6)+V(6)/G/A(6)
C(54,26)=- (R(12)+V(12)/G/A(12))
C(55,41)=1.
C(55,42)=R(24)+V(24)/G/A(24)
C(56,60)=1.
C(57,61)=1.
C(17,63)=1.
C(51,63)=1.

```

C GENERATE TIME DELAY MATRIX FOR FREQUENCY RESPONSE ROUTINE

```

TD(1,1)=2.*L(1)/V(1)
TD(1,2)=2.*L(1)/V(1)
TD(2,3)=L(3)/V(3)
TD(2,5)=L(3)/V(3)
TD(3,1)=L(3)/V(3)
TD(3,4)=L(3)/V(3)
TD(4,1)=2.*L(2)/V(2)
TD(4,2)=2.*L(2)/V(2)
TD(4,4)=2.*L(2)/V(2)
TD(5,7)=L(5)/V(5)
TD(5,8)=L(5)/V(5)
TD(6,18)=L(4)/V(4)
TD(6,19)=L(4)/V(4)
TD(7,3)=L(4)/V(4)
TD(7,5)=L(4)/V(4)
TD(7,6)=L(4)/V(4)
TD(8,3)=L(5)/V(5)
TD(8,6)=L(5)/V(5)

```

```

00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950
00002960
00002970
00002980
00002990
00003000
00003010
00003020
00003030
00003040
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190

```

F S C S M SUBROUTINES

```

TD(9,9)=L(6)/V(6)
TD(9,10)=L(6)/V(6)
TD(10,9)=2.*L(7)/V(7)
TD(10,10)=2.*L(7)/V(7)
TD(10,11)=2.*L(7)/V(7)
TD(11,12)=L(8)/V(8)
TD(11,13)=L(8)/V(8)
TD(12,9)=L(8)/V(8)
TD(12,11)=L(8)/V(8)
TD(13,14)=L(9)/V(9)
TD(13,25)=L(9)/V(9)
TD(14,12)=L(9)/V(9)
TD(14,13)=L(9)/V(9)
TD(15,15)=L(10)/V(10)
TD(15,16)=L(10)/V(10)
TD(16,14)=L(10)/V(10)
TD(16,25)=L(10)/V(10)
TD(19,26)=L(14)/V(14)
TD(19,27)=L(14)/V(14)
TD(20,19)=L(14)/V(14)
TD(20,20)=L(14)/V(14)
TD(21,21)=L(11)/V(11)
TD(21,22)=L(11)/V(11)
TD(22,18)=L(11)/V(11)
TD(22,19)=L(11)/V(11)
TD(22,20)=L(11)/V(11)
TD(23,21)=2.*L(13)/V(13)
TD(23,22)=2.*L(13)/V(13)
TD(23,23)=2.*L(13)/V(13)
TD(24,24)=L(12)/V(12)
TD(24,56)=L(12)/V(12)
TD(25,28)=2.*L(15)/V(15)
TD(25,29)=2.*L(15)/V(15)

```

```

00003200
00003210
00003220
00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300
00003310
00003320
00003330
00003340
00003350
00003360
00003370
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520

```



F S C S M SUBROUTINES

TD(26,26)=L(16)/V(16)  
 TD(26,27)=L(16)/V(16)  
 TD(26,30)=L(16)/V(16)  
 TD(27,28)=L(16)/V(16)  
 TD(27,29)=L(16)/V(16)  
 TD(28,41)=2.\*L(22)/V(22)  
 TD(28,42)=2.\*L(22)/V(22)  
 TD(29,39)=L(24)/V(24)  
 TD(29,40)=L(24)/V(24)  
 TD(30,41)=2.\*L(23)/V(23)  
 TD(30,42)=2.\*L(23)/V(23)  
 TD(30,43)=2.\*L(23)/V(23)  
 TU(31,31)=L(17)/V(17)  
 TD(31,32)=L(17)/V(17)  
 TD(32,27)=L(17)/V(17)  
 TD(32,30)=L(17)/V(17)  
 TD(33,31)=L(21)/V(21)  
 TD(33,33)=L(21)/V(21)  
 TD(34,39)=L(21)/V(21)  
 TD(34,40)=L(21)/V(21)  
 TD(34,44)=L(21)/V(21)  
 TD(35,34)=L(18)/V(18)  
 TD(35,35)=L(18)/V(18)  
 TD(36,31)=L(18)/V(18)  
 TD(36,32)=L(18)/V(18)  
 TD(36,33)=L(18)/V(18)  
 TD(37,34)=2.\*L(19)/V(19)  
 TD(37,35)=2.\*L(19)/V(19)  
 TD(37,36)=2.\*L(19)/V(19)  
 TD(38,38)=L(20)/V(20)  
 TD(38,57)=L(20)/V(20)  
 TD(39,34)=L(20)/V(20)  
 TD(39,36)=L(20)/V(20)

0000353C  
 00003540  
 00003550  
 00003560  
 00003570  
 00003580  
 00003590  
 00003600  
 00003610  
 00003620  
 00003630  
 00003640  
 00003650  
 00003660  
 00003670  
 00003680  
 00003690  
 00003700  
 00003710  
 00003720  
 00003730  
 00003740  
 00003750  
 00003760  
 00003770  
 00003780  
 00003790  
 00003800  
 00003810  
 00003820  
 00003830  
 00003840  
 00003850

F S C S M SUBROUTINES

```

TD(40,45)=L(25)/V(25)
TU(40,46)=L(25)/V(25)
TD(41,39)=L(25)/V(25)
TD(41,44)=L(25)/V(25)
TD(42,47)=L(26)/V(26)
TU(42,48)=L(26)/V(26)
TD(43,45)=L(26)/V(26)
TU(43,46)=L(26)/V(26)
TD(44,47)=2.*L(27)/V(27)
TD(44,48)=2.*L(27)/V(27)
TD(44,49)=2.*L(27)/V(27)
TU(45,50)=L(28)/V(28)
TD(45,51)=L(28)/V(28)
TD(46,47)=L(28)/V(28)
TU(46,49)=L(28)/V(28)
TD(47,37)=L(29)/V(29)
TD(47,52)=L(29)/V(29)
TU(48,50)=L(29)/V(29)
TD(48,51)=L(29)/V(29)
TD(49,53)=L(30)/V(30)
TD(49,54)=L(30)/V(30)
TD(50,37)=L(30)/V(30)
TD(50,52)=L(30)/V(30)
TD(53,7)=L(6)/V(6)
TD(53,8)=L(6)/V(6)
TD(54,22)=L(12)/V(12)
TD(54,23)=L(12)/V(12)
TD(55,41)=L(24)/V(24)
TD(55,43)=L(24)/V(24)
C GENERATE TIME DELAY COEFFICIENT ARRAY FOR FREQUENCY RESPONSE ROUTINE
ATD(1)=1.
ATD(2)=R(1)-V(1)/G/A(1)
ATD(3)=-1.

```

```

00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060
00004070
00004080
00004090
00004100
00004110
00004120
00004130
00004140
00004145
00004150
00004160
00004170

```

F S C S M SUBROUTINES

ATD(4)=-R(3)-V(3)/G/A(3)	00004180
ATD(5)=-1.	00004190
ATD(6)=-V(3)/G/A(3)	00004200
ATD(7)=-1.	00004210
ATD(8)=R(2)-V(2)/G/A(2)	00004220
ATD(9)=-R(2)-V(2)/G/A(2)	00004230
ATD(10)=-1.	00004240
ATD(11)=-R(5)-V(5)/G/A(5)	00004250
ATD(12)=-R(4)-V(4)/G/A(4)	00004260
ATD(13)=-1.	00004270
ATD(14)=-1.	00004280
ATD(15)=-V(4)/G/A(4)	00004290
ATD(16)=V(4)/G/A(4)	00004300
ATD(17)=-1.	00004310
ATD(18)=-V(5)/G/A(5)	00004320
ATD(19)=-1.	00004330
ATD(20)=-R(6)-V(6)/G/A(6)	00004340
ATD(21)=-1.	00004350
ATD(22)=R(7)-V(7)/G/A(7)	00004360
ATD(23)=-R(7)-V(7)/G/A(7)	00004370
ATD(24)=-1.	00004380
ATD(25)=-R(8)-V(8)/G/A(8)	00004390
ATD(26)=-1.	00004400
ATD(27)=-V(8)/G/A(8)	00004410
ATD(28)=-1.	00004420
ATD(29)=-R(9)-V(9)/G/A(9)	00004430
ATD(30)=-1.	00004440
ATD(31)=-V(9)/G/A(9)	00004450
ATD(32)=-1.	00004460
ATD(33)=-R(10)-V(10)/G/A(10)	00004470
ATD(34)=-1.	00004480
ATD(35)=-V(10)/G/A(10)	00004490
ATD(36)=R(14)-V(14)/G/A(14)	00004500

F S C S M SUBROUTINES

ATD(37)=-1.	00004510
ATD(38)=-1.	00004520
ATD(39)=V(14)/G/A(14)	00004530
ATD(40)=-R(11)-V(11)/G/A(11)	00004540
ATD(41)=-1.	00004550
ATD(42)=-V(11)/G/A(11)	00004560
ATD(43)=-1.	00004570
ATD(44)=-V(11)/G/A(11)	00004580
ATD(45)=K(13)-V(13)/G/A(13)	00004590
ATD(46)=-1.	00004600
ATD(47)=-R(13)-V(13)/G/A(13)	00004610
ATD(48)=-R(12)-V(12)/G/A(12)	00004620
ATD(49)=-1.	00004630
ATD(50)=R(15)-V(15)/G/A(15)	00004640
ATD(51)=1.	00004650
ATD(52)=-R(16)-V(16)/G/A(16)	00004660
ATD(53)=-1.	00004670
ATD(54)=-R(16)-V(16)/G/A(16)	00004680
ATD(55)=-V(16)/G/A(16)	00004690
ATD(56)=-1.	00004700
ATD(57)=1.	00004710
ATD(58)=R(22)-V(22)/G/A(22)	00004720
ATD(59)=-1.	00004730
ATD(60)=-R(24)-V(24)/G/A(24)	00004740
ATD(61)=-1.	00004750
ATD(62)=R(23)-V(23)/G/A(23)	00004760
ATD(63)=-R(23)-V(23)/G/A(23)	00004770
ATD(64)=-1.	00004780
ATD(65)=V(17)/G/A(17)	00004790
ATD(66)=-1.	00004800
ATD(67)=K(17)-V(17)/G/A(17)	00004810
ATD(68)=-1.	00004820
ATD(69)=-K(21)-V(21)/G/A(21)	00004830

F S C S M SUBROUTINES

```

ATD(70)=-1.
ATU(71)=-V(21)/G/A(21)
ATD(72)=V(21)/G/A(21)
ATD(73)=-1.
ATD(74)=-R(18)-V(18)/G/A(18)
ATD(75)=-1.
ATD(76)=-V(18)/G/A(18)
ATD(77)=-V(18)/G/A(18)
ATD(78)=-1.
ATD(79)=R(19)-V(19)/G/A(19)
ATD(80)=-R(19)-V(19)/G/A(19)
ATU(81)=-R(20)-V(20)/G/A(20)
ATU(82)=-1.
ATD(83)=1.
ATD(84)=V(20)/G/A(20)
ATD(85)=-1.
ATD(86)=-R(25)-V(25)/G/A(25)
ATD(87)=-1.
ATD(88)=-V(25)/G/A(25)
ATD(89)=-1.
ATD(90)=-R(26)-V(26)/G/A(26)
ATD(91)=-1.
ATD(92)=-V(26)/G/A(26)
ATD(93)=-1.
ATU(94)=R(27)-V(27)/G/A(27)
ATD(95)=-R(27)-V(27)/G/A(27)
ATD(96)=-1.
ATD(97)=-R(28)-V(28)/G/A(28)
ATU(98)=-1.
ATD(99)=-V(28)/G/A(28)
ATD(100)=-R(29)-V(29)/G/A(29)
ATD(101)=-1.
ATD(102)=-1.

```

```

00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
0005000
0005010
0005020
0005030
0005040
0005050
0005060
0005070
0005080
0005090
0005100
0005110
0005120
0005130
0005140
0005150
0005160

```

F S C S M SUBROUTINES

```

ATD(103)=-V(29)/G/A(29)
ATD(104)=-1.
ATD(105)=-R(30)-V(30)/G/A(30)
ATD(106)=-V(30)/G/A(30)
ATD(107)=-1.
ATD(108)=-1.
ATD(109)=-V(6)/G/A(6)
ATD(110)=1.
ATD(111)=V(12)/G/A(12)
ATD(112)=-1.
ATD(113)=-V(24)/G/A(24)
400 IF(INPHYD.GT.1) GO TO 500
      NW=1
500 CALL FRESP(ICPT,IMWRITE,IX)
      IF(IX.LT.2) GO TO 502
      DO 501 I=1,188
      FSAVE(I)=X(I)
501 X(I)=USAVE(I)
      GO TO 510
502 IF(IX.EQ.0) GO TO 510
      DO 503 I=1,188
503 USAVE(I)=X(I)
      GO TO 5
510 IF(INPHYD.LE.1) GO TO 700
      IF(1WRITE.NE.0)WRITE(6,550)(FREQT(J),GINJOT(J),GINJFT(J),J=1,NFREQ
      6T)
      WRITE(ITAPH) (GINJOT(J),GINJFT(J),J=1,NFREQT)
550 FORMAT(1H1,/,/,18X,'FEED SYSTEM RESPONSE PARAMETERS',/,
      1 5X,'FREQUENCY',4X,'OXIDIZER INJECTION RATE',4X,
      2 'FUEL INJECTION RATE',/,20X,'AMPLITUDE',5X,'PHASE',
      3 6X,'AMPLITUDE',5X,'PHASE',/,5X,0PF9.3,5X,1PE11.4,
      4 0PF9.2,5X,1PE11.4,0PF9.2)
      INPHYD=3
00005170
00005180
00005190
00005200
00005210
00005220
00005230
00005240
00005250
00005260
00005270
00005280
00005290
00005300
00005301
00005302
00005303
00005304
00005305
00005306
00005307
00005308
00005309
00005310
00005320
00005330
00005340
00005350
00005360
00005361
00005362
00005363
00005370

```

F S C S M SUBROUTINES

END FILE ITAPH  
700 GIND=GINJUX  
GINF=GINJFU  
RETURN  
999 STOP  
END

00005380  
00005400  
00005410  
00005420  
00005430  
00005440

F S C S M SUBROUTINES

