CR-128748

CHARACTERIZATION OF THE SPACE SHUTTLE REACTION CONTROL SYSTEM ENGINE

by

M.S. Wilson, R.C. Stechman, R.B. Edelman,
O.F. Fortune, and C. Economos

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

May 1, 1972

Contract NAS 9-11740

NASA Manned Spacecraft Center
Houston, Texas

THE MARQUARDT COMPANY
16555 Saticoy Street
Van Nuys, California 91409

23

TMC S-1220

FINAL REPORT

CHARACTERIZATION OF THE SPACE SHUTTLE REACTION CONTROL SYSTEM ENGINE

by

M. S. Wilson, R. C. Stechman, R. B. Edelman,
O. F. Fortune, and C. Economos

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

May 1, 1972

Contract NAS 9-11740

NASA Manned Spacecraft Center
Houston, Texas
D. Hyatt, Project Manager

THE MARQUARDT COMPANY
16555 Saticoy Street
Van Nuys, California 91409

ABSTRACT

A computer program was developed and written in Fortran V which predicts the transient and steady state performance and heat transfer characteristics of a pulsing GO_2/GH_2 rocket engine. This program predicts the dynamic flow and ignition characteristics which, when combined in a quasi-steady state manner with the combustion and mixing analysis program, will provide the thrust and specific impulse of the engine as a function of time. The program also predicts the transient and steady state heat transfer characteristics of the engine using various cooling concepts. The computer program, test case, and documentation are presented in this document. The program is applicable to any system capable of utilizing the Fortran IV or Fortran V language.

FOREWORD

Contract NAS 9-11740, "Characterization of the Space Shuttle Reaction Control System Engine" was performed by The Marquardt Company at Van Nuys, California, and by the principal Subcontractor, The General Applied Science Laboratory (GASL). The period of performance covered by this report is from May 15, 1971, to April 15, 1972. The Program Manager was R. C. Stechman. Principal Investigators included M. S. Wilson, R. B. Edelman, and O. Fortune. The NASA Project Manager was D. Hyatt, NASA-MSC.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHARACTERIZATION OF THE SPACE SHUTTLE REACTION CONTROL SYSTEM ENGINE

By: M. S. Wilson, R. C. Stechman, R. B. Edelman,
O. F. Fortune, and C. Economos

The Marquardt Company
Van Nuys, California

SUMMARY

Under Contract NAS 9-11740, a computer model has been developed to characterize the combustion, ignition, and dynamic performance of GO_2/GH_2 rocket engines. A dynamics module, incorporating valve opening characteristics and ignition system performance, is used to predict pressures, temperatures, and mixture ratios in the various engine cavities during pulse operation. Instantaneous performance may be calculated at intervals utilizing a combustion model to determine the degree of combustion gas mixing and a performance model to calculate friction losses, divergence losses, etc. The combustion model may be used independently to predict the chemical and gas dynamic behavior of the reacting O_2/H_2 gas mixture considering either finite or infinite chemical reaction rates and including the effects of mass addition from film, transpiration, or dump cooling. The characterization program also includes a heat transfer module designed to predict operating temperatures for GO_2/GH_2 rocket engines using radiation, heat sink, film, regenerative, or dump (liner) cooling schemes.

Analytical results from the characterization models have been found to compare favorably with test results obtained with GO_2/GH_2 engines. Comparisons were made with data on dynamic response, specific impulse, and structural temperatures.

INTRODUCTION

Development of analytical techniques and computer models which can characterize various parts of the space shuttle is required in order that the overall system can be analyzed without extensive testing of each proposed design. The models developed can be used to determine system design criteria, influence coefficients, and ultimately to characterize the final system design. An important aspect of the shuttle system is a model of the reaction control system engine. Included in this model is a requirement for a more thorough understanding of the combustion and ignition process and their coupling with the dynamic and heat transfer characteristics of the engine. The successful incorporation of the combustion and mixing model with other parametric models provides a capability to determine the reaction control system performance for a given point design as well as a fixed set of environmental conditions.

This report presents the results of a 10-month study for the National Aeronautics and Space Administration/Manned Spacecraft Center to develop an analytical model and computer program for the prediction of the transient and steady state performance characteristics of reaction control rocket engines which use gaseous oxygen and gaseous hydrogen. Major emphasis has been placed on the development of computer codes to describe the mixing and reaction processes in the combustion chamber. Less emphasis has been placed on the losses and processes in the exit nozzle since these processes have been previously investigated in great detail (Reference 1). The results generated by the program when compared to the results of the GO_2/GH_2 studies now being conducted will provide the necessary data for meaningful correlation.

This report is divided into two discrete parts. The first part describes the methodology used to provide the input for the computer programming. The six parts of the model, dynamics, injection, ignition, combustion, performance, and heat transfer are described and the equations are presented in detail. The results of a comparison of a single test case with the data obtained from a contractor's report (Reference 2) are described.

The second portion of the report contains a detailed description of the computer program, a listing of the program, a description of input requirements, and flow charts which will enable the reader to use the program. The description and results of the test case are also provided.

ANALYSIS

Model Description

The performance of reaction control engines is measured by the instantaneous thrust that the engine generates and the amount of propellant used in producing the thrust. The performance of GO_2/GH_2 engines is basically a function of the parameters listed in Table I. These parameters can, when described adequately, predict the specific impulse and heat transfer characteristics of an engine in both a pulsing cycle and at steady state. As shown, the parameters which affect specific impulse are more numerous than those that affect thrust and mixture ratio. Thrust variations are only a function of system changes and can be varied and thus are not a true evaluator of efficiency of the engine compared to specific impulse. Mixture ratio variations are also caused principally by system variation and time (gas dynamics). Therefore, in the performance analysis of an engine, specific impulse efficiency is the true measure of performance while thrust and mixture ratio characterize the engine due to system changes and time. Heat transfer characteristics, in terms of temperature variation, will influence performance in a minor way.

Based on the above premise, a fully integrated pulsing performance model, based on variations of the system shown in Figure 1, was developed for gaseous oxygen/gaseous hydrogen rocket engines. The model predicts instantaneous thrust and integrated impulse and Isp for specified pulsing conditions taking account for injector mixing, turbulent diffusion, finite chemical kinetics, divergence losses, friction losses, and heat loss.

The basic driver for the program is an engine dynamics model which calculates pressure, mixture ratio, and temperature history for the engine main combustor, pilot combustor, manifolds, etc. The dynamics subprogram incorporates an integral spark ignitor model which tests for sparking, ignition, and flame quenching. Valve modeling allows for sequencing and finite opening and closing times.

A propellant injection model provides a starting flow of specified mixing efficiency to a combustion model developed by General Applied Science Laboratory. The injection and combustion models are called a number of times during the chamber pressurization process. The combustion model predicts propellant mixing and chemistry from the injector face down to the throat. At the user's option, finite chemical kinetics are considered and the effects of film or transpiration cooling mass addition or regenerative cooling are accounted for.

The performance model uses throat enthalpy profiles to predict quasi-steady state performance. The effects of nozzle divergence, boundary layer losses, and mass addition in the bell are included to obtain estimates of actual performance. Before ignition has occurred, a cold flow performance model is used to estimate thrust and Isp. The final performance calculations return an updated thrust coefficient value to the dynamics model where accumulated impulse and Isp are determined at each iteration.

TABLE I. INFLUENCE OF SYSTEM OR DESIGN VARIABLE ON PERFORMANCE PARAMETER

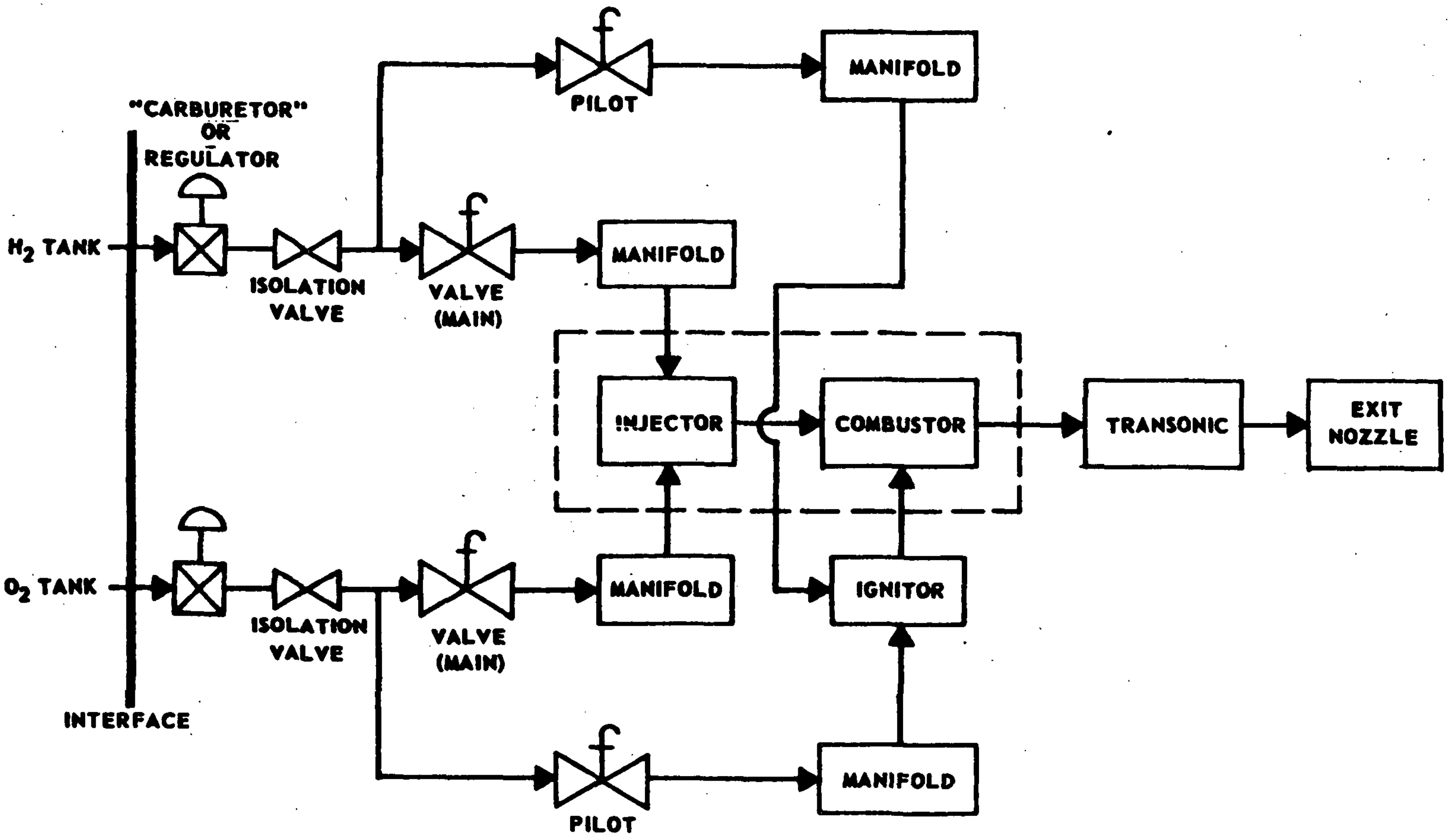
	Performance Parameter						
	I_{sp}		F		O/F		
System or Design Variable	Pulse	SS	Pulse	SS	Pulse	SS	General Comments
Upstream hydraulics or gas dynamics	/	-	/	-	X	-	Compressible flow and wave dynamics cause variations in mixture ratio.
Injector hydraulics or gas dynamics	/	-	/	-	/	-	Injector dynamics during transient operation can cause variations in O/F, flow-which causes variations in F and I_{sp}.
O/F	-	/	-	/	-	-	Minor variations due to O/F shift.
O/F distribution	/	X	/	/	-	-	Striations across injector face can result in wide O/F variation and deviation from maximum I_{sp} point.
Injector orifice size	/	X	-	-	-	-	Effects mixing.
Injector element	/	X	-	-	-	-	Effects mixing.
Valve response and lead/lag	X	-	X	-	X	-	O/F variations on startup can cause ignition delays. Tailoff impulse variation at shutdown.
Ignition timing	X	-	X	-	/	-	Causes wasted propellant if timing not optimized.
Turbulence	X	X	-	-	-	-	Key factors in performance efficiency.
Mixing	X	X	-	-	-	-	
Kinetics	-	X	-	-	-	-	Important at low pressures.
Boundary effects	-	X	-	/	-	-	Important for small nozzles, secondary for large engines.
Nozzle type	-	/	-	/	-	-	Secondary effect.
Cooling technique	/	X	-	-	-	/	Film cooling can result in significant efficiency loss.
Propellant inlet temperature.	-	X	-	X	-	X	Causes shift in O/F, change due to added or subtracted enthalpy and change in density.
Propellant pressure	-	-	-	X	-	X	Causes shift in mixture ratio and change in thrust due to changes in density

X	Important
/	Secondary
-	No effect

FIG. 1

GH_2/GO_2 REACTION CONTROL SYSTEM

SYSTEM SCHEMATIC



Besides predicting performance, the computer program includes a heat transfer model designed to predict both steady state and transient engine temperatures. The heat transfer program creates a thermal model from a minimum of input information. Solution is obtained by finite difference techniques. Cooling options allow the user to specify single and multislot film cooling, regenerative cooling, liner cooling, or combinations of the three.