```

C |
C | SUBROUTINE LOCFAK(JK, X, TX, NX, JX, FX)
C |
C | ROUTINE FOR LINEAR INTERPOLATION
C | IF JK EQ. 1, CHECKS ORDER OF TX ARRAY (NX ITEMS) FOR
C | CONSISTANTLY INCREASING OR DECREASING VALUES.
C | FINDS LOCATION OF FIRST (OR ONLY) ARRAY ITEM FOR SCALING
C | LOCATION OF X FROM TX(JX)
C | CALCULATES SCALING FACTOR FX = (X-TX(JX)) / (TX(JX+1)-TX(JX))
C | PROGRAMMED BY M. D. SCHUMAN, ROCKETDYNE, MAY 1975
C |
C | DIMENSION TX(1)
C |
C | JX = 1
C | FX = 0.
C | IF(NX.LE.1) GO TO 200
C | S = 1.
C | IF(TX(1).GT.TX(NX)) S = -1.
C | XR2 = ABS(TX(NX)-TX(1))*0.5
C | IF(JK.NE.1) GO TO 90
C |
C | JK = 0
C | IF(S.GT.0.) GO TO 30
C | DO 20 I=2,NX
C | IF(TX(I).GT.TX(I-1)) GO TO 50
C | CONTINUE
C | GO TO 90
C | DO 40 I=2,NX
C | IF(TX(I).LT.TX(I-1)) GO TO 50
C | CONTINUE
C | GO TO 90
C |
C | 20 CONTINUE
C | 30 CONTINUE
C | 40 CONTINUE
C |
C | 50 WRITE(6,60)
C | 60 FORMAT(1H1 4X 27ME R R O R I N T A B L E )

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330

```



F S C S M SUBROUTINES

```

70 WRITE(6,80) X,(TX(I),I=1,NX)
80 FORMAT(1H0 4X 27HREFER TO SUBROUTINE LOCFAK //
1 5X 3HX = 1PE15.4 / 4X 4HTX = 6E15.4 / (8X 6E15.4) )
CALL EXIT
STOP
C
90 NX1 = 2
IF(NX.LE.20) GO TO 110
DO 100 I=10,NX,10
JX = I
IF((TX(I)-X)*S) 100,200,110
100 NX1 = I + 1
110 DO 120 I=NX1,NX
JX = I
IF((TX(I)-X)*S) 120,200,130
CONTINUE
120 IF(JX.GT.1) JX = JX-1
FX = (X-TX(JX)) / (TX(JX+1)-TX(JX))
IF(X.LT.AMIN1(TX(1),TX(NX))-XR2) GO TO 150
IF(X.GT.AMAX1(TX(1),TX(NX))+XR2) GO TO 150
GO TO 200
C
150 WRITE(6,160)
160 FORMAT(1H1 22X 64HE R R O R - EXTRAPOLATION OF TABLE IS BEYOND
REASONABLE LIMITS )
GO TO 7C
C
200 RETURN
END
SUBROUTINE NUZADM(IR,GAMX,CO,FREQ,NOZA)
C
C SUBPROGRAM TO CALCULATE THE DOWNSTREAM NOZZLE ADMITTANCE
C PROGRAM DEVELOPED BY GEORGIA INSTITUTE OF TECHNOLOGY

```

F S C S M SUBROUTINES

```

C REF. NASA CR-121129
C MODIFIED FOR THE FEED SYSTEM COUPLED STABILITY MODEL
C BY M. D. SCHUMAN, ROCKETDYNE, MAY 1975
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMPLEX NOZA, NOZAT, GINJOT, GINJFT
C
C REAL FREQ, GAMX, RCCX, RCTX, ANGLEX, CRR, RINJ, CO, FREQMX,
1 FREQMI, FREQT, FREQ1, FREQ2
C
C DIMENSION DY(5,4), G(5), GP(5), Y(5)
C
C COMMON /X1/GAM,SVN,ANGLE,RCT,RCC /X2/T,RT,Q,R1,R2,WC,IP
1 /X3/ZIN,ZII /X4/CM
C
C COMMON /COMTAP/NFREQT, FREQT(100), NOZAT(100), GINJOT(100),
1 GINJFT(100), ITAPN, ITAPC, ITAPH
C
C COMMON /CUMNOZ/ RCCX, RCTX, ANGLEX, CRR, KINJ, INPN0Z, FREQMX,
1 FREQMI, IPRNOZ
C
C IF(INPN0Z.EQ.3) REWIND ITAPN
C
C IF(IR.LE.0) GO TO 2
C
C INPUT DATA REQUIRED BY THE SUBPROGRAM
C
C READ(5,9000) RCCX, RCTX, ANGLEX, CRR
9000 FORMAT(6E12.8)
C WRITE(6,9010) RCCX, RCTX, ANGLEX, CPR
9010 FORMAT(/,5X,'RCCX =',1PE11.4,5X,'RCTX =',E11.4,5X,'ANGLEX =',
1 E11.4,5X,'CRR =',E11.4)
C
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000360
00000370
00000380
00000390
00000400

```

F S C S M SUBROUTINES

```

2 GAM = GAMX
  RCC = RCCX
  RCT = RCTX
  ANGLE = ANGLEX
C
  IF(INPN0Z.EQ.3) WRITE(ITAPN) NFREQT
  IP = 0
  SVN = 0.0
  IF(INPN0Z.LE.1) GO TO 2000
  DMC = 6.283185*RINJ/CO*(FREQMX-FREQMI)/(NFREQT-1)
  WC = 6.283185*RINJ/CO*FREQMI-DWC
  NWC = NFREQT
  GO TO 2010
2000 DMC = 0.0
  WC = 6.283185*RINJ/CO*FREQ
  NWC = 1
2010 CONTINUE
  DP = -0.001
  T = 3.1415927*ANGLE/180.0
  IF(CRK.LE.0.0) GO TO 7
  CMD=0.0
  CM=1./CRR*(2./(GAM+1.)*(1.+(GAM-1.)*0.5*CMD*CMD))
  1  **((GAM+1.)/(2.*(GAM-1.)))
  ER=DABS((CMD-CM)/CM)
  CMD=CM
  IF(ER.GT.1.E-06) GO TO 5
  7 CONTINUE
  IF(IPRNOZ.GT.0.AND.INPN0Z.GT.1)
  1  WRITE(6,1000) CM, SVN, GAM, ANGLE, RCT, RCC
  DO 10 N=1,NWC
  20 WC = WC + DWC
  25 RT = (CM**0.5)*((1+(GAM-1)*CM*CM/2)**((-GAM-1)/(4*(GAM-1))))
  1  *((2/(GAM+1))*((-GAM-1)/(4*(GAM-1))))
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730

```

F S C S M SUBROUTINES

```

Q = (0.25*RT)*((2/(GAM+1))*((GAM+1))/((4*(GAM-1))))
PHIR = 1
PHII = 0
R1 = RT+RCT*(1-DCOS(T))
R2 = 1.0-RCC*(1-DCOS(T))
P = 0
U = 2/(GAM+1)
SRTR = (RT+RCT)*0.5
AIR = -4/((GAM+1)*SRTR)
BUR = -AIR
BOI = 4*WC/(GAM+1)
SVNR = SVN/RT
COR = WC*WC-((SVNR*SVNR)*2/(GAM+1))
COI = -2*WC*(GAM-1)/((GAM+1)*SRTR)
BIR = (24+4*GAM)/(3*RCT*RT*(GAM+1))
BIJ = F*WC/(SRTR*(GAM+1))
CIR = 2*(GAM-1)*SVNR*SVNR/(SRTR*(GAM+1))
CII = -BIR*WC*(GAM-1)*0.5
ZOR = (BUR*COR+BOI*COI)/(BOR*BOR+BOI*BOI)
ZOI = (BOR*COI-BOI*COR)/(BOR*BOR+BOI*BOI)
F1 = BIR*ZOR-BII*ZOI-ZOR*ZOR+AIR*ZOI*ZOI-CIR
F2 = BII*ZOR+EI*ZOI-2*AIR*ZOI*ZOR-CII
ZIR = (F1*(AIR-BOR)-F2*BOI)/((AIR-BOR)*(AIR-BOR)+BOI*BOI)
ZII = (F2*(AIR-BOR)+F1*BOI)/((AIR-BOR)*(AIR-BOR)+BOI*BOI)
C = U
G(1) = U
G(2) = ZOR
G(3) = ZOI
G(4) = PHIR*ZOR-PHII*ZOI
G(5) = PHII*ZOR+ZOI*PHIR
DY(1,1) = -AIR
DY(2,1) = ZIR
DY(3,1) = ZII
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060

```

F S C S M SUBROUTINES

```

DY(4,1) = PHIR
DY(5,1) = PHI
IQZ = 2
DO 30 I=2,4
CALL RKTZ(5,DP,P,G,GP,IQZ)
P = P+DP
U = G(1)
ZR = G(2)
ZI = G(3)
PHIR = G(4)
PHI = G(5)
DY(1,I) = GP(1)
DY(2,I) = GP(2)
DY(3,I) = GP(3)
DY(4,I) = GP(4)
30 DY(5,I) = GP(5)
Y(1) = U
Y(2) = ZR
Y(3) = ZI
Y(4) = PHIR
Y(5) = PHI
CALL ZADAMS(5,DP,P,Y,DY,IQZ)
IF(IP.EQ.1) GO TO 10
U = Y(1)
ZR = Y(2)
ZI = Y(3)
PHIR = Y(4)
PHI = Y(5)
QBAR = U**0.5
C = 1-U*0.5*(GAM-1)
RHO = C**(1/(GAM-1))
F = QBAR/(GAM*RHO)
IF(IQZ.EQ.1) GO TO 35
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390

```

F S C S M SUBROUTINES

```

ZDN = (U*U*ZR*ZR)+(WC+U*ZI)*(WC+U*ZI)
YR = -(U*(ZR*ZR+ZI*ZI)+WC*ZI)*F/ZDN
YI = F*WC*ZR/ZDN
GO TO 40
35 TR = Y(2)
TI = Y(3)
TDN = (J-WC*TI)*(U-WC*TI)+(WC*TR)*(WC*TR)
YR = -F*(U-WC*TI)/TDN
YI = F*WC*TR/TDN
40 SYR = GAM*(C*((GAM+1)/(2*(GAM-1))))*YR
SYI = GAM*(C*((GAM+1)/(2*(GAM-1))))*YI
W = WC*(C*(-0.5))
ZLN = 1/(W*W+SVN*SVN*CM*CM)
IF(W*W-SVN*SVN*(1-C*CM)) 50, 50, 45
45 F = ((W*W-SVN*SVN*(1-C*CM))*0.5)*ZDN
C = SVN*SVN*CM*ZDN
TR = (SYR+C)/F
TI = SYI/F
U = (((1+TR)*(1+TR)+TI*TI)/((1-TR)*(1-TR)+TI*TI))*0.5
ALP = 0.159155*DLOG(U)
U = (1-TR*TR-TI*TI)+((1+TR)*(1+TR)+TI*TI)*((1-TR)*(1-TR)+TI*TI)
1 **0.5
U = U/(2*TI)
BET = 0.31831*DATAN(U)
FREQ1 = WC*CU/(6.283185*RINJ)
FREQ2 = W*CO/(6.283185*KINJ)
IF(IPRNOZ.GT.0.AND.INPNOZ.GT.1)
1 WRITE(6,1005) FREQ1,YR,YI,FREQ2,SVR,SYI,ALP,BET
GO TO 8
50 FREQ1 = WC*CU/(6.283185*RINJ)
FREQ2 = W*CO/(6.283185*RINJ)
IF(IPRNOZ.GT.0.AND.INPNOZ.GT.1)
1 WRITE(6,1005) FREQ1,YR,YI,FREQ2,SYI,ALP,BET

```

00001400  
00001410  
00001420  
00001430  
00001440  
00001450  
00001460  
00001470  
00001480  
00001490  
00001500  
00001510  
00001520  
00001530  
00001540  
00001550  
00001560  
00001570  
00001580  
00001590  
00001600  
00001610  
00001620  
00001630  
00001640  
00001650  
00001660  
00001670  
00001680  
00001690  
00001700  
00001710  
00001720



F S C S M SUBROUTINES

```

SUBROUTINE RKTUIF(P,G,GP)
ROUTINE FOR SUBPROGRAM NOZADM
PROGRAM DEVELOPED BY GEORGIA INSTITUTE OF TECHNOLOGY
REF. NASA CR-121129

IMPLICIT REAL*8(A-H,O-Z)
COMMON /X1/GAM,SVN,ANGLE,RCT,RCC /X2/T,RT,Q,R1,R2,WC,IP
DIMENSION G(5), GP(5)
U = G(1)
TR = G(2)
TI = G(3)
PHIR = G(4)
PHII = G(5)
C = 1-(GAM-1)*U*0.5
R = Q*((C)**(-1.)/(2.*(GAM-1.)))*(U**(-0.25))*4.0
9  FORMAT(3X,'PRINTING FROM CARD 4570',/,3X,'R=',E15.8,
1  3X,'R1=',E15.8,3X,'RT=',E15.8)
IF(R-1) 22,22,50
22 IF(R-R1) 25,30,30
25 DR = -((2*RCT*(R-RT)-(R-RT)*(R-RT))*C.5)/(RT+RCT-R)
GO TO 45
30 IF(R-R2) 35,40,40
35 DR = -DIAN(T)
GO TO 45
40 DR = ((2*RCC*(1-R)-(R-1)*(R-1))*0.5)/(1-R-RCC)
45 UU = -(U**0.75)*(C**((2*GAM-1)/(2*(GAM-1)))/(Q*(1-(GAM+1)*U*.5)))
GP(1) = DU*DK
GO TO 55
50 GP(1) = G.0
55 A = U*(C-U)
BR = U*GP(1)/C
BI = 2*WC*U

```

```

00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950

```



F S C S M SUBROUTINES

```

CR = WC*WC-(SVN*SVN*WC)/(R*R)
CI = -(GAM-1)*WC*U*GP(1)*0.5/C
GP(2) = 1-((BR*TR-BI*TI)-(CR*(TR*TR-TI*TI)-2*CI*TR*TI))/A
GP(3) = (-BR*TI-BI*TR+CI*(TR*TR-TI*TI)+2*CR*TR*TI)/A
T2 = TR*TR+TI*TI
GP(4) = (TR*PHIR-TI*PHII)/T2
GP(5) = (TR*PHII+TI*PHIR)/T2
RETURN
END
00002960
00002970
00002980
00002990
00003000
00003010
00003020
00003030
00003040

```

F S C S M SUBROUTINES

```

C1
C
C
C
C
C
SUBROUTINE RKTZ(NU,H,T1,U,DUM,JOPT)
ROUTINE FOR SUBPROGRAM NOZADM
PROGRAM DEVELOPED BY GEORGIA INSTITUTE OF TECHNOLOGY
REF. NASA CR-121120
IMPLICIT REAL*8(A-H,O-Z)
COMMON /X2/T,KT,Q,R1,R2,WC,IP
DIMENSION U(5), A(5), UZ(5), FZ(4,5), DUM(5)
A(1) = C.6
A(2) = C.0
A(3) = C.5
A(4) = C.5
A(5) = 1.0
TZ = T1
DO 10 J=1,NU
UZ(J) = U(J)
10 DUM(J) = FZ(1,J)
IF(JOPT.EQ.2) GO TO 15
CALL RK2DIF(TZ,UZ,DUM)
GO TO 20
15 CALL RK2DIF(TZ,UZ,DUM)
20 DO 25 J=1,NU
25 FZ(1,J) = DUM(J)
DO 30 I=2,4
TZ = T1+A(I+1)*H
DO 35 J=1,NU
UZ(J) = U(J)+A(I+1)*H*FZ(I-1,J)
35 DUM(J) = FZ(I,J)
IF(JOPT.EQ.2) GO TO 40
CALL RK2DIF(TZ,UZ,DUM)
GO TO 45
40 CALL RK2DIF(TZ,UZ,DUM)

```

```

00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880

```

F S C S M SUBROUTINES

```

45 DO 50 J=1,NU
50 FZ(I,J) = DUM(J)
130 CONTINUE
DO 55 J=1,NU
55 U(J) = U(J)+H*(FZ(1,J)+2*(FZ(2,J)+FZ(3,J))+FZ(4,J))/6.0
60 CALL RKTDIF(TZ,U,DUM)
GO TO 70
65 CALL RKZDIF(TZ,U,DUM)
70 IF(IP.EQ.0) GO TO 75
PR = WC*U(5)-U(1)*DUM(4)
PI = -WC*U(4)-U(1)*DUM(5)
PMAG = DSORT(PR*PR+PI*PI)
PARG = DATAN(PI/PR)
WRITE(6,1000) TZ,PMAG,PARG
1000 FORMAT(46X,F6.4,1X,F10.5,3X,F10.5)
75 RETURN
END
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060

```

F S C S M SUBROUTINES

```

C I
C
C
C
C
C
SUBROUTINE RKZDIF(P,G,GP)
ROUTINE FOR SUBPROGRAM NOZADM
PROGRAM DEVELOPED BY GEORGIA INSTITUTE OF TECHNOLOGY
REF. NASA CR-121129
IMPLICIT REAL*8(A-H,O-Z)
COMMON /X1/GAM,SVN,ANGLE,RCT,RCC /X2/T,RT,Q,R1,R2,WC,IP
1 DIMENSION G(5), GP(5)
U = G(1)
ZR = G(2)
ZI = G(3)
PHIR = G(4)
PHII = G(5)
IF(P) 15,10,15
10 GP(1) = 4/((GAM+1)*((RCT*RT)**0.5))
GP(2) = ZIR
GP(3) = ZII
GP(4) = ZIR
GP(5) = ZII
GO TO 20
15 C = 1-(GAM-1)*U*0.5
R = Q*(C)**(-1.)/(2.*(GAM-1.))*(U**(-0.25))*.4.0
16 FORMAT(3X,'PRINTING FROM CARD 5000',/,3X,'R=',E15.8,
1 3X,'R1=',E15.8,3X,'RT=',E15.8)
IF(R.LT.RT) RT=R
IF(R1.LT.R1) R1=RT
IF(R-1) 22,22,50
22 IF(K-K1) 25,30,30
25 DK = -(2*RCT*(K-RT)-(R-RT)*(R-RT)**0.5)/(RT+RCT-R)
GO TO 45
30 IF(R-R2) 35,40,40
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190
00003200
00003210
00003220
00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300
00003310
00003320
00003330
00003340
00003350
00003360
00003370

```

F S C S M SUBROUTINES

```

35 DR = -DTAN(T)
GO TO 45
40 DR = ((2*RCC*(1-R)-(R-1))*(R-1))*0.5)/(1-R-RCC)
45 DU = -(U**0.75)*(C**((2*GAM-1)/(2*(GAM-1)))/(Q*(1-(GAM+1)*U*.5)))
GP(1) = DU*DR
GO TO 55
50 GP(1) = 0.0
55 A = U*(C-U)
BR = U*GP(1)/C
BI = 2*WC*U
CR = WC*WC-SVN*SVN*C/(R*R)
CI = -(GAM-1)*WC*U*GP(1)*0.5*(1/C)
GP(2) = ((BR*ZR-BI*ZI-CR)/A)-ZR*ZR+ZI*ZI
GP(3) = ((BI*ZR+BR*ZI-CI)/A)-2*ZR*ZI
GP(4) = ZR*PHIR-ZI*PHII
GP(5) = ZR*PHII+ZI*PHIR
20 RETURN
END
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550

```

F S C S M SUBROUTINES

```

SUBROUTINE SOLVW(KWHERE)
SUBPROGRAM SOLVES FOR THE COMPLEX FREQUENCY BASED ON THE
UPSTREAM AND DOWNSTREAM NOZZLE ADMITTANCE
PROGRAMMED BY K. W. FERTIG, ROCKETDYNE, MAY 1975
WHEN ISCNT = 1,4, OR 5 THIS PROGRAM COMPUTES CNOZA-NOZA FOR A
GIVEN OMEGA. WHEN ISCNT = 1 OR 4, THIS ROUTINE ALSO FINDS
IMAG(OMEGA) THAT MINIMIZES CNOZA-NOZA KEEPING REAL(OMEGA) CONSTANT
FURTHER, WHEN ISCNT = 1 OR 4, TESTING IS PERFORMED TO DETERMINE
IF A ROOT IS NEARBY.
COMPLEX OMEGA, P, RHO, V, MR, T, CNOZA, VXO,
1 NOZA, OMEGO,
2 DOM, Z1, Z2, Z3, FN, NOZAMR, GN,
3 CNOZO, NUZU, FNO, FNO, GNO, OMEGSV
COMMON /COMCHM/ P(100), RHO(100), V(100), MR(100), T(100),
1 VXO, OMEGA, CNOZA, DELP
COMMON /FZERO/ NOZA,NOZAMR,GN,FN,FNR,FNI,HN,ISCNT,ISLP
EXTERNAL XIMAGF
DATA PI/3.141593/
COMMON /SOLVE/ FREQ, DELFRQ, DELMX, EPSF, EPSX, EPSFS, EPSXS,
1 FROMAX, CTEST, IPASS, KNTR, ISFR1, KSCNT4, IMSKP,
2 KNTMX, KNTSMX, KNTRMX

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330

```



F S C S M SUBROUTINES

```

115 IF(F1*F2.LE.0.) GO TO 125
C
C USE TIGHTENED CRITERIA TO CHECK TO SEE IF CONVERGENCE HAS BEEN
C REACHED WITHOUT CHANGING SIGN.
C
      IF(ABS(F2/AF).LE.EPSFS.AND.ABS((X1-X2)/AMAX1(1.,X2)).LE.EPSXS
1      .AND.H2.LF.H1) GO TO 145
      IER = 0
      IF(ABS(IWSKP).GE.2) WRITE(6,8490) KNT,IER,X1,X2,F1,F2,OMEGA,
1      FN,GN,HN
8490 FORMAT(/2I5,1F4E13.5/2X,3(1PE13.5,' ',1PE12.5),1PE15.5)
      IF(H2.LT.H1) GO TO 117
C
C LAST STEP TOO LARGE. CUT GUESS IN HALF.
C
      X2 = (X1+X2)/2.
      GO TO 116
117 IF(X1.LT.X2.AND.F1.LT.F2) GO TO 120
      IF(X1.GT.X2.AND.F2.LT.F1) GO TO 120
C
C ESTIMATE NEXT GUESS BY TRYING TO HALVE THE ERROR.
C
      X3 = X2 - HN**2/(2.*F2)
      IER = -10
      KNT = KNT+1
118 IF(KNT.GT.KNTMX) GO TO 140
      X3 = AMIN1(X2+DELMX,AMAX1(X2-DELMX,X3))
      X1 = X2
      F1 = F2
      X2 = X3
      GO TO 116
C
C ESTIMATE NEXT GUESS USING FALSE POSITION.
C

```



F S C S M SUBROUTINES

```

C      120 X3 = (F2*X1-F1*X2)/(F2-F1)
          GO TO 118
C      125 FANS = F2
          ANS = AIMAG(OMEGA)
C      135 IF(ABS(FANS)/AF.LE.EPSF.AND.ABS((X1-X2)/AMAX1(1.,ANS)).LE.
          1 EPSX) GO TO 145
          IF(X1.LT.X2) GO TO 137
          X3 = X2
          F3 = F2
          X2 = X1
          F2 = F1
          X1 = X3
          F1 = F3
C      137 K = 0
          ROOT IS BRACKETED BUT CONVERGENCE NOT YET REACHED. CALL ZERO
          TO FIND ROOT WITHIN EPSX.
          CALL ZERO(XIMAGF,X1,X2,F1,F2,ANS,FANS,EPXF,EPXAF,KNT,KNTMX,IER,K
          1 )
          IF(IER.EQ.0) GO TO 145
C      140 WRITE(6,8601) X1,F1,X2,F2,X3,F3,ANS,FANS,KNT,IER,OMEGA
          8601 FORMAT(//' *** UNABLE TO FIND ROOT FOR IMAG PART OF F ****'
          1 //' X1,F1,X2,F2,X3,F3,ANS,FANS,KNT,IER,OMEGA = ',/
          2 3X,1P8E13.5/3X,2I10/3X,1PE13.5,' : ',1PE13.5)
          GO TO 5000
C      (CNOZA-NOZA) HAS BEEN MINIMIZED W.R.T. IMAG(OMEGA). SAVE VALUES
          FOR COMPUTATION OF DERIVATIVES W.R.T. REAL(OMEGA).
C      145 CNOZO = CNOZA
          NOZO = NOZA

```

00001000  
00001010  
00001020  
00001030  
00001040  
00001050  
00001060  
00001070  
00001080  
00001090  
00001100  
00001110  
00001120  
00001130  
00001140  
00001150  
00001160  
00001170  
00001180  
00001190  
00001200  
00001210  
00001220  
00001230  
00001240  
00001250  
00001260  
00001270  
00001280  
00001290  
00001300  
00001310  
00001320

F S C S M SUBROUTINES

```

FR3 = FNR
FI3 = FNI
XR3 = REAL(OMEGA)
XI3 = AIMAG(OMEGA)
FND = FN
HND = HN
GND = GN
OMEGA = OMEGA
DFRDY = REAL(GN)
DFIDY = AIMAG(GN)
DM = AMAX1(.01,.001*REAL(OMEGA))
ISLP = C
OMEGA = OMEGA+DM
FREQ = REAL(OMEGA)/(2.*PI)
GO TO 15

```

```

COMPUTE DERIVATIVES, FTEST2( TEST FUNCTION), DET2(DETERMINANT
OF JACOBIAN), AND CUND2(CONDITION NUMBER OF JACOBIAN).

```

```

146 DFRDX = (FNR-FR3)/DM
DFIDX = (FNI-FI3)/DM
OMEGA = OMEGA
FREQ = REAL(OMEGA)/(2.*PI)
CNOZA = CNOZO
NOZA = NOZO
FN = FND
GN = GND
HN = HND
DET2 = DFRDX*DFIDY - DFIDX*DFRDY
FNR = FR3
FNI = FI3
FTEST2 = 2.*(FNR*DFRDX + FNI*DFIDX)/DET2

```

00001330  
00001340  
00001350  
00001360  
00001370  
00001380  
00001390  
00001400  
00001410  
00001420  
00001430  
00001440  
00001450  
00001460  
00001470  
00001480  
00001490  
00001500  
00001510  
00001520  
00001530  
00001540  
00001550  
00001560  
00001570  
00001580  
00001590  
00001600  
00001610  
00001620  
00001630  
00001640  
00001650

C  
C  
C  
C  
C

!



F S C S M SUBROUTINES