Each of the main submodels of the engine characterization program may be run by itself by providing additional input. The dynamics model can be used alone to study pulsing operation. The combustion model can be utilized alone to describe mixing and combustion of any oxygen-hydrogen ducted or free jet flow. And the heat transfer model may stand by itself to study engine thermal behavior.

Dynamics. - The dynamics model is designed to predict pressures, temperatures, and mixture ratios in the cavities and volumes of a GO_2/GH_2 rocket engine. A spark ignition model is an integral part of the analysis, and instantaneous and integrated performance parameters are part of the output.

Figure 2 shows a dynamics model of an Aerojet $GO2/GH_2$ rocket engine described in Reference 2. The model consists of large volume oxygen and hydrogen supply tanks, injector manifolds, a pilot combustor, the main combustor, and the space sink. The simulation of the valves feeding the pilot and the main combustor requires input opening and closing times and valve flow area is assumed to vary linearly with time.

The flow passages connecting mass accumulation volumes may be of two types. If the passage is identified as an orifice, entrance effects are assumed to dominate friction losses, and a discharge coefficient is specified. If friction is important, the passage is identified as a duct, and duct length is specified.

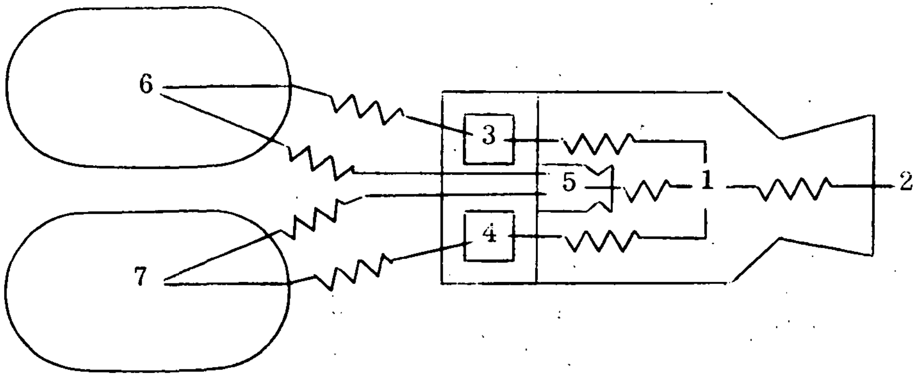
Instanteneous thrust is calculated at each iteration based upon chamber pressure and input thrust coefficient. If the combustion and performance modules have been called, a computed thrust coefficient is passed to the dynamics model for performance computations. Integrated impulse and specific impulse are also calculated.

Injection. - The injection model was written to provide starting profiles of gas velocity, mixture ratio, and temperature used for initialization of the combustion module. The model assumes a flow field consisting of concentric annuli of uniform flow alternating between plugs which are oxygen rich and those which are hydrogen rich relative to the overall mixture ratio. The number of distinct annuli is a function of the number of injection elements.

The velocity, temperature, and flow area for the hydrogen rich and oxygen rich regions are determined to satisfy continuity, momentum, and energy, and to simulate the mixing produced by the injection elements. The degree of injector mixing is specified by an input mixing factor identical to the one usually used to characterize experimental injector performance.

FIG. 2

GO_2/GH_2 ENGINE DYNAMICS MODEL



Cavity	Volume (in^3)
1 (Combustor)	48.3
2 (Space)	∞
3 (H_2 Manifold)	23.8
4 (O_2 Manifold)	32.8
5 (Pilot)	.467
6 (H_2 Tank)	∞
7 (O_2 Tank)	∞

Passage	Open Area (in^2)	Opening Time (MS)	Closing Time (MS)	Electrical On Time (MS)	Electrical Off Time (MS)
1-2 (Throat)	2.895 (fixed)	-	-	-	-
5-1 (Pilot throat)	.0855 (fixed)	-	-	-	-
3-1 (H_2 Orfices)	.438 (fixed)	-	-	-	-
4-1 (O_2 Orfices)	.349 (fixed)	-	-	-	-
6-3 (H_2 Main valve)	.5	10	8.7	28	130
7-4 (O_2 Main valve)	.332	8.7	7.4	26	122
6-5 (H_2 Pilot valve)	.0073	10	10	10	130
7-5 (O_2 Pilot valve)	.012	10	10	0	120

The mass flow, overall mixture ratio, and total pressure required as input by the injection model are provided by the dynamics module when transient combustion and performance calculations are made. Otherwise, these parameters are user specified.

Ignition. - The ignition model is an integral part of the dynamics subroutine. Its purpose is to test for the occurrence of ignition each time a voltage surge is applied to the spark plug. The dynamics model provides the instantaneous mixture ratio and pressure in the vicinity of the plug. When ignition is indicated, the combustor or pilot pressure and temperature are assumed to rise instantly to conditions of chemical equilibrium.

In engine models which incorporate a pilot combustor, the spark plug is assumed to be in the pilot. Otherwise it is assumed to be in the combustor. The ignition model checks first for the occurrence of a spark based upon the spark gap, the local pressure, and the applied potential. If a spark occurs, an ignition test is made which depends upon spark energy and local mixture ratio. If ignition occurs and the model includes a pilot, a test is made for flame quenching in the combustor. The ignition model is based upon theoretical considerations and upon a number of recent O_2/H_2 ignition test results.

Combustion. - The high performance demands placed upon the proposed space shuttle systems has required that more sophisticated prediction techniques be developed for the detailed analysis of the thrust chamber combustion and mixing processes. Most performance evaluations are based upon one-dimensional models. These models may provide adequate definition of the potential performance of a particular motor, but can, at best, account for non-ideal behavior by using correction factors based upon existing engine data. This method may be sufficient for engines of similar, well investigated design, but relatively new thrust chamber designs are not amenable to this type of treatment.

The processes occurring within the thrust chamber are controlled by mechanisms involving fluid mechanical and chemical kinetic phenomena. Turbulent mixing, reaction kinetics and mass transfer along the chamber walls contribute to the overall performance of the thrust chamber. To date little has been done to quantify the coupling of these processes within rocket thrust chambers.

Accordingly, the combustion module was developed using a rather detailed mathematical model designed to describe the turbulent reacting thrust chamber flow field including the effects of film and transpiration cooling.

Performance. - The principal purpose of the performance model is to determine the thrust and specific impulse of the engine at a specified point in time (the time at which a combustion calculation is made). The enthalpy and mixture ratio at each grid point of the combustion program is used as the starting point for the kinetic recombination in the nozzle using a modified Bray criteria. The Isp at each streamline is then mass averaged over all streamlines to find the overall Isp. Boundary layer and divergence losses are deducted along with the Isp loss due to mass addition in the exhaust nozzle.

Heat Transfer. - The heat transfer model is designed to predict engine structure and coolant temperature distribution for GO_2/GH_2 engines utilizing a number of cooling options. The model exchanges no information with other parts of the characterization program and, hence, can be run independently.

At the user's option, the heat transfer subprogram will model the following cooling techniques:

(1)	Conduction cooling	Passive methods
(2)	Heat sink cooling	
(3)	Radiation cooling	
(4)	Film cooling	Active methods
(5)	Regenerative cooling	
(6)	Combustor liner cooling	

Film cooling, with either single or multiple injection stations, may be used with any of the other methods.

In general, the user is required only to specify the combustor shape and combustor material characteristics plus coolant mass flow and delivery geometry. It is assumed that hydrogen gas is used for all active cooling.

The heat transfer subroutine has provision for including an injector thermal model. When this option is utilized, the heat transfer characteristics of the injector must be specified. The subroutine computes transient or steady state injector temperature using the input characteristics and accounting for conduction and radiation from the combustor.

The heat transfer program creates from the input a thermal network which is solved by a finite difference method. All thermal admittances are automatically computed.

Analysis Methods

Dynamics Analysis. - The dynamics model predicts transient and steady state pressures, mixture ratios, and temperatures in the various gas accumulation volumes of a GH_2/GO_2 rocket engine. The gas accumulation volumes are the manifolds, regenerative cooling passages, the combustion chamber, the pilot combustor, etc., which are connected by a system of flow resistances (orifices and ducts). A typical system is shown in Figure 2.

The dynamics model is solved using a finite difference technique. At a given instant in time, the mass flows between the volumes are calculated. Then for a small increment of time, a new system pressure distribution is computed along with new temperatures and mixture ratios. An integral spark ignition model tests for ignition at sparking times. Point and integrated performance parameters are calculated using a thrust coefficient passed from the performance model.

Mass flow calculations: Two types of flow restrictions are provided in the dynamics model. When friction is unimportant, an orifice model is used requiring specification of the discharge coefficient. A duct model is also available for flow restrictions which are characterized by friction losses that are large relative to entrance losses. For either case, a flow admittance is calculated conforming to the following definition

$$\dot{w} = A_D\left(P_{o_1} - P_{o_2}\right) \tag{1}$$

For orifices, the admittance is given by

$$A_D = \frac{AC_d P_{o_1}}{\left(P_{o_1} - P_{o_2}\right)} \sqrt{\frac{\gamma}{RT_{o_1}}}\, F(\gamma) \tag{2}$$

$$F(\gamma) = \sqrt{\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}} \quad \text{for choked flow} \tag{3}$$

$$= \sqrt{\frac{2}{\gamma-1}} \sqrt{\left(\frac{P_{o_2}}{P_{o_1}}\right)^{\frac{2}{\gamma}} - \left(\frac{P_{o_2}}{P_{o_1}}\right)^{\frac{\gamma+1}{\gamma}}} \quad \text{for unchoked flow}$$

The choking pressure ratio is given by

$$\left(\frac{P_{o_2}}{P_{o_1}}\right)_{\text{choking}} = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{4}$$

For ducts, where pressure loss due to friction is important, mass flow depends on the friction factor parameter, $4fL/_D$, as well as the pressures. The choking pressure ratio is given by a polynomial function of the friction parameter obtained by curve fitting the results given in Reference 3, Page 175.

$$\left(\frac{P_{o_2}}{P_{o_1}}\right)_{\text{choking}} = C_1 + C_2\left(\frac{4\,fL}{D}\right) + C_3\left(\frac{4\,fL}{D}\right)^2 + \ldots\ldots \tag{5}$$

The flow admittance is also obtained from the results shown in Reference 3.

$$A_D = \frac{A\,B\,P_{o1}}{\left(P_{o1} - P_{o2}\right)\sqrt{RT_{o1}}} \qquad (6)$$

B is a function of $(4fL/_D)$ for choked flow and a function of $4fL/_D$ and P_{o2}/P_{o1} for unchoked flow. The dynamics model uses B in the form of polynomial equations.