```

150 FR2 = FR3
    XR2 = XR3
    FI2 = FI3
    XI2 = XI3
    FTST1 = FTST2
    COND1 = COND2
    DET1 = DET2
C
C IF ISCNT = 1, UPDATE FREQ BY DELFRQ. IF ISCNT = 4, FREQ HAS
C ALREADY BEEN CHANGED BY METHOD OF FALSE POSITION.
C
    IF(ISCNT.EQ.4) GO TO 1G
    ISTR1 = 1
    FREQ = FREQ + DELFRQ
    GO TO 1C
160 IF(ISCNT.EQ.1) GO TO 18C
    ISCNT = 3
170 IF(ABS(FR3)/CABS(NOZA).LE.EPSFS.AND. A.MINI(ABS((XR3-XR1)/XR3),
    1 ABS((XR3-XR2)/XR3)).LE.EPSXS) GO TO 22G
    IF(FR1*FR3.LE.0.) GO TO 175
    FR1 = FR3
    XR1 = XR2
    GO TO 177
175 XR2 = XR3
    FR2 = FR3
177 XR3 = (FR2*XR1-FR1*XR2)/(FR2-FR1)
    FREQ = XR3/(2.*PI)
    GO TO 1G
C
C TENTATIVE ROOT HAS BEEN BRACKETED. CONTINUE WITH ISCNT = 4(I.E.
C FREQ WILL BE CHANGED BY FALSE POSITION) UNTIL EITHER IT IS
C DETERMINED THAT THE JACOBIAN IS SINGULAR OR THAT A ROOT IS
C ACTUALLY THERE. THE LATTER IS ASSUMED BY DEFAULT IF THREE

```

```

00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300
00002310

```

F S C S M SUBROUTINES

```

C      ITERATIONS HAVE BEEN PERFORMED WITHOUT THE JACOBIAN CHANGING SIGN000002220
C      AND THE CONDITION NUMBER REMAINS LESS THAN CTEST AND THE ERROR 00002330
C      (CABS(CNOAZA-NOZA)) IS LESS THAN 2.5* EPSF. 00002340
C      00002350
C      00002360
C      00002370
C      00002380
C      00002390
C      00002400
C      00002410
C      00002420
C      00002430
C      00002440
C      00002450
C      00002460
C      00002470
C      00002480
C      00002490
C      00002500
C      00002510
C      00002520
C      00002530
C      00002540
C      00002550
C      00002560
C      00002570
C      00002580
C      00002590
C      00002600
C      00002610
C      00002620
C      00002630
C      00002640

180  ISCNT = 4
      IP = 0
      OMEGSV = OMEGA
      DET2S = DET2
      FTST2S = FTST2
      COND2S = COND2
      FR3S = FR3
      XR3S = XR3
      FI3S = FI3
      XI3S = XI3
      KSCNT4 = 0
181  FTSTU = FTST1
      DETO = DET1
      CONDU = COND1
      IP = MAX0(IP,0) + 1
182  KSCNT4 = KSCNT4 + 1
      XR4 = (FTST2*XR2-FTST1*XR3)/(FTST2-FTST1)
      IF (IABS(IP).LE.2) GO TO 1825
      IP = 0
      XR4 = (XR3+XR2)/2.
1825 CONTINUE
      OMEGA = XR4 + (0.,1.)*AIMAG(OMEGA)
      FREQ = XR4/(2.*PI)
      GO TO 147
183  KNTR = KNTR - 1
      IF(COND2.GT.CTEST.OR.DET2*DETI.LE.0.) GO TO 184
      IF((HN/CABS(NOZA)).LE.EPSFS.AND.AMIN1(CABS(OMEGA-XR2-(0.,1.)*XI2)
1      ,CABS(UMEGA-XR1-(0.,1.)*XI1)/CABS(OMEGA)).LE.EPSXS) GO TO 220
      IF((HN/CABS(NOZA)).LE.2.5*EPSF.AND.KSCNT4.GE.3) GO TO 185

```



F S C S M SUBROUTINES

```

C      TU CONVERGE ON THIS ROOT (UP TO STATEMENT 205).
C
185  ISCNT = 5
      KNTS = 0
190  DET = FR2*FI1-FR1*FI2-FR3*FI1+FR1*FI3+FR3*FI2-FI3*FR2
      PIO = (FR2*FI1-FI2*FR1)/DET
      PII = -(FR3*FI1-FR1*FI3)/DET
      PI2 = (FR3*FI2-FR2*FI3)/DET
      XI4 = PI0*XI3 + PII*XI2 + PI2*XI1
      XR4 = PIC*XR3 + PII*XR2 + PI2*XR1
      OMEGA = OMEGA
      OMEGA = XR4 + (0.,1.)*XI4
      DOM = OMEGA - OMEGO
      SF = AMINI(1.,50./CABS(DOM))
      OMEGA = OMEGO + SF*DOM
      XR4 = REAL(OMEGA)
      XI4 = AIMAG(OMEGA)
      FREQ = REAL(OMEGA)/(2.*PI)
      GO TO 15
195  FR4 = FNK
      FI4 = FNI
      KNTS = KNTS + 1
      IF(IABS(IWSKP).GE.1) WRITE(6,8515) ISCNT,KNTS,OMEGA,CNOZA,NOZA,HN
8515  FORMAT(/2I5,3(1PE14.5,0 : 0,1PE12.5),1PE14.5)
      IF(IABS(IWSKP).GE.2) WRITE(6,8520) XR1,XI1,FR1,FI1,XR2,XI2,FR2,FI2
1      ,XR3,XI3,FR3,FI3,XR4,XI4,FR4,FI4,FXN
8520  FORMAT(2X,4(1PE14.5,0 : 0,1PE12.5)/2X,4(1PE14.5,0 : 0,
1      1PE12.5)/2X,1PE14.5,0 : 0,1PE12.5)
      Z1 = XR1 + (0.,1.)*XI1
      Z2 = XR2 + (0.,1.)*XI2
      Z3 = XR3 + (0.,1.)*XI3
      ADM = CABS(OMEGA)
C
00002980
00002990
00003000
00003010
00003020
00003030
00003040
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190
00003200
00003210
00003220
00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300

```





F S C S M SUBROUTINES

```

F11 = F14
FR1 = FR4
GO TO 190
202 IF(AF3.GT.AF2) GO TO 203
X12 = X14
XR2 = XR4
F12 = F14
FR2 = FR4
GO TO 190
203 X13 = X14
XR3 = XR4
F13 = F14
FR3 = FR4
GO TO 190
205 WRITE(6,8602) IER,KNTS,KNTR,ISCNT,XR1,XI1,FR1,F11,XR2,XI2,FR2,F12
1 ,XR3,XI3,FR3,F13,XR4,XI4,FR4,F14
8602 FORMAT(//, '*** EXCEED CONVERGENCE LIMIT ****'/
1 , IER,KNTS,KNTR,ISCNT = ,4I10/, X,F FROM 1-4 = ,/
2 (3X,1PE15.5, : ,1PE15.5,1PE18.5, : ,1PE15.5))
GO TO 5000
C
C
C
C
10 KWHERE = 1
GO TO 6000
15 KWHERE = 2
GO TO 6000
220 KWHERE = 3
GO TO 6000
225 KWHERE = 4
GO TO 6000
5000 KWHERE = 5
C
C
C
C
KWHERE DETERMINES TRANSFER LOCATION IN CALLING PROGRAM
UPON RETURN.

```

```

00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960

```

F S C S M SUBROUTINES

6000 RETURN  
1 END

00003970  
00003980

F S C S M SUBROUTINES

```

SUBROUTINE STEADY(IPRSTE)
C
C
C SUBPROGRAM STEADY FOR THE FEED SYSTEM COUPLED STABILITY MODEL
C PROGRAM CALCULATES STEADY STATE PARAMETERS
C PROGRAMMED BY M. D. SCHUMAN, ROCKETDYNE, MAY 1975
C
C
C COMPLEX COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
C COX10, COX11, COX12, COX13, COX14, COX15, COX16,
C CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
C CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
C
C REAL MRBINJ, MBOXI, MBFUI, MRB, MRGI, MGI, NUBOX, NUBFU, MWG
C
C COMMON /COMCBM/ XKOX, XKFU, MBOXI, MBFUI, TAUBOX, TAUBFU, VBOX,
C VBFU, GAMO, RGO, DELHOX, DELHFU, PC, CO,
C COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
C COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
C CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
C CFU12, CFU13, CFU14, CFU15, CFU16, MWG, XIMPFU, XIMPOX,
C CS, DCSDMR, OHDMR, DRGDMR, ADVUX, ADDOX, TDRAGO, DELVOX,
C NUBOX, DTOXDM, ADVFU, ADDFU, TORAGF, DELVUFU, NUBFU, DTFUDM
C
C COMMON /CONSTS/ MRB(100), TB(100), KHDB(100), VB(100),
C DMRB(100), DRHDB(100), DVB(100), VAPBOX(100), VAPBFU(100),
C SSV1(100), SSV2(100), SSV3(100), SSV4(100),
C SSV5(100), SSV6(100), SSV7(100), SSV8(100), SSV9OX(100),
C SSV9FU(100), SSV10(100), SSV11(100), SSV12(100), SSV13(100),
C SSV14(100), SSV15(100), SSV16(100),
C RHOGI, VGI, MRGI, MGI
C
C COMMON /COMARE/ NXP, X(100), XM(100), A(100), DA(100), DELX,
C X0, XNOZ, AINJ
C
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330

```

F S C S M SUBROUTINES

```

C
C |
PRINT TITLES FOR STEADY STATE VARIABLES
IF(1PRSTE.GT.0) WRITE(6,10)
10 FORMAT(1H1,/,/,31X,'STEADY STATE SOLUTION',/,/,
1 7X,'DISTANCE',
2 4X,'TEMPERATURE',2X,'VELOCITY',3X,'MIXTURE',
3 6X,'DENSITY',7X,'PERCENT VAPORIZED',/,7X,'(INCHES)',5X,
4 '(RANKINE)',4X,'(FT/S)',5X,'RATIO',5X,
5 '(LBM/FT**3)',6X,'FUEL',3X,'OXIDIZER',/)
C
C |
INJECTOR END GAS MIXTURE RATIO AND GAS DENSITY
MRBINJ = MDOXI/MBFUI
IF(MGI.GT.0.0) GO TO 15
MRGI = MRBINJ*TAUBFU*VBFU/(TAUBOX*VBOX)
RHOGI = GAMC*PC*(MRGI+1.)/(GAMD-1.)/(MRGI*DELHOX+DELHFU
1 -MRGI*(1.+MRGI)*DHDMR)
GO TO 16
15 RHOGI = MGI/(AINJ*VGI)
16 CONTINUE
C
C |
MAIN DISTANCE DO LOOP
DO 30 I=1,NXP
C
C |
VAPORIZATION RATE PARAMETERS
PHICX = EXP(-(XM(I)-X0)/(TAUBOX*VBOX))
PHIFU = EXP(-(XM(I)-X0)/(TAUBFU*VBFU))
C
C |
GAS MIXTURE RATIO, VELOCITY, DENSITY, TEMPERATURE, OXIDIZER
AND FUEL VAPORIZATION RATES
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660

```

F S C S M SUBROUTINES

```

MRB(I) = (MRGI/(1.+MRGI)*MGI+MBOXI*(1.-PHIOX))/(1./(1.+MRGI))*
1 MGI+MBFUI*(1.-PHIFU)
AMDDOT = MGI+MBOXI*(1.-PHIOX)+MBFUI*(1.-PHIFU)
VB(I) = 1./(GAMO*PC*A(I))*(GAMO*PC*MGI/RHOGI+(GAMO-1.))*
1 MBOXI*DELHOX*(1.-PHIOX)+MBFUI*DELHFU*(1.-PHIFU)-AMDDOT*
2 MKB(I)*DHDMR+MGI*MRGI*DHDMR))
RHOB(I) = AMDDOT/(A(I)*VB(I))
TB(I) = PC/(RHOB(I)*RGO*(1.+DRGDMR/RGO*(MRB(I)-MRBINJ)))
VAPBOX(I) = MBOXI*PHIOX/(TAUBOX*VBOX)/A(I)
VAPBFU(I) = MBFUI*PHIFU/(TAUBFU*VBFU)/A(I)

C
C DERIVATIVES WITH RESPECT TO DISTANCE
C
DMRB(I) = (MRB(I)+1.)/(RHOB(I)*VB(I))*(VAPBOX(I)-MRB(I))*
1 VAPBFU(I))
DVB(I) = -VB(I)/A(I)*DA(I)+(GAMO-1.)/(GAMO*PC)*(VAPBOX(I))*
1 (DELHOX-DHDMR*(2.*MRB(I)+1.))+VAPHFU(I)*(DELHFU+
2 DHDMR*MRB(I)*MRB(I))
DRHOB(I) = -RHOB(I)/VB(I)*DVB(I)-RHOB(I)/A(I)*DA(I)+
1 (VAPBOX(I)+VAPBFU(I))/VB(I)

C
C STEADY STATE PARAMETERS REQUIRED BY THE CHAMBER DYNAMICS
C SUBPROGRAM
C
SSV1(I) = VB(I)/CO
SSV2(I) = 2.*DELX*(GAMO-1.)/(PC*CO)
SSV3(I) = DELX*DVB(I)/CO
SSV4(I) = DELX*SSV1(I)*(DA(I)/A(I)+DRHGB(I)/RHOB(I))+SSV3(I)
SSV5(I) = DELHOX-DHDMR*(2.*MRB(I)+1.)
SSV6(I) = DELHFU+DHDMR*MRB(I)*MRB(I)
SSV7(I) = DELX*GAMO*(SSV1(I)*DA(I)/A(I)+DVB(I)/CO)
SSV8(I) = DELX*GAMO*DA(I)/A(I)
SSV9(I) = VAPBOX(I)*DHDMR

```

00000670  
00000680  
00000690  
00000700  
00000710  
00000720  
00000730  
00000740  
00000750  
00000760  
00000770  
00000780  
00000790  
00000800  
00000810  
00000820  
00000830  
00000840  
00000850  
00000860  
00000870  
00000880  
00000890  
00000900  
00000910  
00000920  
00000930  
00000940  
00000950  
00000960  
00000970  
00000980  
00000990

F S C S M SUBROUTINES

```

SSV9FU(I) = VAP8FU(I)*DHDMR*MRB(I)
SSV10(I) = 2*DELX/(RHOB(I))*CO)
SSV11(I) = DELX*(DA(I)/A(I)+DRHOB(I)/RHO8(I))
SSV12(I) = 2.*PC/(RHOB(I))*CO*CO)
SSV13(I) = 2.*DELX*(MRB(I)+1.)/ (RHOB(I))*CO)
SSV14(I) = DELX*SSV1(I)*DMRB(I)
SSV15(I) = DELX*(VAPBOX(I)-(2.*MRB(I)+1.)*VAP8FU(I))/
      (RHOB(I))*CO)
1 IF(1.GT.1) SSV16(I-1) = DRGDMR*(RHOB(I-1)*TB(I-1)/PC+RHOB(I))*
1 TB(I)/PC)/2.
C
C PRINTOUT OF STEADY STATE VARIABLES
C
IF(IPRSTE.LE.0) GO TO 30
PBOX = 100.*(1.-PHIOX)
P8FU = 100.*(1.-PHIFU)
XP = XM(I)/C.0254
TP = TB(I)*1.6
VP = VB(I)/0.2048
RHOP = PHOB(I)/16.01846
WRITE(6,20) XP, TP, VP, MRB(I), RHOP,
1 P8FU, PBOX
20 FORMAT(6X,F9.4,2X,2F11.2,F11.4,1PE15.5,0PF11.2,F9.2)
C
30 CONTINUE
C
RETURN
END

```

00001000  
00001010  
00001020  
00001030  
00001040  
00001050  
00001060  
00001070  
00001080  
00001090  
00001100  
00001110  
00001120  
00001130  
00001140  
00001150  
00001160  
00001170  
00001180  
00001190  
00001200  
00001210  
00001220  
00001230  
00001240  
00001250  
00001260  
00001270

F S C S M SUBROUTINES

```

SUBROUTINE TADAMS(N,H,X,Y,DY,IQZ,IQ)
ROUTINE FOR SUBPROGRAM NOZADM
PROGRAM DEVELOPED BY GEORGIA INSTITUTE OF TECHNOLOGY
REF. NASA CR-121129

IMPLICIT REAL*(A-H,O-Z)
COMMON /X1/GAM,SVN,ANGLE,RCT,RCC /X2/T,RT,Q,R1,R2,WC,IP
COMMON /X4/CM
DIMENSION COR(5), DP(5), DY(5,4), PRED(5), Y(5), G(5), GP(5)
10 CONTINUE
DO 15 I=1,N
  PRED(I) = Y(I)+H*(55.*DY(I,4)-59.*DY(I,3)+37.*DY(I,2)-9.*DY(I,1))
  1 /24.0
15 CONTINUE
X = X+H
U = PRED(1)
TR = PRED(2)
TI = PRED(3)
PHIR = PRED(4)
PHII = PRED(5)
C = 1.-(GAM-1.)*U*0.5
R = Q*(C)**(-1./(2.*(GAM-1.)))*(U**(-0.25))*4.0
16 FORMAT(3X,'PRINTING FROM CARD 2180',/,3X,'R=',E15.8,
1 3X,'R1=',E15.8,3X,'RT=',E15.8)
IF(R-1.) 17,17,100
17 IF(R-R1) 20,25,25
20 DR = -((2.*RCT*(R-RT)-(R-RT)*(R-RT))**.5)/(RT+RCT-R)
GO TO 40
25 IF(R-R2) 30,35,35
30 DR = -DTAN(T)
GO TO 40
35 DR = ((2.*RCC*(1.-R)-(1.-R)*(1.-R))**.5)/(1.-R-RCC)

```

F S C S M SUBROUTINES

```

40 DU = -(U**.75)*(C**((2.*GAM-1.)/(2.*(GAM-1.)))/(O*(1.-((GAM+1.))
1 *U*.5))
UP(1) = UR*DU
A = U*(C-U)
BR = U*DP(1)/C
BI = 2.*WC*U
CR = WC*WC-(SVN*SVN*G)/(R*R)
CI = -(GAM-1.)*WC*U*DP(1)*.5/C
DP(2) = 1.+(-BR*TR+BI*TI+CR*(TR*TR-TI*TI)-2.*CI*TR*TI)/A
DP(3) = (-BR*TI-BI*TR+CI*(TR*TR-TI*TI)+2.*CR*TR*TI)/A
T2 = TR*TR+TI*TI
DP(4) = (TR*PHIR-TI*PHII)/T2
DP(5) = (TR*PHII+TI*PHIR)/T2
DE 45 I=1,N
COR(I) = Y(I)+H*(DY(I,2)-5.*DY(I,3)+19.*DY(I,4)+9.*DP(I))/24.0
45 Y(I) = (251.*COR(I)+19.*PRED(I))/270.
U = Y(1)
TK = Y(2)
TI = Y(3)
PHIR = Y(4)
PHII = Y(5)
C = 1.-((GAM-1.)*U*.5
52 DO 55 I=1,N
DY(I,1) = DY(I,2)
UY(I,2) = DY(I,3)
55 DY(I,3) = DY(I,4)
T2 = TR*TR+TI*TI
TMAG = T2**.5
IF(TMAG-IU.) 60,9C,90
60 K = Q*(C)**(-1./((2.*(GAM-1.)))*(U**(-0.25))*.4.0
61 FORMAT(3X,'PKINTING FROM CARD 2570',/,3X,'R=',E15.8,
1 3X,'K1=',E15.8,3X,'RT=',E15.8)
IF(R-1.) 62,62,100
00000340
00000350
00000360
0000037C
00000380
00000390
00000400
00000410
00000420
0000043C
00000440
00000450
00000460
00000470
0000048C
0000049C
00000500
00000510
00000520
0000053C
0000054C
00000550
00000560
00000570
00000580
0000059C
00000600
00000610
00000620
0000063C
00000640
0000065C
00000660

```



F S C S M SUBROUTINES

```

62 IF(R-R1) 65,70,70
65 DR = -((2.*RCT*(R-RT)-(R-RT)*(R-RT))**.5)/(RT+RCT-R)
GO TO 85
70 IF(R-R2) 75,80,80
75 DR = -DTAN(T)
GO TO 85
80 DR = ((2.*RCC*(1.-R)-(1.-R)*(1.-R))**.5)/(1.-R-RCC)
85 DU = -(U**.75)*(C**((2.*CAM-1.)/(2.*(GAM-1.)))/(Q*(1.--(GAM+1.)*
1 *U*.5)))
DY(1,4) = DR*DU
A = U*(C-U)
BR = U*DY(1,4)/C
BI = 2.*WC*U
CR = WC*WC-(SVN*SVN*C)/(R*R)
CI = -(GAM-1.)*WC*U*DY(1,4)*0.5/C
DY(2,4) = 1.+(-BR*TR+BI*TI+CR*(TR*TR-TI*TI)-2.*CI*TR*TI)/A
DY(3,4) = (-BR*TI-BI*TR+CI*(TR*TR-TI*TI)+2.*CR*TR*TI)/A
DY(4,4) = (TR*PHIR-PHII*TI)/T2
DY(5,4) = (TR*PHII+PHIR*TI)/T2
IF(IP.EQ.0) GO TO 87
PR = WC*PHII-U*DY(4,4)
PI = -WC*PHIR-U*DY(5,4)
PMAG = (PR*PR+PI*PI)**.5
PARG = DATAN(PI/PR)
WRITE(6,1000) X,PMAG,PARG
87 GO TO 1C
90 IQZ = 2
Y(2) = TR/T2
Y(3) = -TI/T2
TPI = DY(2,4)
TPI = DY(3,4)
DY(2,4) = -(TPI*(TR*TR-TI*TI)+2.*TR*TI*TPI)/(T2*T2)
DY(3,4) = (2.*TPI*TR*TI-TPI*(TR*TR-TI*TI))/(T2*T2)
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990

```

F S C S M SUBROUTINES

```

G(1) = U
G(2) = Y(2)
G(3) = Y(3)
G(4) = PHIR
G(5) = PHII
DY(1,1) = DY(1,4)
DY(2,1) = DY(2,4)
DY(3,1) = DY(3,4)
DY(4,1) = (PHIR*TR-PHII*TI)/T2
DY(5,1) = (PHII*TR-PHIR*TI)/T2
DO 95 I=2,4
CALL RKTZ(5,H,X,G,GP,IOZ)
X = X+H
U = G(1)
ZR = G(2)
ZI = G(3)
PHIR = G(4)
PHII = G(5)
DY(1,I) = GP(1)
DY(2,I) = GP(2)
DY(3,I) = GP(3)
DY(4,I) = GP(4)
DY(5,I) = GP(5)
95 Y(1) = U
Y(2) = ZR
Y(3) = ZI
Y(4) = PHIR
Y(5) = PHII
IQ = 1
GO TO 105
100 IQ = 2
1000 FORMAT(40X,F0.4,1X,F10.5,3X,F10.5)
105 RETURN

```

```

00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320

```

F S C S M SUBROUTINES

00001330

END

F S C S M SUBROUTINES

```

C |
C |
C |
C |
SUBROUTINE TDPLOT(W,Y,NFP,TL,XR,LL )
ROUTINE TO GENERATE CRT PLOTS
PROGRAMMED BY J. K. HUNTING, ROCKETDYNE, MAY 1975

DIMENSION W(101),Y(101)
YMAX=Y(2)
YMIN=Y(2)
DO1COJ=3,NFP
YMAX=AMAX1(YMAX,Y(J))
YMIN=AMIN1(YMIN,Y(J))
100 CONTINUE
DY=AINT(2.0+(YMAX-YMIN)/10.0)
YT=DY*AINT(1.5+YMAX/DY)
YB=DY*AINT(YMIN/DY-1.5)
IF(W(2)/TL-10.1)201,201,5
5 XL=W(2)
I=1
GO TO 200
201 YMAX=AMAX1(YMAX,Y(I))
YMIN=AMIN1(YMIN,Y(I))
XL=TL
I=2
200 CONTINUE
K=0
L=0
10 L=L+1
IF(L.GT.9)GO TO 30
IF(10.0**K/XL.LE.1.0.AND.10.0**K/XL.GT.0.1)GO TO 20
IF(10.0**K/XL.GT.1.0)K=K-1
IF(10.0**K/XL.LE.0.1)K=K+1
GO TO 10
20 XL=10.0**K
00011000
00011001
00011002
00011003
00011004
00011010
00011020
00011030
00011040
00011050
00011060
00011070
00011080
00011090
00011100
00011110
00011120
00011130
00011140
00011150
00011160
00011170
00011180
00011190
00011200
00011210
00011220
00011230
00011240
00011250
00011260
00011270
00011280

```

F S C S M SUBROUTINES

```

30 K=0
   IF(LL-1) 31,31,60
31 CALL CAMRAV(9)
   IF(LL-1) 50,50,60
50 CALL SETMIV(40,10,24,560)
   GO TO 61
60 CALL SETMIV(40,10,500, 70)
61 CALL SMXYV(1,0)
   CALL GRIDIV(LL,XL,XR,YB,YT,1.0,DY,1,1,-1,-1,-2, -3)
180 GO TO(181,182),I
181 CALL APLDITV(NFP-1,W(2),Y(2),1,1,1,1,44,IERR)
   GO TO 40
182 CALL APLDITV(NFP,W(1),Y(1),1,1,1,1,44,IERR)
40 K=K+1
   GO TO (180,180,180,190),K
190 IF(LL-1) 191,191,192
191 CALL LABLV(Y(1),600,470, 6, 2, 3)
   GO TO 193
192 CALL LABLV(Y(1),600,995,6,2,3)
193 RETURN
   END
00011290
00011300
00011310
00011320
00011330
00011340
00011350
00011360
00011370
00011380
00011390
00011400
00011410
00011420
00011430
00011440
00011450
00011460
00011470
00011480
00011490

```

F S C S M SUBROUTINES

```

FUNCTION XIMAGF(XIMOMG)
ROUTINE TO COMPUTE NOZZLE ADM. DIFF AS FUNCTION OF
IMAGINARY PART OF OMEGA
PROGRAMMED BY K. W. FERTIG, ROCKETDYNE, MAY 1975

COMPLEX OMEGA,NOZA,CNOZA,P,RHO,V,MR,T,VXG,FN,NOZAMR,GN,
1 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9,
2 COX10, COX11, COX12, COX13, COX14, COX15, COX16,
3 CFU1, CFU2, CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9,
4 CFU10, CFU11, CFU12, CFU13, CFU14, CFU15, CFU16
5 ,OMLGO

REAL MBOXI, MBFUI, NUBOX, NUBFU, MWG

COMMON /COMCHM/ XKOX, XKFU, MBOXI, MBFUI, TAUBOX, TAUBFU, VBOX,
1 VBFU, GAMO, RGO, DELMOX, DELHFU, PC, CO,
2 COX1, COX2, COX3, COX4, COX5, COX6, COX7, COX8, COX9, COX10,
3 COX11, COX12, COX13, COX14, COX15, COX16, CFU1, CFU2,
4 CFU3, CFU4, CFU5, CFU6, CFU7, CFU8, CFU9, CFU10, CFU11,
5 CFU12, CFU13, CFU14, CFU15, CFU16, MWG, XIMPFU, XIMPOX,
6 CS, DCSDMR, DHDMR, DRGDMR, ADVOX, ADDOX, TDRAGO, DELVOX,
7 NUBOX, DTOXDM, ADVFU, ADDFU, TDRAGF, DELVFU, NUBFU, DTFUDM

COMMON /FZERO/ NOZA,NOZAMR,GN,FN,FNR,FNI,HN,ISCNT,ISLP

COMMON /COMCHM/ P(100),RHO(100),V(100),MR(100),T(100),
1 VXO,OMEGA,CNOZA,DELPH

COMMON /COMARE/ NXP,X(100),XM(100),A(100),DA(100),DELX,
1 XO,XNOZ,AIPIJ

RECOMPUTE OMEGA USING NEW IMAGINARY PART.