The friction factor, 4f, is a function of the flowrate itself, and hence an iterative scheme is required to determine admittance. Starting with an initial guess at 4f, the flow Reynolds number is calculated to give a revised estimate. If the flow is laminar ($Re < 2100$), the friction factor is given by

$$4f = \frac{64}{Re} \qquad (7)$$

If the flow is turbulent, 4f is calculated using a polynomial curve fit of friction factor data for smooth pipes.

Pressure calculations: The pressure change in engine volume I between time t and time $t + \Delta t$ is given by

$$P_o(I)_{t+\Delta t} = P_o(I)_t + \frac{\Delta t}{C(I)_{t+\Delta t}} \cdot \sum_{\substack{\text{all J}\\ \text{connected to I}}} \dot{w}\,(J) \qquad (8)$$

C (I) is the volume capacity defined as follows

$$C(I)_{t+\Delta t} = \frac{V\,(I)\;MW(I)_{t+\Delta t}}{\bar{R}\;\;T_o\;(I)_{t+\Delta t}} \qquad (9)$$

The temperature in volume I when volume I gas is unignited at time $t + \Delta t$ is given by

$$T_o(I)_{t+\Delta t} = \frac{H(I)_t + \Delta t \sum_{\text{all J connected to I}} \dot{q}\,(J)_t}{\left[W(I)_t + \Delta t \sum \dot{w}\,(J)_t \left(3.5 - 3.26\;M\;(I)_{t+\Delta t}\right)\right]} \qquad (10)$$

The parameter $\dot{q}$ (J) is the thermal energy flow through restriction J given by

$$\dot{q}\;(J) = T_o\;(J)\;\dot{w}\;(J)\;\;(3.5 - 3.26\;m\;(J))$$

for cold flow. For ignited flow, $\dot{q}\,(J)$ is calculated using a polynomial equation in $m(J)$. $T_o(J)$ and $m\,(J)$ are the total temperature and percent O_2 associated with restriction J, i.e., those properties existing in the higher pressure volume of the two which J connects.

It is assumed that only the ignitor and the combustor can sustain ignition. After ignition is detected, the temperature and molecular weight in these volumes is determined using polynomial equations which were based upon chemical equilibrium calculations for the GO_2/GH_2 system.

In order to simulate valve opening time and valve sequencing, the dynamics model has provision to express the restriction area as a linear function of time. For time less than the electrical ON time or greater than the sum of the electrical OFF time and the closing time, the restriction area is zero. For time greater than the sum of the electrical ON time and the opening time, but less than the electrical OFF time, the restriction area is that given in the input form. During opening and closing, the restriction area is assumed to be a linear function of time. Figure 3 shows restriction area as a function of time.

Time increment calculations: The time increment, Δt, used to advance time in the dynamics model, is determined each pass through the model to assure that the solution remains stable. The stable time increment is derived below.

Equation (8) can be rewritten to express the pressure in Volume I at time $(t + \Delta t)$ as the weighted sum of the pressure in I at time t and the pressures in all volumes connected to I

$$P_o(I)_{t+\Delta t} = P_o(I)_t \left[1 - \frac{\Delta t}{C(I)_{t+\Delta t}} \sum_J A_D(J) \right] + \frac{\Delta t}{C(I)_{t+\Delta t}} \sum_J \left[A_D(J)\, P_o(K)_t \right] \quad (11)$$

where restrictions J connect volume I to volumes K.

The coefficients of $P_o(K)_t$ are always positive, but the coefficient of $P_{ot}(I)$ will be negative if

$$\Delta t > \frac{C(I)_{t+\Delta t}}{\sum A_D(J)} \quad (12)$$

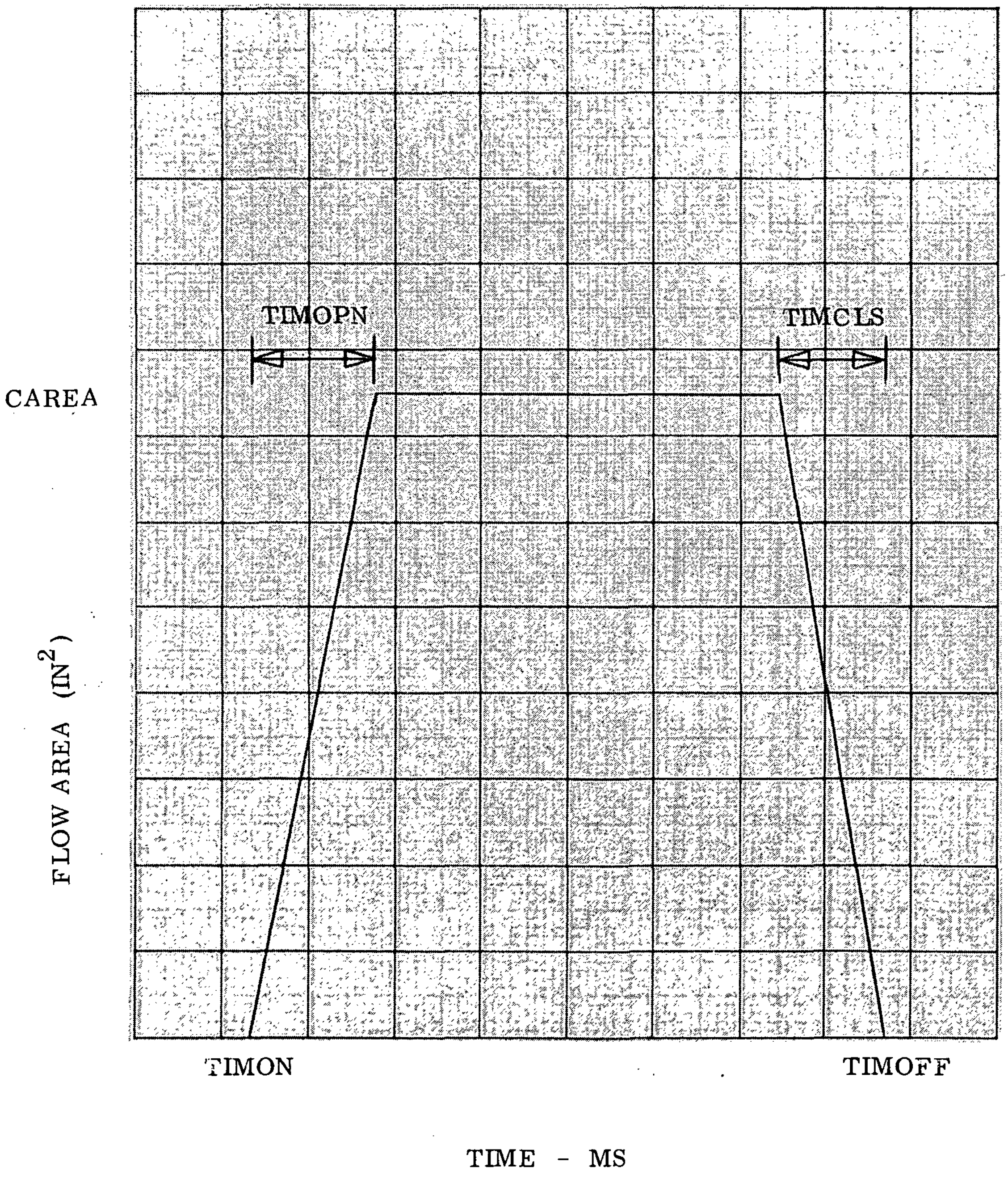
A negative coefficient implies that the higher $P_o(I)$ is at time t, the lower it will be at time $(t + \Delta t)$ which is physically absurd. Therefore, Δt is made small enough to avoid negative coefficients in the system by assuring that

$$\Delta t \le \min_{\text{all } I} \left[\frac{C\,(I)_{t+\Delta t}}{\sum\limits_{\text{all J connected to I}} A_D(J)} \right] \quad (13)$$

FIG. 3

SUBROUTINE DYNAM VARIABLE

FLOW AREA MODEL



Dynamic performance calculations: Thrust, specific impulse, and integrated specific impulse are calculated at each increment of time. The thrust is calculated using:

$$F = P_o A^* C_F \tag{14}$$

Thrust coefficient, C_F, is input or, if combustion has been called, is passed from the performance routine.

Accumulated impulse if given by

$$I\,(t) = \sum_{i=1}^{N} F_i (\Delta t)_i \tag{15}$$

Integrated specific impulse if given by

$$I_{sp}(t) = \frac{I(t)}{\dot{w}_T^*\,(t)} \tag{16}$$

Injection analysis. - The injection model provides starting profiles to the combustion module. The user must input the injector face diameter, the number of injection elements, the propellant temperatures, and the required mixing efficiency. Total mass flow, total pressure, and overall mixture ratio are either user supplied or passed from the dynamics model.

The injection model calculates a flow field consisting of concentric annuli of uniform flow alternating between fuel rich and oxidizer rich regions. The flow conditions are determined to satisfy continuity, conserve energy and momentum, and to achieve a desired degree of injector-inducted mixing.

The injection model analysis begins with the assumption that the sum of the oxygen mass flows in the oxygen rich annuli is equal to the sum of the oxygen mass flows in the fuel rich annuli.

$$\begin{aligned}
w_{O_2}{}' &= 1/2\, m \dot{w}_T - \delta \\
w_{O_2}{}'' &= 1/2\, m \dot{w}_T + \delta \\
w_{H_2}{}' &= (1 - m)\ 1/2\ \dot{w}_T + \frac{\delta}{m} \\
w_{H_2}{}'' &= (1 - m)\ 1/2\ \dot{w}_T - \frac{\delta}{m}
\end{aligned} \tag{17}$$

The δ is determined by iteration to yield the input mixing factor defined as in Reference 14 by

$$E_m = 1 - \left[\left(\frac{\dot{w}_{O_2}{}' + \dot{w}_{H_2}{}'}{\dot{w}_T}\right)\left(\frac{m - m'}{m}\right) + \left(\frac{\dot{w}_{O_2}{}'' + \dot{w}_{H_2}{}''}{\dot{w}_T}\right)\left(\frac{m'' - m}{1 - m}\right)\right] \qquad (18)$$

$$\text{where } m' = \frac{\dot{w}_{O_2}{}'}{\dot{w}_{O_2}{}' + \dot{w}_{H_2}{}'}$$

$$m'' = \frac{\dot{w}_{O_2}{}''}{\dot{w}_{O_2}{}'' + \dot{w}_{H_2}{}''}$$

It is assumed that both oxidizer rich and fuel rich regions have the same total pressure, the same ratio of specific heat and have the same static pressure and Mach number. Therefore, the ratio of total fuel rich flow area to total oxidizer rich flow area is given by

$$\frac{A'}{A''} = \frac{\dot{w}_{O_2}{}' + \dot{w}_{H_2}{}'}{\dot{w}_{O_2}{}'' + \dot{w}_{H_2}{}''}\sqrt{\frac{T_o{}'}{T_o{}''}\left(\frac{MW''}{MW'}\right)} \qquad (19)$$

The total temperature in the fuel rich region is

$$T_o{}' = \frac{.24\, m' T_{O_2} + 3.5\, T_{H_2} - 3.5\, m' T_{H_2}}{3.5 - 3.26\, m'} \qquad (20)$$

and the molecular weight is

$$MW' = \frac{64.51}{2.016\, m' + (32\,(1-m'))} \qquad (21)$$

Similar expressions apply in the oxidizer rich region.