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330

```

F S C S M SUBROUTINES

```

C      OMEGA = REAL(OMEGA) + (0.,1.)*XIMOMG
C      IP = 0
C      IF((ISCNT.NE.1.AND.ISCNT.NE.4).OR.ISLP.NE.1) GO TO 10
C      SAVE OMEGA THEN ALTER IMAG PART BY 0.1 PERCENT TO COMPUTE DERIV.
C      OMEGO = OMEGA
C      DM = AMAX1(.001*AIMAG(OMEGA),.1)
C      OMEGA = OMEGA + (0.,1.)*DM
C      CHAMDY COMPUTES UPSTREAM NOZZLE ADMITTANCE AND OSCILLATORY
C      PROFILES.
C      10 CALL CHAMDY
C      NOZA = NOZAMR*(1.-DCSDMR*MR(NXP)*2.*GAMD/(CS*(NXP)*(GAMD-1.)))
C      FN = (CNOZA - NOZA)
C      IF(IP.EQ.0 .AND.((ISCNT.EQ.1.OR.ISCNT.EQ.4).AND.ISLP.EQ.1))
C          1      GO TO 20
C      FNR = REAL(FN)
C      FNI = AIMAG(FN)
C      HN = CABS(FN)
C      XIMAGF = FNI
C      IF IP EQUALS ZERO AT THIS POINT , THEN DERIVATIVES WERE NOT
C      REQUIRED; HENCE, RETURN WITHOUT COMPUTING THEM.
C      IF(IP.EQ.0) RETURN
C      GO TO 30
C      20 IP = 1
C      RESTORE OMEGA AND SAVE VALUES FOR COMPUTATION OF DERIVATIVES.
C
C
C

```

```

00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660

```

F S C S M SUBROUTINES

```

      OMEGA = OMEGO
      HND = CABS(FN)
      GN = FN
      GO TO 10
C
C      COMPUTE D(CNUZA-NOZA)/D(IMAG(OMEGA)), CALLED GN, AND
C      D(CABS(CNOZA-NOZA)**2)/D(IMAG(OMEGA)), CALLED XIMAGF.
C
      30 GN = (GN-FN)/DM
      XIMAGF = (HND**2-HN**2)/DM
      RETURN
      END
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780

```



F S C S M SUBROUTINES

```

C |
C |
C |
C |
C |
SUBROUTINE ZADAMS(N,H,X,Y,DY,IQZ)
ROUTINE FOR SUBPROGRAM NOZADM
PROGRAM DEVELOPED BY GEORGIA INSTITUTE OF TECHNOLOGY
REF. NASA CR=121129
IMPLICIT REAL*(A-H,O-Z)
COMMON /X1/GAM,SVN,ANGLE,RCT,RCC /X2/T,R1,Q,R1,R2,WC,IP
      /X4/CM
DIMENSION COR(5), DP(5), DY(5,4), PRED(5), Y(5), G(5), GP(5)
DO 15 I=1,N
PREU(I)=Y(I)+H*(55.*DY(I,4)-59.*DY(I,3)+37.*DY(I,2)-9.*DY(I,1))
15 CONTINUE /24.C
X = X+H
U = PRED(1)
ZR = PRED(2)
ZI = PRED(3)
PHIR = PRED(4)
PHII = PRED(5)
C = 1-(GAM-1)*U*0.5
R = Q*((C)**(-1./(2.*(GAM-1))))*(U**(-0.25))*4.0
16 FORMAT(3X,'PRINTING FROM CARD 3430',/,3X,'R=',E15.8,
1 3X,'R1=',E15.8,3X,'RT=',E15.8)
IF(R-1) 17,17,1G0
17 IF(R-R1) 20,25,25
20 DR = -((2*RCT*(R-RT)-(R-RT)*(R-RT))*0.5)/(RT+RCT-R)
GO TO 40
25 IF(R-R2) 30,35,35
30 DK = -DTAN(T)
GO TO 40
35 DK = ((2*RCC*(1-R)-(1-R)*(1-R))*0.5)/(1-R-RCC)
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660

```

F S C S M SUBROUTINES

```

40 DU = -(U**0.75)*(C**((2*GAM-1)/(2*(GAM-1)))/(Q*(1-(GAM+1)*U*0.5)))
DP(1) = DR*DU
A = U*(C-U)
BR = U*DP(1)/C
BI = 2*WC*U
CR = WC*WC-(SVN*SVN*WC)/(R*R)
CI = -(GAM-1)*WC*U*DP(1)*0.5/C
DP(2) = ((BR*ZR-BI*ZI-CR)/A)-ZR*ZR+ZI*ZI
DP(3) = ((BI*ZR+BR*ZI-CI)/A)-2*ZR*ZI
DP(4) = ZR*PHIR-ZI*PHII
DP(5) = ZR*PHII+ZI*PHIR
DO 45 I=1,N
COR(I) = Y(I)+H*(DY(I,2)-5.*DY(I,3)+19.*DY(I,4)+9.*DP(I))/24.0
45 Y(I) = (251.*COR(I)+19.*PRED(I))/270.0
U = Y(1)
ZR = Y(2)
ZI = Y(3)
PHIR = Y(4)
PHII = Y(5)
C = 1-(GAM-1)*U*0.5
52 DO 55 I=1,N
DY(I,1) = DY(I,2)
DY(I,2) = DY(I,3)
DY(I,3) = DY(I,4)
ZMAG = (ZR*ZR+ZI*ZI)**0.5
IF(ZMAG-10) 60,90,90
60 R = Q*(C)**(-1.)/(2.*(GAM-1.))*(U**(-0.25))*4.0
61 FORMAT(3X,'PRINTING FROM CARD 3790',/,3X,'R=',E15.8,
1 3X,'R1=',E15.8,3X,'RT=',E15.8)
IF(R-1) 62,62,100
62 IF(N-R1) 65,70,70
65 DR = -((2*RCI*(R-RT)-(R-RT)*(R-RT))*U.5)/(RT+RCT-R)
GO TO 85

```

F S C S M SUBROUTINES

```

70 IF(R-R2) 75,8C,80
75 DR = -DTAN(T)
GO TO 85
80 DR = ((2*WCC*(1-R)-(1-R)*(1-R))*0.5)/(1-R-RCC)
85 DU = -(U*0.75)*(C**((2*GAM-1)/(2*(GAM-1)))/(Q*(1-(GAM+1)*U/2)))
DY(1,4) = DR*DU
A = U*(C-U)
BR = U*DY(1,4)/C
BI = 2*WC*U
CR = WC*WC-(SVN*SVN*WC)/(R*R)
CI = -(GAM-1)*WC*U*DY(1,4)*0.5/C
DY(2,4) = (BK*ZR-BI*ZI-CR)/A-ZR*ZR+ZI*ZI
DY(3,4) = (BI*ZR+BR*ZI-CI)/A-2.*ZR*ZI
DY(4,4) = ZR*PHIR-ZI*PHII
DY(5,4) = ZR*PHII+ZI*PHIR
IF(IP.EQ.0) GO TO 87
PR = WC*PHII-U*DY(4,4)
PI = -WC*PHIR-U*DY(5,4)
PMAG = (PR*PR+PI*PI)**0.5
PARG = DATAN(PI/PR)
WRITE(6,100C) X, PMAG, PARG
87 GO TO 10
90 IQZ = 1
Z2 = ZMAG*ZMAG
Y(2) = ZR/Z2
Y(3) = -ZI/Z2
ZPR = DY(2,4)
ZPI = DY(3,4)
DY(2,4) = -(ZPR*(ZR*ZR-ZI*ZI)+2.*ZK*ZI*ZPI)/(Z2*Z2)
DY(3,4) = (2.*ZPR*ZR*ZI-ZPI*(ZR*ZR-ZI*ZI))/(Z2*Z2)
G(1) = U
G(2) = Y(2)
G(3) = Y(3)
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300
00002310
00002320

```

F S C S M SUBROUTINES

```

G(4) = PHIR
G(5) = PHII
DY(1,1) = DY(1,4)
DY(2,1) = DY(2,4)
DY(3,1) = DY(3,4)
DY(4,1) = PHIR*ZR-PHII*ZI
DY(5,1) = PHII*ZR+PHIR*ZI
DO 95 I=2,4
CALL RKTZ(5,H,X,G,GP,IQZ)
X = X+H
U = G(1)
TR = G(2)
TI = G(3)
PHIR = G(4)
PHII = G(5)
DY(1,I) = GP(1)
DY(2,I) = GP(2)
DY(3,I) = GP(3)
DY(4,I) = GP(4)
DY(5,I) = GP(5)
95 Y(1) = U
Y(2) = TR
Y(3) = TI
Y(4) = PHIR
Y(5) = PHII
CALL TADAMS(N,H,X,Y,DY,IQZ,IQ)
GO TO (10,100),IQ
1000 FORMAT(4X,F6.4,1X,F10.5,3X,F10.5)
100 RETURN
END
00002230
00002340
00002350
00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620

```



F S C S M SUBROUTINES

```

A      ICNT = ICNT + 1
      IF ( ICNT .GT. NCNT) GO TO 500
      F3 = F(T3)
      T4OLD = T4
C      CONVERGENCE TEST FOR BISECTION GUESS.
C      IF(ABS((T1-T2)/T3).LE.EPSX.AND.ABS(F3)/AF.LE.EPSF) GO TO 300
      A = F1-2.*F3+F2
      B = F2-F1
      C = 2.*F3
C      TEST TO MAKE SURE MAXIMUM/MINIMUM OF PARABOLA FITTING T1,T2,T3
C      IS OUTSIDE THE INTERVAL (T1,T2). IF SO, UPDATE T USING PARABOLIC
C      INTERPOLATION.
      IF ( ABS( 2.*A/B) .GT. 0.9) GO TO 245
      B4AC = B**2-4.*A*C
      IF ( B4AC.LT. 0.) GO TO 600
      B4AC = SQRT(B4AC)
      IF ( ABS( A*C/B**2) .GT. EPS/100. ) GO TO 230
      T4 = T3 + (T3-T1)*(-C/B -A*C**2/B**3)
      GO TO 246
C      230 T4 =((-B+B4AC)/(2.*A))*(T3-T1) + T3
C      MAKE SURE T4 IS COMPATABLE WITH PREVIOUS VALUES; TEST FOR
C      CONVERGENCE AND THEN REPLACE T1,T2,F1, AND F2 APPROPRIATELY.
C      240 IF ( T4.LE.T3 .AND. T4.GE.T1 .AND. F3.GE. 0.) GO TO 250
C      IF ( T4.LE.T2 .AND. T4.GE.T3 .AND. F3.LE. 0.) GO TO 260
C      245 IF ( F3.LE. 0.) GO TO 270
C      T2 = T3
C      F2 = F3

```

```

01000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640

```

F S C S M SUBROUTINES