To find velocities in the fuel and oxidizer rich zones, the following mass flux expression is solved for Mach number using the Newton-Ralphson method:

$$\frac{\dot{w}_{O_2}{}' + \dot{w}_{H_2}{}'}{A'} = \left(\rho_o{}' \; a_T{}'\right) \left(1 - \frac{M^2}{5}\right)^{-\frac{5}{2}} \left(\frac{M^2}{1 + .2M^2}\right)^{1/2} \qquad (22)$$

where

$$\rho_o{}' = \frac{P_o\,(MW')}{(18540)\;T_o{}'}$$

$$a_T{}' = \sqrt{\frac{\gamma \bar{R}\, T_o{}'}{MW'}} \qquad (23)$$

$$u' = a_T{}' \sqrt{\frac{M^2}{1 + .2M^2}}$$

Again, similar expressions apply to the oxidizer rich zone.

The number of flow annuli used in the injection model is determined by the following heuristic rule based on the number of injection elements:

$$N_E = 1 + \sum_{n=1}^{N_A} 6n \qquad (24)$$

The width of each annulus (and the radius of the inner circular region) is simply the injector radius divided by N_A. Each annulus contains a region of oxidizer rich flow and a region of fuel rich flow. The radii of the discontinuities are determined to satisfy the flow area requirement, equation 19.

After the physical flow field has been determined, the injector model converts to the stream function coordinates used in the combustion model. For axisymmetric coordinates used in the injection model, the von Mises transformation is defined by

$$\int_o^{\psi} \psi \, d\psi = \int_o^{\psi} \rho u y dy \qquad (25)$$

Once the stream function coordinates of the flow discontinuities have been calculated, point flow conditions for the input number of combustion model grid points can be determined. The grid points are separated by equal stream function increments. It is recommended that, for good resolution, about (10 x N_A) grid points be specified.

Ignition. - The ignition model is an integral part of the dynamics calculation designed to predict sparking, ignition, and quenching phenomena. If a pilot is present in the dynamic system, sparking and ignition is tested for in the pilot and a test is made for flame quenching in the combustor. If there is no pilot in the dynamic system, the sparking and ignition tests are made on combustor conditions.

A spark is assumed to occur when voltage is applied to the plug and the potential is sufficient for breakdown. Breakdown potential, it is assumed, is a function of spark gap and local pressure. Second order effects of gas velocity, electrode design, and gas mixture ratio are not considered. Figure 4 shows breakdown potential versus the product of pressure and spark gap following Paschen's Law modified by published test results (Reference 4). This criterion is utilized in the ignition model as a sparking test using a polynomial curve fit to describe the breakdown line.

If a spark occurs, a test is made to determine if the spark energy is sufficient to ignite the ambient gas. The primary factor influencing minimum ignition energy is gas mixture ratio. Figure 5 shows minimum ignition energy as a function of mixture ratio. This curve, in the form of a polynomial equation, is used as the ignition criterion. Figure 5 was derived using numerous test results, some of which are shown. Secondary effects which are not included in the ignition criterion are gas velocity, electrode design, and spark gap.

If the dynamics model contains a pilot, a test is made to determine if the flame will be sustained in the combustor. Figure 6, the result of work reported in Reference 4, is used as a quenching criterion. The quenching parameters are mixture ratio, pressure, and combustor diameter. The flammability threshold is used in the ignition routine in the form of a polynomial equation. The second order effects of flame temperature and wall temperature are not considered in the quenching model.

Mixing and combustion model. -

Description of techniques:

1. Describing equations. - The starting point for the mixing and combustion model is the boundary layer form of the conservation equations for global mass momentum and energy and element and species diffusion. A solution of this system provides the details of the flow field including the velocity, temperature, and species fields.

The global continuity equation can be eliminated from the system of differential equations by introducing the von Mises coordinates as the independent variable. The transformation $(x,r)\rightarrow(x,\Psi)$ is defined according to the relations:

$$\rho u r^N = \Psi^N \Psi_Y \tag{26}$$

$$-\rho v r^N = \Psi^N \Psi_X$$

FIG. 4

SPARK BREAKDOWN VOLTAGE CRITERIA

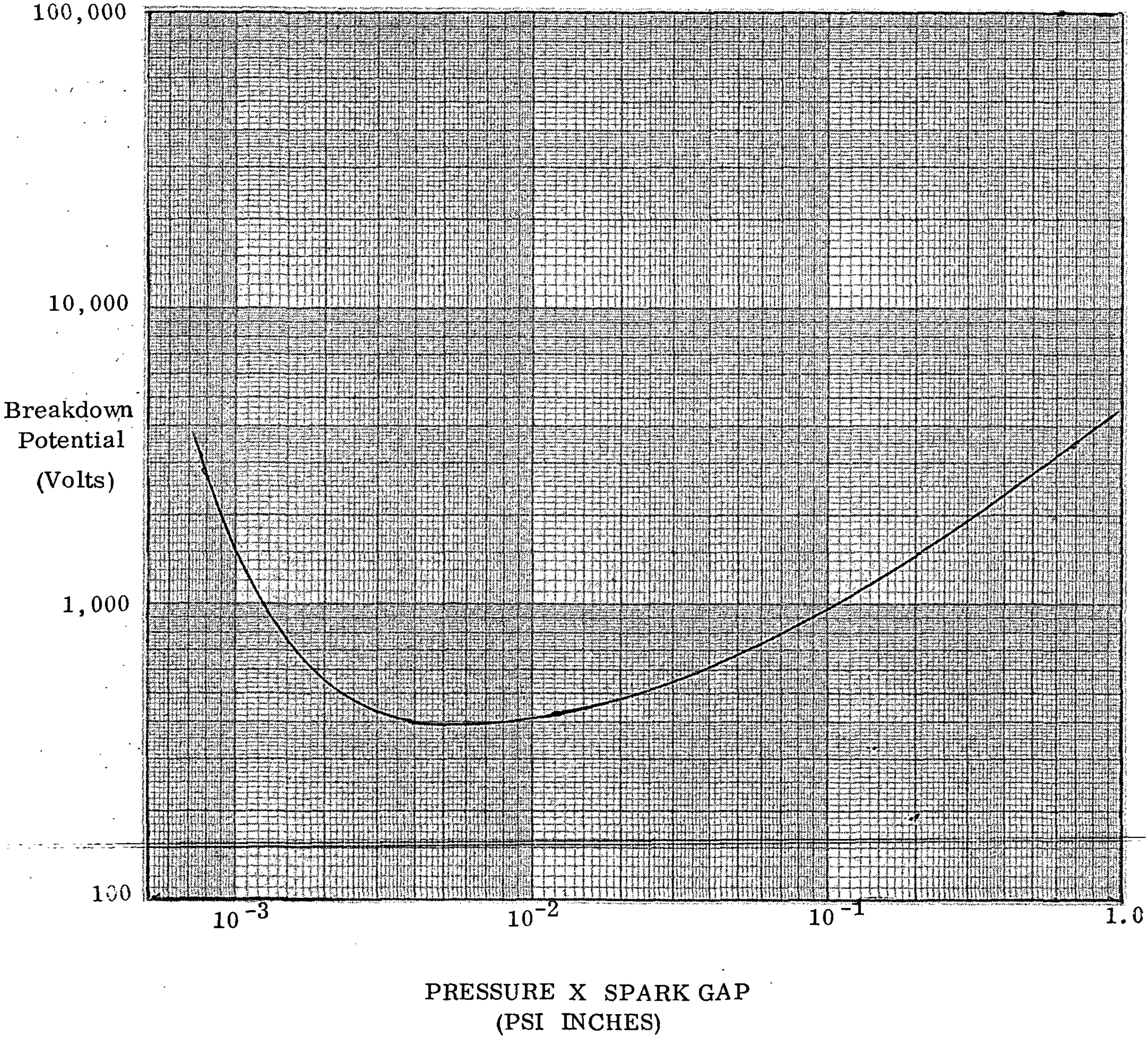


FIG. 5

MINIMUM IGNITION ENERGY

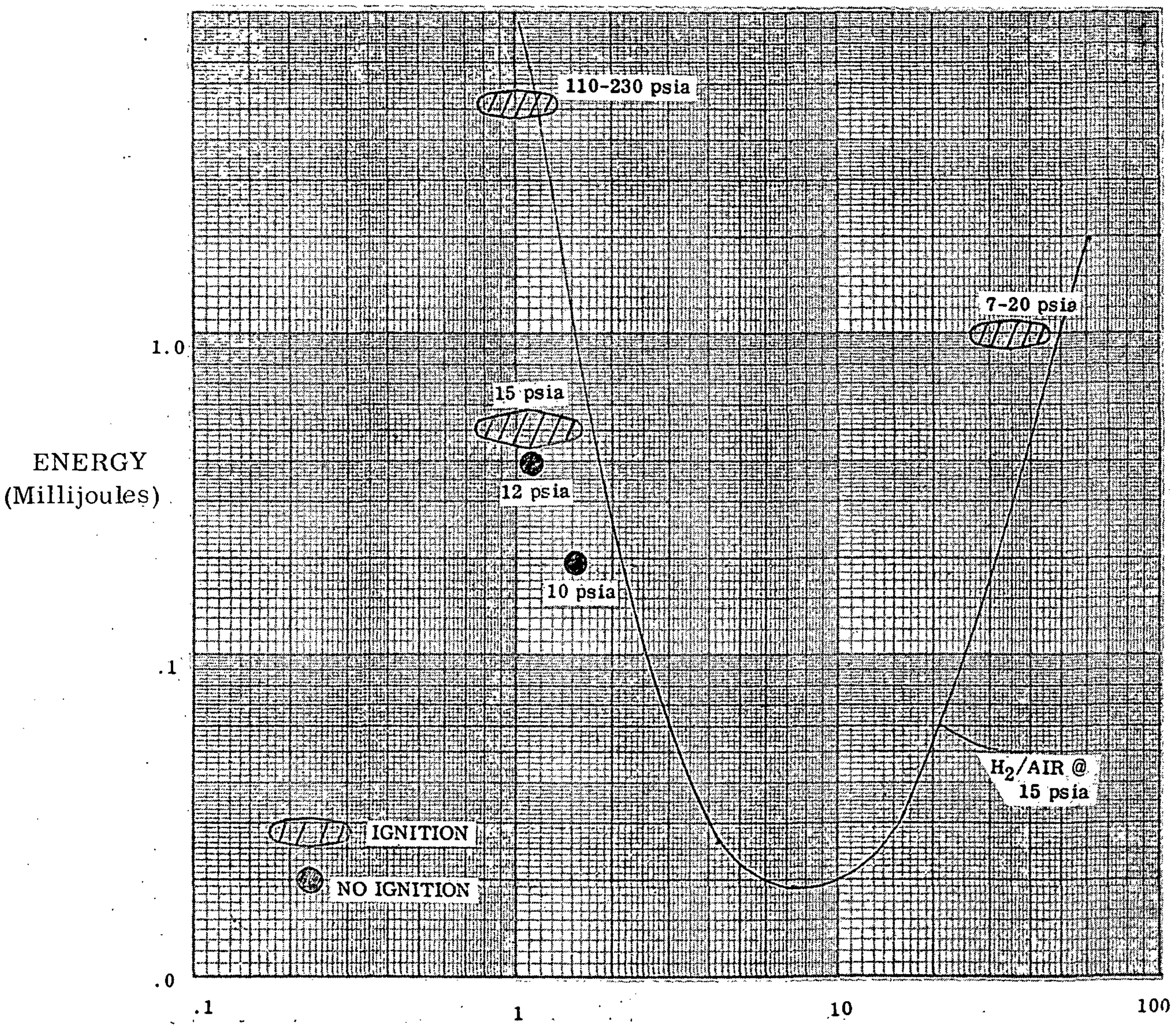


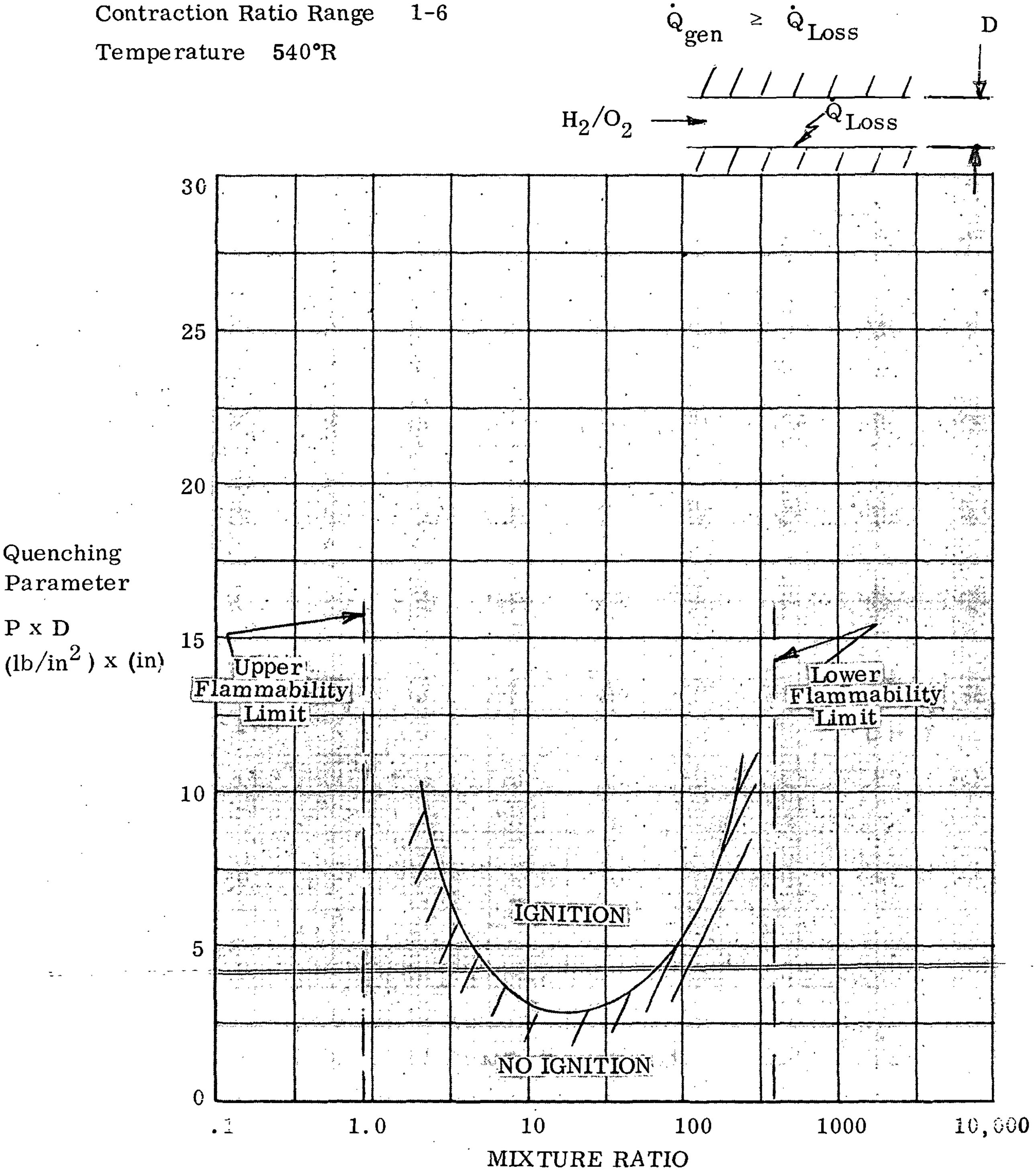
FIG. 6

EFFECT OF MIXTURE RATIO ON FLAME QUENCHING PARAMETER*

$$(PD)_{Limit} \approx RT_o \left[\frac{\Delta h(T_F - T_W)}{K \Delta H_r \exp(-E/RT_F)} \right]$$

Contraction Ratio Range 1-6

Temperature 540°R



* "Ignition System for Space Shuttle APS" Aerojet QTPN 1678-Q-1, 12 Oct. 1970

where N = 0 - plane two-dimensional flow
1 - axisymmetric flow

Introduction of these equations into the differential equations results in:

Element Conservation

$$\frac{\partial \tilde{\alpha}_j}{\partial x} = (1/\Psi^N) \frac{\partial}{\partial \Psi} \left[(Le/Pr) \quad (\rho u \mu/\Psi^N) r^{2N} \frac{\partial \tilde{\alpha}_j}{\partial \Psi} \right] \tag{27}$$

where

$$\tilde{\alpha}_j = \sum_i \nu_{ji} \ (W_j/W_i) \ \alpha_i \tag{28}$$

and ν_{ji} is the amount of element j in specie i and the W's are the molecular weights.

Specie Conservation

$$\frac{\partial \alpha_i}{\partial x} = (1/\Psi^N) \frac{\partial}{\partial \Psi} \left[(Le/Pr) \quad (\rho u \mu/\Psi^N) r^{2N} \frac{\partial \alpha_i}{\partial \Psi} \right] + \dot{w}_i/\rho u \tag{29}$$

where $\dot{w}_i$ is the volumetric rate of production of specie i. Note also that:

$$\sum_i \nu_{ji} \ \dot{w}_i (W_j/W_i) = 0 \tag{30}$$

Equations (27) and (29) are used depending upon whether equilibrium or non-equilibrium chemistry is considered. If equation (27) is used then i-j equilibrium relations are required which are supplied by the indetermancy approached by the production term, $\dot{w}_i$, as equilibrium is attained. This formulation is used in connection with the complete combustion, or diffusion controlled, limit. Equation (29) is used when the full H_2/O_2 kinetics is considered.

Momentum

$$\frac{\partial u}{\partial x} = (1/\Psi^N) \frac{\partial}{\partial \Psi} \left[(\rho u \mu/\Psi^N) r^{2N} \frac{\partial u}{\partial \Psi} \right] - (1/\rho u) \frac{dp}{dx} \tag{31}$$

Energy

$$\frac{\partial H}{\partial x} = (1/\Psi^N) \frac{\partial}{\partial \Psi} \left\{ (\rho u \mu / Pr \Psi^N) r^{2N} \left[\frac{\partial H}{\partial \Psi} + (Pr-1) \frac{\partial u^2/2}{\partial \Psi} + \right.\right.$$

$$\left.\left. + \sum_i h_i (Le-1) \frac{\partial \alpha_i}{\partial \Psi} \right] \right\} \qquad (32)$$

To supplement these conservation equations, relations among the thermodynamic variables are required, viz.,

State:

$$\rho = \frac{p}{RT \sum_i (\alpha_i/W_i)} \qquad (33)$$

also

$$H = h + \frac{u^2}{2} \qquad (34)$$

where

$$h = \sum_i \alpha_i h_i \qquad (35)$$

with*

$$h_i = h_i(T) \qquad (36)$$

In addition, representations for the turbulent transport coefficients μ, Pr and Le, are required as well as specification of the chemical system and its associated rate constants.

*This equation is implemented within the program by standard enthalpy temperature subroutines based on thermochemical data from the JANNAF tables.

With regard to the transport coefficients, the numerical analysis has been structured in such a way as to provide complete generality in evaluation of these parameters. That is, they are computed locally and could ultimately be specified as functions of the local values of the mean flow variables. At the present time, however, the options which have been provided for include only the following:

For Le and Pr - any non-zero constant value specified by the user

For μ - (a) any non-zero constant value specified by the user

(b) a modified form of the model due to Hirsch (Reference 5) which can be written

$$\mu = k \frac{\rho u D}{1+x/S} \tag{37}$$

where k is a constant which can be input by the user, D is the local thrust chamber diameter and S is the injector element spacing which is also a program input.

For the chemical system we note first that the $\dot{w}_i$ are given by

$$\dot{w}_i = W_i \sum_{p=1}^{R} (\nu''_{ip} - \nu'_{ip}) k_{f,p} \rho^{m_p} \prod_{i=1}^{N} \left(\frac{\alpha_i}{W_i}\right)^{\nu'_{ip}} \left[1 - \left(\frac{\rho^{N_p}}{k_{c,p}}\right) \prod_{i=1}^{N} \left(\frac{\alpha_i}{W_i}\right)^{\nu''_{ip} - \nu'_{ip}}\right]$$

$$m_p \equiv \sum_{i=1}^{N} \nu'_{ip} \quad ; \quad N_p = \sum_{i=1}^{N} (\nu''_{ip} - \nu'_{ip}) \tag{38}$$

for a chemical system containing N species entering into R elementary reactions given by

$$\sum_{i=1}^{N} \nu'_{ip} M_i \underset{k_{b,p}}{\overset{k_{f,p}}{\rightleftharpoons}} \sum_{i=1}^{N} \nu''_{ip} M_i \qquad p = 1,2 \ldots R \tag{39}$$

The present study employs the reaction mechanism given in Table II which involves 8 species entering into a total of 17 reactions.

TABLE II. H_2/O_2 CHEMICAL SYSTEM

Reaction No.	Reaction	A	B	E/R(1°K)
1	$HO_2 + H = O_2 + H_2$	6×10^{13}	0	0
2	$HO_2 + H = OH + OH$	6×10^{13}	0	0
3	$HO_2 + O = O_2 + OH$	1×10^{13}	0	0
4	$HO_2 + OH = O_2 + H_2O$	1×10^{13}	0	0
5	$H_2 + OH = H + H_2O$	2.19×10^{13}	0	2,593
6	$OH + OH = O + H_2O$	5.75×10^{12}	0	392.7
7	$H_2 + O = H + OH$	1.74×10^{13}	0	4,758
8	$O_2 + H = OH + O$	2.29×10^{14}	0	8,459
9	$H_2O_2 + OH = H_2O + HO_2$	1×10^{13}	0	906.3
10	$H_2O_2 + H = H_2 + HO_2$	2.34×10^{13}	0	4,632
11	$H_2O_2 + H = OH + H_2O$	3.18×10^{14}	0	4,532
12	$O + H + M = OH + M$	1×10^{16}	0	0
13	$O + O + M = O_2 + M$	9.38×10^{14}	0	0
14	$H + H + M = H_2 + M$	5×10^{15}	0	0
15	$H + OH + M = H_2O + M$	1×10^{17}	0	0
16	$O_2 + H + M = HO_2 + M$	1.59×10^{15}	0	-503.5
17	$OH + OH + M = H_2O_2 + M$	8.4×10^{14}	0	-2,669

Notes:

$$k_f = AT^B e^{-E/RT}$$

$$k_b = k_f/k_c$$

k_c is calculated internally by the computer at each temperature utilizing free energy of formation of each reactant

To complete this formulation, initial and boundary conditions must be specified.

- Initial conditions - The initial conditions must represent the details of the flow emerging from the near region and, therefore, must allow for the specification of velocity, temperature, and composition profiles. Modeling of the initial conditions is carried out as follows: referring to Figure 7, the actual pattern emerging from the near region is divided into an arbitrary number of annuli. To remove any three-dimensionality, the pertinent variables are circumferentially mass-averaged in each annulus. The mass averaging ensures conservation of mass, energy, and momentum. Depending upon the particular pattern, smooth rather than stepped profiles may be appropriate.