```

250 GO TO 220
    F4 = F(T4)
    IF (ABS((T4-T4OLD)/T4).LE.EPSX.AND.ABS(F4)/AF.LE.EPSF) GO TO 400
    IF ( F4.LT. 0.) GO TO 255
    T2 = T4
    F2 = F4
    GO TO 220
255 T1 = T4
    F1 = F4
    T2 = T3
    F2 = F3
    GO TO 220
260 F4 = F(T4)
    IF (ABS((T4-T4OLD)/T4).LE.EPSX.AND.ABS(F4)/AF.LE.EPSF) GO TO 400
    IF ( F4. GT. 0.) GO TO 265
    T1 = T4
    F1 = F4
    GO TO 220
265 T1 = T3
    F1 = F3
    T2 = T4
    F2 = F4
    GO TO 220
270 T1 = T3
    F1 = F3
    GO TO 220
300 T4 = T3
    F4 = F3
400 ANS = T4
    FANS = F4
    RETURN
500 IER = 1
    RETURN
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970

```

F S C S M SUBROUTINES

600 IER = 2  
RETURN  
END

00000960  
00000990  
00001000



APPENDIX D  
SAMPLE CASE INPUT



APPENDIX E

SAMPLE CASE OUTPUT

ANALYTICAL DESCRIPTION

FEED SYSTEM COUPLED

STABILITY MODEL

COMPUTER MODEL

PROGRAM NAME: FSCSM, FIV VERSION, MAY 1975

DEVELOPED BY: M. D. SCHUMAN, J. K. HUNTING, AND K. W. FERTIG  
ADVANCED PROGRAMS, ROCKEFORDYNE  
DIVISION OF ROCKWELL INTERNATIONAL  
CANOGA PARK, CALIF 91304

SPONSERED BY: NASA/LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS 77058  
UNDER CONTRACT NAS9-14315

FEED SYSTEM COUPLED STABILITY MODEL  
 OME FEED SYSTEM COUPLED STABILITY INVESTIGATION - MODEL VERIFICATION  
 6K OMS ENGINL TECHNOLOGY PROGRAM - TEST NUMBER 12 - CASE #1

INPHYD = 2 | INPCOM = 1 IMPNOZ = 3 ITAPH = 1 ITAPC = 2 ITAPN = 3  
 IPRHYD = 1 IPRCUM = 1 IPRNOZ = 1 IPRCHM = 1 IPRSTE = 1 NXP = 70  
 XO = 8.0000E-01 XNUZ = 7.7000E+00 RINJ = 4.1000E+00 GAMO = 1.2300E+00  
 CU = 3.9851E+03 DELP = 1.0000E-01  
 NRDOT = -1 IWRT = 0 IWSKP = 0 KNTMX = 50 KNTKMX = 100 KNTSMX = 20  
 OMEGA(R) = 1.9000E+02 OMEGA(I) = -1.0000E-01 FRQMAX = 4.0000E+02 DELFRQ = 5.0000E+00  
 DELMX = 5.0000E+01 CTFST = 7.5000E+01  
 EPSF = 1.0000E-02 EPSX = 1.0000E-02 EPSFS = 5.0000E-04 EPSXS = 5.0000E-04  
 PC = 7.1150E+01 MBOXI = 6.5730E+00 MBFUI = 3.4030E+00  
 NFREQT = 20 FREQMI = 1.5000E+02 FREOMX = 4.0000E+02  
 RCCX = 4.1100E+00 RCTX = 9.9300E-01 ANGLEX = 1.9550E+01 CRR = 1.9850E+00

THEORETICAL NOZZLE ADMITTANCES

MACH NUMBER = .31 SVN = 0.0 GAMMA = 1.250

NOZZLE ANGLE = 19.6 RADIUS OF CURVATURE: THROAT = 0.9930 ENTRANCE = 4.1100

FC	YR	YI	F	SYR	SVI	ALPHA	BETA
150.0	0.03305	0.09816	150.85	0.03849	0.11432	0.00190	0.49704
160.0	0.03334	0.10477	160.90	0.03883	0.12202	0.00107	0.49663
170.0	0.03365	0.11139	170.96	0.03919	0.12972	0.00115	0.49620
180.0	0.03398	0.11802	181.02	0.03957	0.13744	0.00123	0.49573
190.0	0.03432	0.12466	191.07	0.03997	0.14518	0.00131	0.49524
200.0	0.03469	0.13132	201.13	0.04040	0.15293	0.00139	0.49473
210.0	0.03508	0.13799	211.19	0.04085	0.16070	0.00148	0.49418
220.0	0.03549	0.14468	221.24	0.04133	0.16849	0.00157	0.49361
230.0	0.03592	0.15136	231.30	0.04183	0.17630	0.00166	0.49301
240.0	0.03637	0.15810	241.36	0.04235	0.18413	0.00175	0.49238
250.0	0.03684	0.16484	251.41	0.04290	0.19198	0.00185	0.49173
260.0	0.03733	0.17160	261.47	0.04348	0.19985	0.00195	0.49104
270.0	0.03785	0.17838	271.52	0.04408	0.20774	0.00205	0.49033
280.0	0.03839	0.18518	281.56	0.04471	0.21560	0.00216	0.48959
290.0	0.03895	0.19200	291.64	0.04537	0.22360	0.00227	0.48882
300.0	0.03954	0.19884	301.69	0.04605	0.23157	0.00238	0.48803
310.0	0.04015	0.20571	311.75	0.04676	0.23957	0.00250	0.48720
320.0	0.04078	0.21260	321.81	0.04750	0.24759	0.00262	0.48635
330.0	0.04144	0.21951	331.86	0.04827	0.25565	0.00274	0.48540
340.0	0.04213	0.22645	341.92	0.04906	0.26373	0.00287	0.48451
350.0	0.04284	0.23342	351.98	0.04989	0.27184	0.00300	0.48361
360.0	0.04357	0.24041	362.03	0.05075	0.27999	0.00314	0.48263
370.0	0.04434	0.24743	372.09	0.05163	0.28816	0.00328	0.48163
380.0	0.04513	0.25448	382.15	0.05255	0.29638	0.00343	0.48060
390.0	0.04594	0.26156	392.20	0.05351	0.30462	0.00358	0.47954
400.0	0.04679	0.26868	402.26	0.05449	0.31291	0.00374	0.47845







FEED SYSTEM RESPONSE PARAMETERS

FREQUENCY	OXIDIZER INJECTION RATE		FUEL INJECTION RATE	
	AMPLITUDE	PHASE	AMPLITUDE	PHASE
150.847	1.8274E+00	165.42	3.7451E-01	262.56
160.904	5.3892E-01	167.51	4.5190E-01	260.85
170.960	2.4679E-01	98.76	5.5606E-01	258.46
181.017	1.2781E-01	95.61	7.1297E-01	254.68
191.073	5.8226E-02	94.64	9.9354E-01	247.46
201.130	8.7560E-03	104.80	1.6312E+00	228.22
211.186	3.2075E-02	266.27	2.1617E+00	164.52
221.243	6.8631E-02	267.14	9.3812E-01	118.97
231.299	1.0466E-01	266.67	4.1875E-01	106.63
241.356	1.4327E-01	265.84	1.7884E-01	103.57
251.412	1.8866E-01	264.69	3.8063E-02	119.61
261.469	2.4836E-01	263.06	6.8477E-02	259.11
271.525	3.4028E-01	260.41	1.4746E-01	264.44
281.582	5.2332E-01	254.86	2.1603E-01	264.79
291.638	1.1343E+00	234.15	2.7931E-01	264.28
301.694	1.1594E+00	129.46	3.4106E-01	265.46
311.751	3.6691E-01	163.02	4.0468E-01	262.44
321.807	1.5683E-01	97.16	4.7377E-01	261.22
331.864	5.6576E-02	95.90	5.5330E-01	259.71
341.920	7.6234E-03	250.46	6.5131E-01	257.73
351.977	5.5028E-02	267.03	7.8301E-01	254.93
362.033	9.5480E-02	267.03	9.8190E-01	250.42
372.090	1.3307E-01	266.40	1.3357E+00	241.63
382.146	1.7096E-01	265.55	2.0667E+00	218.24
392.203	2.1235E-01	264.52	2.1736E+00	157.47
402.259	2.6173E-01	263.20	9.5958E-01	120.13

FEED SYSTEM COUPLED STABILITY MODEL  
 ONE FEED SYSTEM COUPLED STABILITY INVESTIGATION - MODEL VERIFICATION  
 6K OMS ENGINE TECHNOLOGY PROGRAM - TEST NUMBER 12 - CASE #1

XKDX = 2.0040E+00	TAUBUX = 6.6900E-03	V6UX = 2.4849E+01	DELHUX = 9.5900E+03
	TURAGO = 0.0	ADVUX = -7.5000E-01	
ADDOX = 5.7000E-01	DELVOX = 1.0000E-02	NUB0X = 3.2300E+00	DT0XDM = 0.0
	XIMPOX = 1.8800E-01		
XKFU = 2.0000E+00	TAUBFU = 4.6300E-03	V6FU = 3.5139E+01	DELHFU = -1.0750E+03
	TURAGF = 0.0	ADVFU = -7.5000E-01	
ADDFU = 5.7000E-01	DELVFU = 1.0000E-02	NUBFU = 3.6720E+00	DTFU0DM = 0.0
	XIMPFU = 1.8000E-01		
MWG = 2.1628E+01	CS = 5.6671E+03	DRGDMR = -1.5200E+01	UCSDMR = 0.0
	DHDMR = 1.6517E+03		

STEADY STATE SOLUTION

DISTANCE (INCHES)	TEMPERATURE (RANKINE)	VELOCITY (FT/S)	MIXTURE RATIO	DENSITY (LBM/FT*3)	PERCENT FUEL	PERCENT VAPOORIZED OXIDIZER
0.8500	5435.33	24.89	1.6309	2.4796E-02	2.53	2.14
0.9500	5440.96	72.97	1.6374	2.48129E-02	7.40	6.27
1.0500	5446.45	118.85	1.6438	2.48097E-02	12.02	10.23
1.1500	5451.81	162.63	1.6501	2.48168E-02	16.41	14.02
1.2500	5457.01	204.41	1.6564	2.48242E-02	20.59	17.65
1.3500	5462.08	244.27	1.6625	2.48319E-02	24.55	21.13
1.4500	5467.02	282.30	1.6686	2.48399E-02	28.32	24.46
1.5500	5471.80	318.59	1.6746	2.48482E-02	31.90	27.65
1.6500	5476.47	353.21	1.6805	2.48566E-02	35.30	30.71
1.7500	5481.00	386.24	1.6864	2.48653E-02	38.53	33.64
1.8500	5485.42	417.75	1.6921	2.48742E-02	41.60	36.44
1.9500	5489.71	447.82	1.6978	2.48832E-02	44.51	39.13
2.0500	5493.88	476.49	1.7033	2.48925E-02	47.28	41.70
2.1500	5497.92	503.85	1.7088	2.49019E-02	49.92	44.16
2.2500	5501.86	529.94	1.7142	2.49114E-02	52.42	46.52
2.3500	5505.66	554.83	1.7195	2.49211E-02	54.79	48.78
2.4500	5509.37	578.57	1.7248	2.49308E-02	57.05	50.94
2.5500	5512.97	601.21	1.7299	2.49407E-02	59.20	53.02
2.6500	5516.45	622.81	1.7349	2.49506E-02	61.23	55.00
2.7500	5519.84	643.41	1.7399	2.49606E-02	63.17	56.90
2.8500	5523.12	663.05	1.7448	2.49707E-02	65.01	58.72
2.9500	5526.31	681.78	1.7496	2.49807E-02	66.75	60.47
3.0500	5529.40	699.65	1.7543	2.49909E-02	68.41	62.14
3.1500	5532.39	716.68	1.7589	2.50010E-02	69.99	63.74
3.2500	5535.29	732.93	1.7634	2.50111E-02	71.49	65.27
3.3500	5538.11	748.43	1.7678	2.50212E-02	72.91	66.73
3.4500	5540.83	763.20	1.7722	2.50313E-02	74.27	68.14
3.5500	5543.47	777.29	1.7765	2.50414E-02	75.55	69.49
3.6500	5546.02	790.72	1.7807	2.50514E-02	76.77	70.77
3.7500	5548.49	803.53	1.7848	2.50615E-02	77.93	72.01
3.8500	5550.89	815.74	1.7888	2.50714E-02	79.03	73.19
3.9500	5553.21	827.38	1.7927	2.50813E-02	80.08	74.32
4.0500	5555.45	838.49	1.7965	2.50911E-02	81.07	75.41
4.1500	5557.62	849.07	1.8003	2.51009E-02	82.02	76.45
4.2500	5559.73	859.16	1.8040	2.51105E-02	82.92	77.44
4.3500	5561.77	868.79	1.8076	2.51201E-02	83.77	78.40

4.4500	5563.73	877.96	1.8111	2.51295E-02	84.58	74.31
4.5500	5565.64	886.71	1.6146	2.51389E-02	85.25	80.18
4.6500	5567.48	895.05	1.6179	2.51482E-02	86.06	81.02
4.7500	5569.26	903.00	1.5212	2.51573E-02	86.78	81.82
4.8500	5570.99	910.58	1.8244	2.51663E-02	87.44	82.59
4.9500	5572.65	917.80	1.8276	2.51752E-02	88.06	83.33
5.0500	5574.26	924.69	1.8306	2.51840E-02	8 .66	84.03
5.1500	5575.82	931.26	1.8336	2.51927E-02	89.23	84.70
5.2500	5577.32	937.52	1.8365	2.52012E-02	89.76	85.35
5.3500	5578.77	943.49	1.8394	2.52096E-02	90.28	85.97
5.4500	5580.18	949.18	1.8422	2.52178E-02	90.76	86.56
5.5500	5581.54	954.61	1.8449	2.52259E-02	91.22	87.13
5.6500	5582.85	959.78	1.8475	2.52339E-02	91.66	87.67
5.7500	5584.12	964.71	1.8501	2.52417E-02	92.08	88.19
5.8500	5585.34	969.41	1.8526	2.52494E-02	92.47	88.69
5.9500	5586.52	973.89	1.8550	2.52569E-02	92.85	89.17
6.0500	5587.66	978.16	1.8574	2.52643E-02	93.21	89.63
6.1500	5588.77	982.23	1.8597	2.52716E-02	93.55	90.07
6.2500	5589.84	986.12	1.8619	2.52787E-02	93.87	90.49
6.3500	5590.86	989.82	1.8641	2.52856E-02	94.17	90.89
6.4500	5591.86	993.35	1.8663	2.52924E-02	94.46	91.27
6.5500	5592.81	996.71	1.8683	2.52991E-02	94.74	91.64
6.6500	5593.75	999.92	1.8704	2.53056E-02	95.00	91.99
6.7500	5594.64	1002.98	1.8723	2.53120E-02	95.25	92.33