These conditions are given by:

@ $x = 0$ (starting station) and $0 \leq r \leq r_w(0)$

$$0 \leq r \leq r_1 \qquad \begin{cases} u = u_1(r) \\ T = T_1(r) \\ \alpha_i = \alpha_{i,1}(r) \end{cases}$$

$$r_1 < r \leq r_2 \qquad \begin{cases} u = u_2(r) \\ T = T_2(r) \\ \alpha_i = \alpha_{i,2}(r) \end{cases}$$

$$\vdots$$

$$r_{k-1} < r \leq r_k \qquad \begin{cases} u = u_k(r) \\ T = T_k(r) \\ \alpha_i = \alpha_{i,k}(r) \end{cases}$$

- Boundary conditions - thrust chamber geometry - To provide the versatility of either predicting behavior of existing hardware of designing new hardware, the analysis was developed to permit specification of the chamber contour or the axial pressure distribution. Specifying one renders the other a dependent variable. Thus, for $x \geq 0$

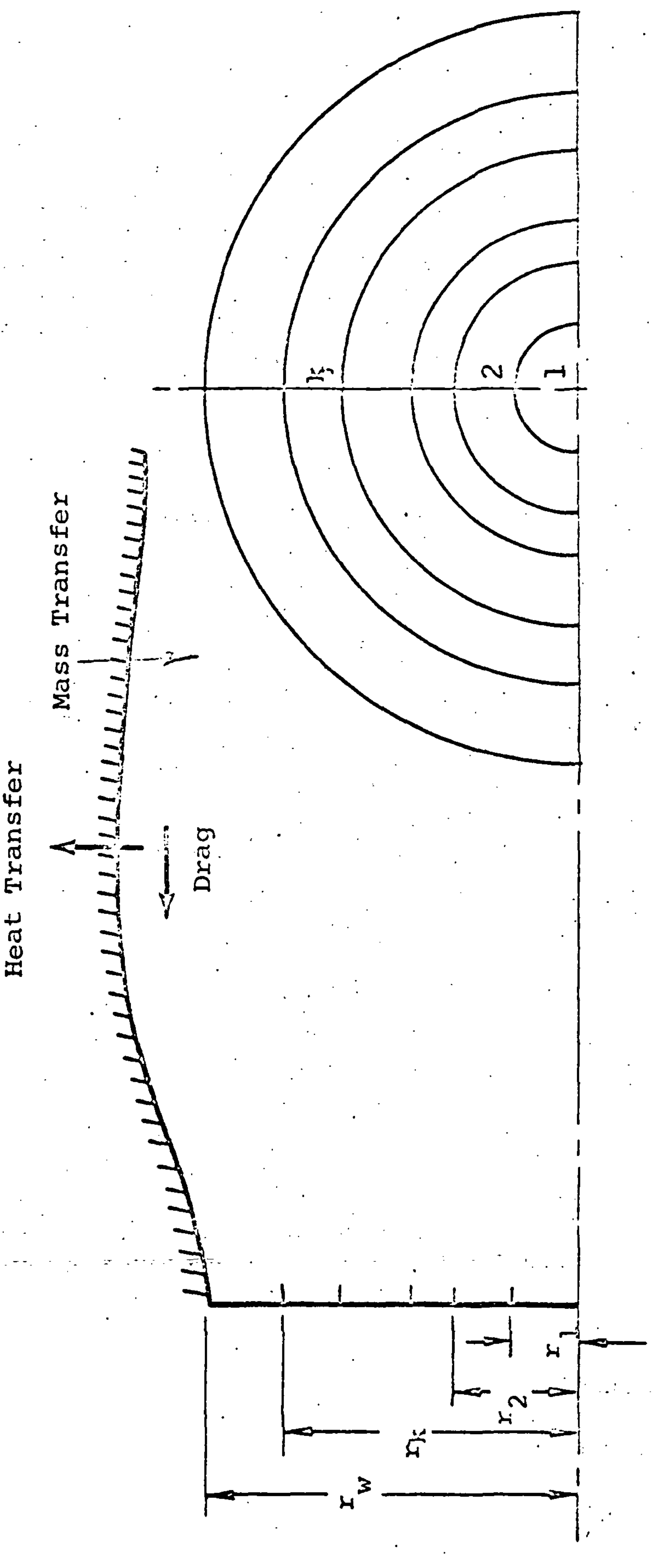
if $p(x)$ is prescribed

$r_w(x)$ is computed

if $r_w(x)$ is prescribed

$p(x)$ is computed.

FIG. 7



MODEL OF INITIAL CONDITIONS

This option provides the capability to evaluate the effect of acceleration or deceleration of the chamber flow upon the mixing rate, ignition and flame propagation rate, and local wall heat transfer for a given injection configuration.

- Boundary Conditions - Axis

for $x \geq 0$ and $r = 0$

$$\frac{\partial \alpha_i}{\partial \Psi} = \frac{\partial \tilde{\alpha}_j}{\partial \Psi} = \frac{\partial T}{\partial \Psi} = \frac{\partial H}{\partial \Psi} = \frac{\partial u}{\partial \Psi} = 0$$

- Wall boundary conditions - The interaction of the chamber flow with the wall involves drag, heat transfer, and, in general, mass transfer - the latter being a consideration for transpiration and/or film cooled chambers. The various combinations of boundary conditions which have been implemented in this computer program are indicated in Table III. Note that for the transpiration cooling model an explicit boundary condition for a "tracer" specie is included. This is needed for proper implementation of the boundary condition in this case as discussed with the chamber cooling models. The manner in which "bulk" values of the several parameters indicated by subscript b are evaluated, is also described in the chamber cooling section.

2. Computational procedures. - The solution of the above system of equations is obtained employing an explicit finite difference technique. Figure 8 shows a generic point, (n+1, m) in the x-ψ grid network. The finite difference formulation for the calculation of the flow at the point (n+1, m) is obtained by using the following explicit/difference relations where P is any one of the pertinent variables.

$$\frac{\partial P}{\partial x} = \frac{1}{\Delta x}(P_{n+1,m} - P_{n,m}) \tag{40}$$

$$\frac{\partial P}{\partial \Psi} = \frac{1}{2\Delta\Psi}(P_{n,m+1} - P_{n,m-1}) \tag{41}$$

$$\frac{\partial}{\partial \Psi} b \frac{\partial P}{\partial \Psi} = \frac{1}{\Delta\Psi^2}\left[b_{n,m+\frac{1}{2}}(P_{n,m+1} - P_{n,m}) - b_{n,m-\frac{1}{2}}(P_{n,m} - P_{n,m-1})\right] \tag{42}$$

TABLE III. WALL BOUNDARY CONDITIONS

(@ $x \geq 0$; $r = r_W(x)$)

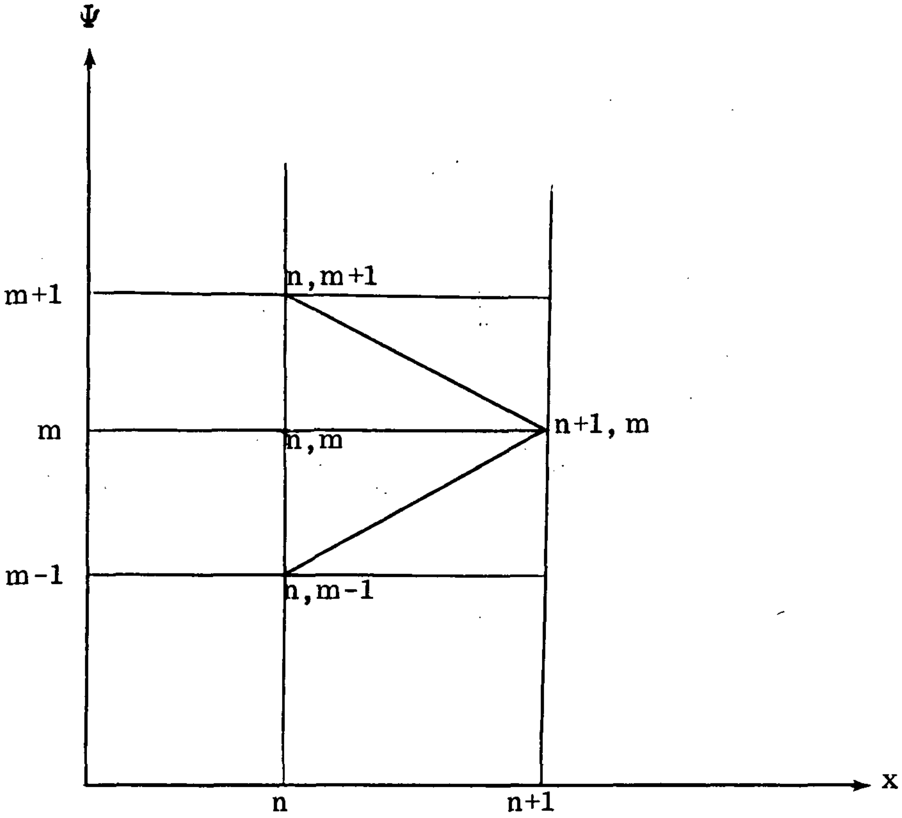
Cooling Model / Variable	Film	External Regenerative	Transpiration
H(energy transfer)	$\frac{\partial H}{\partial \Psi} = 0$	$\frac{\partial H}{\partial \Psi} = - q_L (\frac{Pr}{\mu})_b \frac{\Psi^N}{\rho u r^N}$	$H = H(\alpha_i, T)$
α_i (mass transfer)	$\frac{\partial \alpha_i}{\partial \Psi} = 0$		$\frac{\partial \alpha_c}{\partial \Psi} = \dot{m}_c (1-\alpha_c) \left(\frac{P_r}{L_e \mu}\right)_b \left(\frac{\Psi^N}{\rho u r^N}\right)$; coolant $\frac{\partial \alpha_T}{\partial \Psi} = \dot{m}_c (\alpha_{T_c} - \alpha_T) \left(\frac{P_r}{L_e \mu}\right)_b \left(\frac{\Psi^N}{\rho u r^N}\right)$; tracer $\frac{\partial \alpha_i}{\partial \Psi} = - \dot{m}_c \alpha_i \left(\frac{P_r}{L_e \mu}\right)_b \left(\frac{\Psi^N}{\rho u r^N}\right)$; $i \neq c, T$
u(shear)	$\frac{\partial u}{\partial \Psi} = - (\frac{\rho u^2}{\mu})_b \frac{\Psi^N}{\rho u r^N} \frac{c_f}{2}$		

Note: For the transpiration cooling model $\dot{m}_c(x)$ is specified and $T_w(x)$ computed
or $T_w(x)$ is specified and $\dot{m}_c(x)$ computed

For the external regenerative cooling model
$q_L(x)$ is specified and $T_w(x)$ computed
or $T_w(x)$ is specified and $q_L(x)$ computed

FIG. 8

FINITE DIFFERENCE GRID NETWORK



where

$$b = (\rho u \mu)/\Psi^{N})\, r^{2N} \tag{43}$$

$$b_{n,m\pm\frac{1}{2}} = \frac{1}{2}(b_{n,m} + b_{n,m\pm 1}) \tag{44}$$

and

$$\Psi = m \Delta \Psi \tag{45}$$

The conservation equations in difference form are:

Elements:

m = 0;

$$(\tilde{\alpha}_j)_{n+1,0} = (\tilde{\alpha}_j)_{n,0} \cdots$$

$$\cdots + \frac{2\Delta x\,(N+1)}{(\Delta\Psi)^2}\left[(\mathrm{Le}\mu/\mathrm{Pr})\,(\rho u)^{1-N}\right]_{n,0}\left[(\tilde{\alpha}_j)_{n,1} - (\tilde{\alpha}_j)_{n,0}\right] \tag{46A}$$

m≠0

$$(\tilde{\alpha}_j)_{n+1,m} = (\tilde{\alpha}_j)_{n,m} + \frac{\Delta x}{m^N (\Delta\Psi)^{2+N}}\Big\{(\mathrm{Le}\ b/\mathrm{Pr})_{n,m+\frac{1}{2}}(\tilde{\alpha}_j)_{n,m+1} \cdots$$

$$\cdots - \left[(\mathrm{Le}\ b/\mathrm{Pr})_{n,m+\frac{1}{2}} + (\mathrm{Le}\ b/\mathrm{Pr})_{n,m-\frac{1}{2}}\right](\tilde{\alpha}_j)_{n,m} \cdots$$

$$\cdots + (\mathrm{Le}\ b/\mathrm{Pr})_{n,m-\frac{1}{2}}\ (\tilde{\alpha}_j)_{n,m-1}\Big\} \tag{46B}$$

Note: The species conservation equations have the identical form with the production term added to the right hand side.

Momentum

$m = 0$

$$u_{n+1,0} = u_{n,0} + \frac{2\Delta x\,(N+1)}{(\Delta\Psi)^2}\left[\mu\,(\rho u)^{1-N}\right]_{n,0}\left[u_{n,1} - u_{n,0}\right] \cdots$$

$$\cdots - \frac{\Delta x}{(\rho u)_{n,0}}\left(\frac{dp}{dx}\right)_{n+1} \tag{47}$$

$m \neq 0$

$$u_{n+1,m} = u_{n,m} + \frac{\Delta x}{m^N\,(\Delta\Psi)^{2+N}}\Big\{b_{n,m+\frac{1}{2}} \quad u_{n,m+1}$$

$$\cdots - u_{n,m}\left(b_{n,m+\frac{1}{2}} + b_{n,m=\frac{1}{2}}\right) + b_{n,m-\frac{1}{2}}\,u_{n,m-1}\Big\}$$

$$\cdots - \left(\frac{\Delta x}{\rho u}\right)_{n,m}\left(\frac{dp}{dx}\right)_{n+1} \tag{48}$$

Energy

$m = 0$

$$H_{n+1,0} = H_{n,0} + \frac{2\Delta x\,(N+1)}{(\Delta\Psi)^2}\left[\mu\,(\rho u)^{1-N}_{n,0}\right]\Big\{\left(\frac{1}{Pr}\right)_{n,0}\left(H_{n,1} - H_{n,0}\right) \cdots$$

$$\cdots \frac{1}{2}\left(1 - \frac{1}{Pr}\right)_{n,0}\left(u^2_{n,1} - u^2_{n,0}\right) \cdots$$

$$\cdots \sum_i\left[(\alpha_i)_{n,1} - (\alpha_i)_{n,0}\right]\left(h_i\,\frac{Le-1}{Pr}\right)_{n,0}\Big\} \tag{49}$$

$m \neq 0$

$$
\begin{aligned}
H_{n+1,m} = H_{n,m} + \frac{\Delta x}{m^N (\Delta\Psi)^{2+N}} \Big\{ & (b/Pr)_{n,m+\frac{1}{2}} H_{n,m+1} \cdots \\
& - (b/Pr)_{n,m+\frac{1}{2}} + (b/Pr)_{n,m-\frac{1}{2}} H_{n,m} \cdots \\
& + (b/Pr)_{n,m-\frac{1}{2}} H_{n,m-1} + \frac{1}{2}(b-b/Pr)_{n,m+\frac{1}{2}} u^2_{n,m+1} \cdots \\
& - \frac{1}{2}\Big[(b-b/Pr)_{n,m+\frac{1}{2}} + (b-b/Pr)_{n,m-\frac{1}{2}}\Big] u^2_{n,m} \cdots \\
+ \frac{1}{2} (b-b/Pr)_{n,m-\frac{1}{2}} u^2_{n,m-1} & + \sum_i (\alpha_i)_{n,m+1} \Big[bh_i \Big(\frac{Le-1}{Pr}\Big)\Big]_{n,m+\frac{1}{2}} \cdots \\
- \sum_i (\alpha_i)_{n,m} \Big[& \Big(bh_i \frac{Le-1}{Pr}\Big)_{m,m+\frac{1}{2}} + \Big(bh_i \frac{Le-1}{Pr}\Big)_{n,m-\frac{1}{2}} \Big] \cdots \\
+ \sum_i (\alpha_i)_{n,m-1} & \Big(bh_i \frac{Le-1}{Pr}\Big)_{m,m-\frac{1}{2}} \Big\}
\end{aligned}
\tag{50}
$$

The boundary conditions are expressed in finite difference form by using Equation (41). The use of such a central differencing scheme at the wall is implemented by carrying along an additional streamline above the wall.

- Step size control - The step size in the explicit finite difference scheme is controlled by a stability criterion and from studies of linear parabolic partial differential equations there results the following condition:

$$
\frac{\Delta\Psi^2}{6(1+N)} \left[\frac{Pr/Le\ \mu}{(\rho u)^{1-N}} \right]_{n,0} \geq \Delta x \leq \frac{1}{3} \frac{m^N\ \Delta\Psi^{2+N}}{(Le\ b/Pr)_{n,m+\frac{1}{2}} + (Le\ b/Pr)_{n,m-\frac{1}{2}}}
$$

Description of cooling techniques

Slot cooling

Describing equations: The mathematical model selected for implementation within the present program utilizes the Hatch-Pappel (Reference 6) correlation together with the Bartz (Reference 7) method for evaluation of the requisite heat transfer coefficient and the Sellers (Reference 8) procedure for including the effect of multiple slots. The manner in which the latter is accounted for is indicated schematically in Figure 9 where the various parameters pertinent to the slot cooling problem are defined. According to the analysis cited above, the wall temperature distribution between the Ith and the Ith+1 slot can be determined from*

$$\ell n\, \eta^I = (Pr_c R_c^I)^{1/8} \quad \Phi^I[.04 - (x-x_s^I)/G^I s^I] \tag{51}$$

where the effectiveness parameter is defined by

$$\eta^I \equiv (T_w^{I-1} - T_w^I)/(T_w^{I-1} - T_c^I) \tag{52}$$

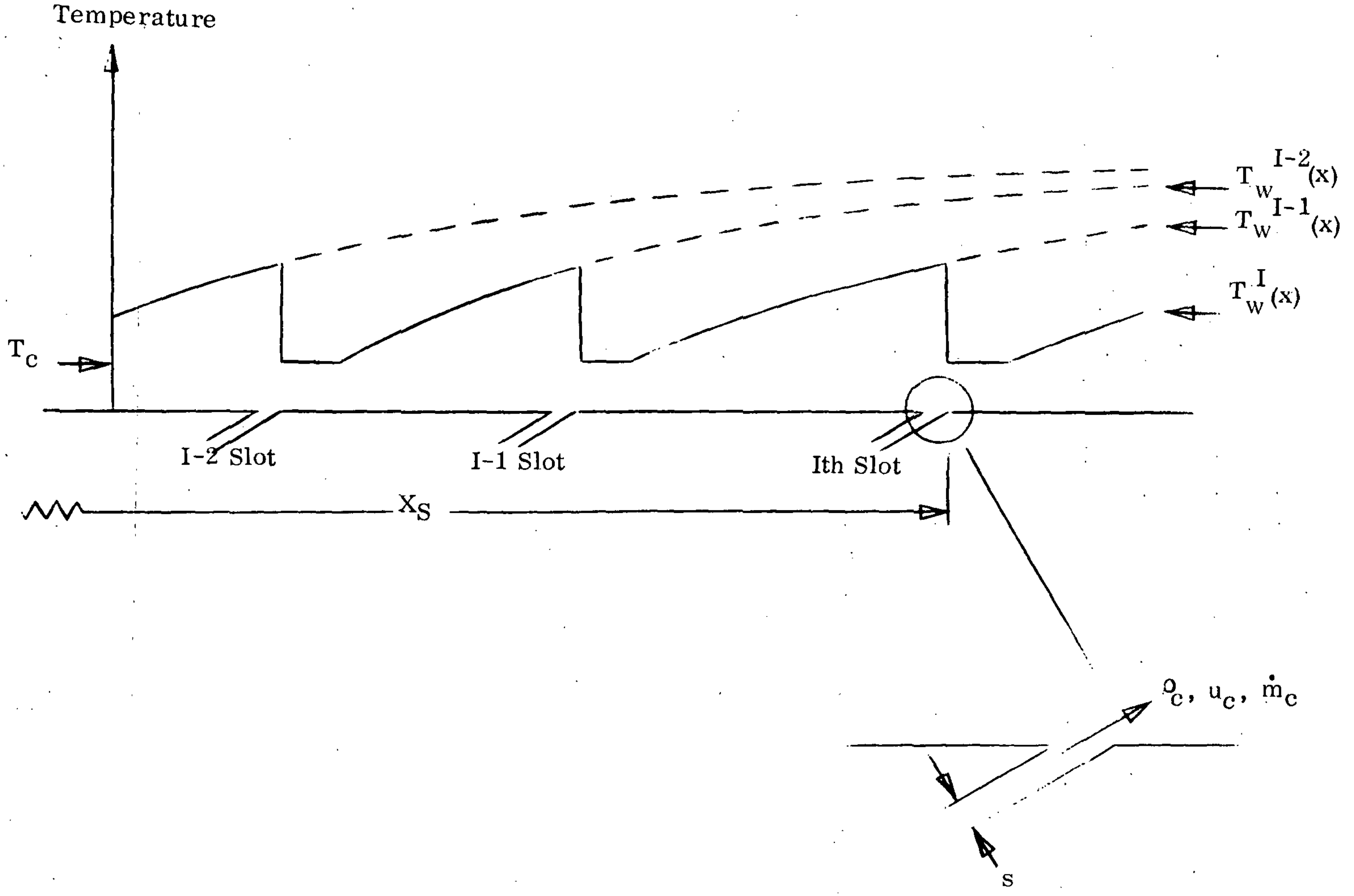
and

$$G^I \equiv (\rho u c_p)_c^I / h^I \tag{53}$$

$$\Phi^I = \begin{cases} (U^I)^{1/8}[1+0.4 \arctan(U^I-1)] & ;\ U^I \geq 1.0 \\ (U^I)^{13/8}(U^I)^{[1-1/U^I]} & ;\ U^I \leq 1.0 \end{cases} \tag{54}$$

*More precisely, these equations apply in the region bounded by $(x-x_S^I)/G^I s^I \geq .04$ and $x \leq x_S^{I+1}$. For axial locations upstream of this lower limit but beyond x_S^I the temperature is constant and equal to T_c^I as has been indicated in Figure 9.

FIG. 9



SCHEMATIC OF MULTIPLE SLOT COOLING CONFIGURATION

Following Bartz, the heat transfer coefficient h_g^I , appearing in Equation (53), is calculated from

$$h_g^I = \sigma^I \left(\frac{A^*}{A}\right)^{0.9} \left[\frac{.025}{(D^*)^{.2}} \left(\frac{c_p\mu^{.2}}{Pr^{.6}}\right)_b \left(\frac{p}{c^*}\right)^{.8} \left(\frac{D^*}{R^*}\right)^{.1}\right] \qquad (55)$$

where

$$\sigma^I = \left[\frac{1}{2}\left(1 + \frac{\rho_w^I}{\rho_b}\right)\right]^{.8} \left[\frac{1}{2}\left(1 + \frac{\mu_w^I}{\mu_b}\right)\right] \qquad (56)$$

In these relations, the subscript b indicates the bulk properties of the main thrust chamber. The manner in which these are evaluated is indicated in the next section.

For a single slot (or for the first slot) it is necessary to define more explicitly the significance of T_w^o . Consistent with the manner in which the Hatch-Pappel model was developed for single slots, we take T_w^o equal to the adiabatic wall temperature based on bulk properties $(T_{aw})_b$. In turn, this is related, in the manner which will be indicated in the next section, to $(H_{aw})_b$,

where

$$(H_{aw})_b = h_b + (Pr)_b^{1/3}\, u_b^2/2 \qquad (57)$$

Computational procedures: The slot height s^I, axial location x_s^I , unit mass flux rate of coolant $\dot{m}_c^I$, coolant reservoir temperature T_{tc}^I, slot exit pressure p_c^I, coolant Prandtl number Pr_c and specific heat c_{p_c} is considered to have been prescribed for all I so that all parameters with subscript c appearing in Equations (51) through (54) have been evaluated. At a generic point $x = x_n$ such that $x_s^I \leq x_n \leq x_s^{I+1}$ it is further assumed that all flow parameters throughout the thrust chamber have been determined. The problem is to determine the wall temperature at the next integration step $x = x_{n+1}$ which will be denoted by $(T_w)_{n+1}^I$. The computational procedure utilized to accomplish this (assuming $x_{n+1} < x_s^{I+1}$) is as follows:

(a) compute bulk properties of conservation variables at $x = x_n$ from

$$H_b = \frac{2\pi\int_0^{r_w} \rho u H r dr}{\dot{m}}$$

$$u_b = \frac{2\pi\int_0^{r_w}\rho u^2 r dr}{\dot{m}}$$

$$(\alpha_i)_b = \frac{2\pi\int_0^{r_w}\rho u\alpha_i r dr}{\dot{m}}$$

where r_w and $\dot{m}$ denote the values of thrust chamber radius and total mass flux through the motor at $x = x_n$, respectively.