6.8500	5595.50	1005.89	1.8742	2.53182E-02	95.49	92.66
6.9500	5596.34	1008.67	1.8761	2.53243E-02	95.72	92.97
7.0500	5597.14	1011.32	1.8779	2.53302E-02	95.93	93.26
7.1500	5597.91	1013.85	1.8796	2.53360E-02	96.13	93.54
7.2500	5598.67	1016.26	1.8813	2.53417E-02	96.33	93.82
7.3500	5599.38	1018.55	1.8830	2.53472E-02	96.51	94.08
7.4500	5600.09	1020.74	1.8846	2.53525E-02	96.68	94.33
7.5500	5600.77	1022.63	1.8861	2.53578E-02	96.85	94.57
7.6500	5601.42	1024.62	1.8876	2.53629E-02	97.01	94.80
7.7500	5602.04	1026.72	1.8891	2.53679E-02	97.16	95.02

FEED SYSTEM COUPLED STABILITY MODEL  
 ONE FEED SYSTEM COUPLED STABILITY INVESTIGATION - MODEL VERIFICATION  
 GK OMS ENGINE TECHNOLOGY PROGRAM - TEST NUMBER 12 - CASE #1

COMBUSTION DYNAMIC COEFFICIENTS

COX( 1) = 2.0719E-01:	5.4497E-02	CFU( 1) = -5.2935E+00:	1.1404E+01
COX( 2) = 0.0	: 0.0	CFU( 2) = 0.0	: 0.0
COX( 3) = 0.0	: 0.0	CFU( 3) = 0.0	: 0.0
COX( 4) = 1.9040E-01:	0.0	CFU( 4) = 2.2767E-01:	0.0
COX( 5) = 0.0	: 0.0	CFU( 5) = 0.0	: 0.0
COX( 6) = 0.0	: 0.0	CFU( 6) = 0.0	: 0.0
COX( 7) = 0.0	: 0.0	CFU( 7) = 0.0	: 0.0
COX( 8) = 0.0	: 0.0	CFU( 8) = 0.0	: 0.0
COX( 9) = 1.1451E-02:	-9.5491E-03	CFU( 9) = 5.2994E-01:	7.4232E-01
COX(10) = 0.0	: 0.0	CFU(10) = 0.0	: 0.0
COX(11) = 0.0	: 0.0	CFU(11) = 0.0	: 0.0
COX(12) = -1.9040E-01:	0.0	CFU(12) = -2.2767E-01:	0.0
COX(13) = 0.0	: 0.0	CFU(13) = 0.0	: 0.0
COX(14) = 0.0	: 0.0	CFU(14) = 0.0	: 0.0
COX(15) = 0.0	: 0.0	CFU(15) = 0.0	: 0.0
COX(16) = 0.0	: 0.0	CFU(16) = 0.0	: 0.0

FREQUENCY = 210.42 HZ,  
 DECUREMENT = -0.06144

NOZZLE ADMITTANCE = 0.04082: 0.16011

FEED SYSTEM RESPONSE  
 OXIDIZER = -0.00210: 0.02891  
 FUEL = -2.00708: -0.43958

OME FEED SYSTEM COUPLED STABILITY INVESTIGATION -- MODEL VERIFICATION  
 6K OMS ENGINE TECHNOLOGY PROGRAM -- TEST NUMBER 1L -- CASE #1

DISTANCE (INCHES)	OSCILLATORY PRESSURE		OSCILLATORY VELOCITY		OSCILLATORY TEMPERATURE		OSCILLATORY MIXTURE	
	AMPLITUDE	RATIO PHASE	AMPLITUDE	RATIO PHASE	AMPLITUDE	RATIO PHASE	AMPLITUDE	RATIO PHASE
0.8000	0.09998	-0.00	0.00180	93.52	0.33561	88.64	1.67995	91.83
0.9000	0.10000	0.03	0.00409	-111.60	0.26984	102.58	1.64734	101.09
1.0000	0.10003	0.14	0.00896	-97.10	0.26425	111.05	1.04569	110.16
1.1000	0.10001	0.31	0.01314	-86.70	0.25801	119.75	1.68232	118.97
1.2000	0.09989	0.51	0.01656	-77.12	0.25036	128.35	1.65705	127.55
1.3000	0.09960	0.70	0.01920	-67.83	0.24129	136.75	1.62026	135.88
1.4000	0.09934	0.86	0.02109	-58.69	0.23094	144.94	1.57259	143.96
1.5000	0.09896	0.95	0.02224	-49.62	0.21946	152.89	1.51489	151.78
1.6000	0.09857	0.96	0.02271	-40.60	0.20705	160.56	1.44816	159.30
1.7000	0.09822	0.90	0.02256	-31.61	0.19390	167.93	1.37360	166.51
1.8000	0.09794	0.76	0.02186	-22.65	0.18022	174.90	1.29252	173.35
1.9000	0.09778	0.58	0.02068	-13.72	0.16622	-178.42	1.20640	174.79
2.0000	0.09775	0.38	0.01912	-4.83	0.15213	-172.20	1.11623	-174.25
2.1000	0.09786	0.18	0.01726	4.01	0.13816	-166.52	1.02552	-168.84
2.2000	0.09808	0.01	0.01518	12.77	0.12455	-161.46	0.93433	-164.09
2.3000	0.09840	-0.11	0.01296	21.39	0.11152	-157.13	0.84523	-160.12
2.4000	0.09877	-0.17	0.01068	29.77	0.09831	-153.07	0.76038	-157.08
2.5000	0.09915	-0.16	0.00840	37.69	0.08815	-151.23	0.66207	-155.13
2.6000	0.09951	-0.09	0.00619	44.69	0.07830	-149.95	0.61271	-154.42
2.7000	0.09980	0.04	0.00411	49.39	0.06998	-149.91	0.55468	-155.01
2.8000	0.09995	0.20	0.00224	46.48	0.06338	-151.08	0.50997	-156.80
2.9000	0.10009	0.38	0.00103	3.69	0.05859	-153.13	0.47968	-159.47
3.0000	0.10008	0.55	0.00175	-52.00	0.05555	-155.74	0.46345	-162.47
3.1000	0.09997	0.71	0.00302	-57.96	0.05404	-158.21	0.45933	-165.19
3.2000	0.09978	0.82	0.00419	-53.62	0.05368	-160.10	0.46422	-167.19
3.3000	0.09954	0.89	0.00519	-40.37	0.05407	-161.13	0.47469	-168.25
3.4000	0.09927	0.90	0.00601	-37.94	0.05465	-161.24	0.48760	-168.38
3.5000	0.09900	0.86	0.00666	-28.93	0.05569	-160.54	0.50047	-167.70
3.6000	0.09876	0.77	0.00716	-19.63	0.05641	-159.16	0.51153	-166.36
3.7000	0.09858	0.64	0.00752	-10.23	0.05685	-157.28	0.51965	-164.61
3.8000	0.09846	0.49	0.00776	-0.85	0.05694	-155.05	0.52420	-162.55
3.9000	0.09842	0.32	0.00788	8.39	0.05666	-152.63	0.52498	-160.34
4.0000	0.09845	0.15	0.00791	17.37	0.05602	-150.13	0.52110	-156.09
4.1000	0.09854	0.00	0.00785	25.98	0.05505	-147.65	0.51592	-155.92

4.2000	0.09808	-0.13	0.00771	34.00	0.05381	-145.29	0.50699	-15.91
4.3000	0.09864	-0.22	0.00750	41.60	0.05236	-143.12	0.49598	-152.13
4.4000	0.09902	-0.26	0.00723	48.34	0.05079	-141.20	0.48367	-150.64
4.5000	0.09919	-0.32	0.00691	54.16	0.04917	-139.57	0.47086	-149.47
4.6000	0.09934	-0.32	0.00657	58.89	0.04758	-138.25	0.45834	-148.64
4.7000	0.09945	-0.31	0.00622	62.34	0.04608	-137.24	0.44683	-148.14
4.8000	0.09952	-0.28	0.00590	64.36	0.04473	-136.51	0.43694	-147.92
4.9000	0.09954	-0.24	0.00563	64.86	0.04358	-136.02	0.42907	-147.91
5.0000	0.09951	-0.21	0.00545	63.97	0.04266	-135.70	0.42345	-148.04
5.1000	0.09945	-0.20	0.00540	62.03	0.04196	-135.46	0.42005	-148.20
5.2000	0.09936	-0.19	0.00550	59.64	0.04148	-135.23	0.41866	-148.31
5.3000	0.09925	-0.21	0.00574	57.43	0.04118	-134.93	0.41894	-148.30
5.4000	0.09913	-0.24	0.00611	55.87	0.04102	-134.51	0.42041	-148.13
5.5000	0.09901	-0.30	0.00657	55.18	0.04097	-133.94	0.42259	-147.77
5.6000	0.09891	-0.37	0.00708	55.33	0.04049	-133.20	0.42502	-147.22
5.7000	0.09882	-0.45	0.00761	56.20	0.04102	-132.30	0.42728	-146.51
5.8000	0.09875	-0.55	0.00814	57.62	0.04164	-131.27	0.42906	-145.66
5.9000	0.09871	-0.64	0.00864	59.40	0.04102	-130.12	0.43014	-144.71
6.0000	0.09869	-0.74	0.00911	61.40	0.04096	-128.91	0.43039	-143.71
6.1000	0.09870	-0.83	0.00951	63.49	0.04083	-127.65	0.42979	-142.68
6.2000	0.09871	-0.91	0.00987	65.54	0.04065	-126.39	0.42838	-141.66
6.3000	0.09874	-0.98	0.01016	67.49	0.04041	-125.15	0.42628	-140.69
6.4000	0.09876	-1.04	0.01041	69.26	0.04014	-123.96	0.42365	-139.79
6.5000	0.09879	-1.09	0.01061	70.79	0.03983	-122.83	0.42070	-138.97
6.6000	0.09880	-1.13	0.01077	72.05	0.03951	-121.79	0.41761	-138.24
6.7000	0.09880	-1.16	0.01091	73.03	0.03919	-120.83	0.41460	-137.61
6.8000	0.09879	-1.19	0.01104	73.72	0.03889	-119.94	0.41183	-137.05
6.9000	0.09877	-1.22	0.01118	74.15	0.03863	-119.13	0.40944	-136.55
7.0000	0.09873	-1.25	0.01133	74.36	0.03840	-118.38	0.40751	-136.11
7.1000	0.09868	-1.28	0.01150	74.39	0.03822	-117.68	0.40609	-135.69
7.2000	0.09862	-1.32	0.01171	74.31	0.03808	-116.99	0.40516	-135.27
7.3000	0.09855	-1.36	0.01195	74.19	0.03799	-116.32	0.40469	-134.84
7.4000	0.09848	-1.41	0.01223	74.07	0.03795	-115.64	0.40458	-134.38
7.5000	0.09840	-1.46	0.01253	74.00	0.03794	-114.95	0.40474	-133.89
7.6000	0.09834	-1.52	0.01286	74.01	0.03795	-114.22	0.40507	-133.34
7.7000	0.09827	-1.58	0.01320	74.12	0.03799	-113.47	0.40545	-132.76

ANALYTICAL DESCRIPTION

FEED SYSTEM COUPLED

STABILITY MODEL

COMPUTER MODEL

PROGRAM NAME: FSCSM, FIV VERSION, MAY 1975

DEVELOPED BY: M. D. SCHUMAN, J. K. HUNTING, AND K. W. FERTIG  
ADVANCED PROGRAMS, ROCKETDYNE  
DIVISION OF ROCKWELL INTERNATIONAL  
CANOGA PARK, CALIF 91304

SPONSERED BY: NASA/LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS 77058  
UNDEK CONTRACT NAS9-14315



FEED SYSTEM COUPLED STABILITY MODEL  
 OME FEED SYSTEM COUPLED STABILITY INVESTIGATION - MODEL VERIFICATION  
 6K OMS ENGINE TECHNOLOGY PROGRAM - TEST NUMBER 12 - CASE #2

INPHYD = 3 | INPCOM = 1 | IMPNOZ = 4 | ITAPH = 1 | ITAPC = 2 | ITAPN = 3  
 IPRHYD = 0 | IPRCOM = 0 | IPRNOZ = 0 | IPRCHM = 0 | IPRSTE = 0 | NXP = 70  
  
 XU = 8.0000E-01 | XNOZ = 7.7000E+00 | RINJ = 4.1000E+00 | GAMO = 1.2300E+00  
 CO = 3.9851E+03 | DELP = 1.0000E-01  
  
 NROOT = -1 | IMRT = 0 | IMSKP = 0 | KNTMX = 50 | KNTRMX = 100 | KNTSMX = 20  
  
 OMEGA(R) = 2.6500E+02 | OMEGA(I) = -1.0000E-01 | FRQMAX = 4.0000E+02 | DELFRQ = 5.0000E+00  
 DELMX = 5.0000E+01 | CTEST = 7.5000E+01  
  
 EPSF = 1.0000E-02 | EPSX = 1.0000E-02 | EPSFS = 5.0000E-04 | EPSXS = 5.0000E-04  
  
 PC = 7.1150E+01 | MLOXI = 6.5730E+00 | MBFUI = 3.4035E+00  
  
 XKOX = 2.0000E+00 | TAUBOX = 6.6900E-03 | VBOX = 2.8859E+01 | DELMOX = 9.5908E+03  
 TDRAGO = 0.0 | ADVOX = -7.5000E-01  
  
 ADDOX = 5.7000E-01 | DELVOX = 1.0000E-02 | NUBOX = 3.2300E+00 | OTOXDM = 0.0  
 XIMPOX = 1.8800E-01  
  
 XKFU = 2.0000E+00 | TAUBFU = 4.6300E-03 | VBFU = 3.5139E+01 | DELHFU = -1.0790E+03  
 TDRAGF = 0.0 | ADVFU = -7.5000E-01  
  
 ADDFU = 5.7000E-01 | DELVFU = 1.0000E-02 | NUBFU = 3.6720E+00 | DTFU DM = 0.0  
 XIMPFU = 1.8800E-01  
  
 MWG = 2.1628E+01 | CS = 5.6671E+03 | DRGDMR = -1.5200E+01 | DCSDMR = 0.0  
 DHDMR = 1.6517E+03

FEEED SYSTEM COUPLED STABILITY MODEL  
OME FEEED SYSTEM COUPLED STABILITY INVESTIGATION - MODEL VERIFICATION  
6K OMS ENGINE TECHNOLOGY PROGRAM - TEST NUMBER 12 - CASE #2

FREQUENCY = 250.64 HZ,  
DECREMENT = 0.09211

NOZZLE ADMITTANCE = 0.04465: 0.21491

FEEED SYSTEM RESPONSE

OXIDIZER = -0.12902: 0.48896  
FUEL = -0.01911: 0.20861

ANALYTICAL DESCRIPTION

FEED SYSTEM COUPLED

STABILITY MODEL

COMPUTER MODEL

PROGRAM NAME: FSCSM, FIV VERSION, MAY 1975

DEVELOPED BY: M. D. SCHUMAN, J. K. HUNTING, AND K. W. FERTIG  
ADVANCED PROGRAMS, ROCKETDYNE  
DIVISION OF ROCKWELL INTERNATIONAL  
CANOGA PARK, CALIF 91304

SPONSERED BY: NASA/LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS 77058  
UNDER CONTRACT NAS9-14315