(b) compute all other bulk thermodynamic properties such as μ_b, ρ_b, (T_{aw}) b, etc. from appropriate auxiliary thermodynamic relations, e.g.:

$$(T_{aw})_b = fcn\ [(H_{aw})_b\ ,\ (\alpha_i)_b]$$

where $(H_{aw})_b$ follows from Equation (57) and the functional notation implies the use of the internal enthalpy-temperature fits incorporated by the program.*

(c) determine the wall composition at $x = x_{n+1}$, $(\alpha_{iw})_{n+1}$, by application of the film cooling boundary condition for species diffusion (c.f. Table III).

Since

$$\frac{\rho_w^I}{\rho_b} = \frac{W_w}{\bar{W}_b}\frac{T_b}{T_w^I}$$

and

$$\mu_w = f_{cn}\ (T_w^I)$$

the only unknows appearing in Equation (51) are $(T_w)_{n+1}^{I}$ and $(T_w)_{n+1}^{I-1}$

*It is important to note here that μ_b represents a laminar (molecular) viscosity which depends on both the composition and temperature. Although more complicated representations for the viscosity of a mixture could be incorporated to evaluate this parameter, it has been deemed sufficiently accurate for the present purpose to ignore the dependence on composition and evaluate μ from the Sutherland formula for air; viz:

$$\mu_b = 2.27\ \frac{T_b^{3/2}}{T_b+198.6} \times 10^{-8}\ \frac{\text{lb.sec.}}{\text{ft}^2}$$

where T_b is in degrees Rankine.

But

$$\ln \eta^{I-1} = (Pr_c R_c^{I-1})^{1/8} \, \bar{\Phi}^{I-1} [.04-(x-x_S^{I-1})/G^{I-1} s^{I-1}]$$

$$\eta^{I-1} = (T_w^{I-2} - T_w^{I-1})/(T_w^{I-2} - T_c^{I-1})$$

etc., where the subscript n+1 has been omitted for clarity. Thus, the next step in the procedure is to

(d) systematically determine in order $(T_w)^1_{n+1}$, $(T_w)^2_{n+1}$ ..., $(T_w)^{I-1}_{n+1}$, $(T_w)^I_{n+1}$, by iterative solution of the transcendental Equation (51).

Solution of the last member in this sequence gives the desired value of $(T_w)^I_{n+1}$.

Transpiration cooling

Describing equations: The mathematical model selected for implementation within the present program uses the Bartle-Leadon (Reference 9) correlation together with a reference state approach to evaluate the requisite Stanton number. The reference state is defined by utilizing the Eckert (Reference 10) definition of reference enthalpy and Knuth's (Reference 11) representation for the reference composition. In accordance with these formulations, the temperature of the transpired surface is related to the coolant flow rate by the relation:

$$\frac{(T_{aw})_o - T_w}{(T_{aw})_o - T_c} = 1 - (1 + \frac{G}{3})^{-3} \qquad (58)$$

where

$$G \equiv \frac{1}{S_{t_o}} \frac{(\dot{m}c_p)_c}{(\rho u c_p)_e} \qquad (59)$$

and

$$(T_{aw})_o = \text{fcn}\,[(H_{aw})_o\,,\ \alpha_{ie}]$$

$$(H_{aw})_o = he + (Pr)^{1/3}\ \frac{u_e^{\,2}}{2}$$

Here the subscript o indicates properties evaluated under "no blowing" conditions, while subscript e indicates edge conditions. Evaluation of the latter is discussed later.

Use of a Reynolds analogy and reference state in conjunction with an incompressible skin friction law leads to the following representation for the Stanton number:

$$S_{t_o} = .0592\,(Pr)_e^{-2/3}\ \left(\frac{\rho^*}{\rho e}\right)\ \left(\frac{\rho^* u_e x}{\mu^*}\right)^{-1/5} \qquad (60)$$

where the asterisk implies that the thermodynamic variables are to be evaluated at the reference state condition. The latter is defined by combining Eckert's form for the reference enthalpy

$$h^* = \frac{1}{2}\ (H_w + h_e)\ +\ 0.22\,(H_{aw_o} - h_e) \qquad (61)$$

with Knuth's representation for the reference composition:

$$\alpha_i^* = \alpha_{i_e}\ \left(\frac{\bar{\alpha}^*}{1-\alpha_{c_e}}\right) \qquad i \neq c$$

$$\alpha_c^* = 1 - \bar{\alpha}^* \qquad (62)$$

where

$$\bar{\alpha}^* = \left(\frac{\bar{W}_e}{\bar{W}_e - W_c}\right)\ \frac{\ell n\,(W_e/W_w)}{\ell n\,[\,(1-\alpha_{c_e})W_e/(1-\alpha_{c_w})W_w\,]} \qquad (63)$$

$$\bar{W}_e = (1-\alpha_{c_e})\,(\sum_{i\neq c} \alpha_i/W_i)_e^{-1}$$

$$W_w = (\sum_i \alpha_i/W_i)_w^{-1}$$

We note here that the parameter $\bar{W}_e$ represents the molecular weight of the subsystem consisting of all molecular species except the injected species (coolant) at the edge of the boundary layer.

Computational procedure: The coolant reservoir temperature $T_c(x)$ is considered to be given and either the coolant mass flux rate $\dot{m}_c$ (x) (Option A) or the wall temperature $T_w(x)$ (Option B) are prescribed. Assuming that all flow properties throughout the thrust chamber are known at the generic point $x = x_n$, the problem is to determine either T_w at $x = x_{n+1}$ or $\dot{m}_c$ at $x = x_{n+1}$. From the point of view of integrating the describing equations, these two options differ in a very fundamental way. Thus, for Option A ($\dot{m}_c$ specified; T_w to be determined), Equation (58) is directly coupled to the system of finite difference equations which are to be integrated. In contrast, for Option B (T_w specified; $\dot{m}_c$ to be determined) the integration could proceed independently of whether or not a solution of Equation (58) is obtained. The procedure used in the case of Option A will be described first.

Option A

(a) Compute "edge" properties at $x = x_n$. Provision has been made for evaluation of these edge properties in two different ways. One of these is identical to the procedure used for the film cooling model wherein the edge conditions are equated to the bulk properties of the conservation variables. The alternate procedure involves determination of the edge of the boundary layer by establishing the extent to which the tracer gas has penetrated laterally.* At this point in the flow field, the values of H, u and α_i can be determined from which all parameters appearing in Equations (58) through (63) with subscript e, can be evaluated.

(b) Determine the wall composition at $x = x_{n+1}$, $(\alpha_{i_w})_{n+1}$, by application of the transpired wall boundary condition for species diffusion (c.f. Table III). Since $\dot{m}_c$ at $x = x_n$ is known, this can be readily accomplished.

(c) Determine the reference composition $\alpha_i{}^*$ at $x = x_{n+1}$, from Equations (62) and (63). Note here that in this determination we are "lagging" on the edge condition.

(d) Determine T_w at $x = x_{n+1}$, by iterative solution of Equation (58). Note that this is required since the parameter G depends implicitly upon T_w, via Equations (59), (60) and (61) and the fact that H_w is a function of α_{i_w} and T_w at $x = x_{n+1}$. With T_w and, therefore, H_w determined at $x = x_{n+1}$, all data needed to continue the integration of the conservation equations are available.

Option B

For Option B, steps (a) through (c) are identical. However, since T_w is known at $x = x_{n+1}$, H_w at that point can be evaluated directly which allows h* and therefore ρ^* and μ^* to be evaluated, using Equation (61). The only unknown appearing in Equation (58) therefore is the mass flux rate $\dot{m}_c$ at $x = x_{n+1}$ which can be solved for in a straightforward manner, if desired.

*The criterion used to determine this penetration is to locate the point where $\alpha_T \leq .01\, \alpha_{T_W}$.

Performance model. - The output of the combustion model provides the characteristics of the gases at the throat of the rocket engine in terms of mixture ratio and enthalpy along each streamline. The data thus obtained can be used to determine the specific impulse and thrust of the engine as a function of the nozzle characteristics. The output of the combustion model can be used as input to the various CPIA standard programs (Reference 1), but because of the complexity and relatively long run time of these programs, a more simplified method was used to determine the losses in the nozzle. The losses are due to:

(1) Kinetic recombination
(2) Boundary layer or viscous effects
(3) Divergence
(4) Film and dump cooling

The performance of the rocket engine can be estimated using approximate methods as described in Reference 1 for the boundary layer or viscous losses and the divergence losses, while the kinetic recombination losses are determined for each streamline using the methodology outlined in Reference 12. The enthalpy and mixture ratio at each streamline is used as the input to determine the specific impulse after kinetic loss. The characteristics of the streamline are used in determination of the viscous loss, while the loss due to divergence is based solely on the area ratio and an infinite pressure ratio ($\gamma = 1.20$). The losses due to kinetics and mixture ratio are thus mass averaged and the viscous and divergence losses subtracted directly. In addition, any hydrogen or oxygen dumped in the expansion bell is assumed not to mix and the I_{sp} of the cold gas is assumed to apply. In summary, the equation used to obtain the performance is

$$I_{sp} = \frac{\Sigma\left(I_{spK_i}\right)\left(\dot{w}_i\right)}{\Sigma\left(\dot{w}_i - \dot{w}_c\right)} - \left(I_{sp_S}\right)\left(1 - \eta_D\right) - \left(\frac{\Delta F_{BL}}{\dot{w}_T} + \frac{I_{sp_C}}{\dot{w}_c}\right) \qquad (64)$$

Kinetic recombination: The method used to determine the amount of recombination which takes place in the nozzle for each streamline is based on an approximate method developed by United Aircraft Research Laboratores and is described in Reference 12. The losses are evaluated using the Modified Bray criterion for predicting the point in the recombining nozzle flow where the reactions have departed significantly from equilibrium. This criteria has been successfully used to analyze flow in which only one reaction is kinetically important. The Bray method was extended to multicomponent-multireaction performance calculations and the method was incorporated in the performance model. The basic method used to determine the kinetic loss for each streamline is as described below.

(a) The specific impulse for each streamline based on total pressure, enthalpy, and mixture ratio in both equilibrium and frozen flow is determined.

(b) Based on the nozzle contour and design, the amount of recombination at mixture ratios of 4,6, and 8 are determined at the known pressure.

(c) Assuming that the frozen and equilibrium specific impulse below a mixture ratio of 2 are identical and that the percent of recombination above O/F = 8 is the same as O/F = 8, the data for the recombination in (b) was linearly interpolated for each streamline.

The data presented in Reference 12 was based on liquid O_2 and H_2. In order to make the data adaptable to propellant at any temperature, the data was modified to the form of percent recombination of

$$\eta_R = \frac{I_{sp_K} - I_{sp_F}}{I_{sp_S} - I_{sp_F}}$$

Thus, if the equilibrium and frozen specific impulse at a known enthalpy and mixture ratio are known, the kinetic data can be readily determined.

Figures 10, 11, and 12 show typical data used in the determination of the kinetic I_{sp}. The nozzle design area ratio is obtained from the Figure 10 based on the dimensional characteristics. The nozzle gradient (See Figure 11) is then determined for various area ratios (in the case of this analysis from 1.01 to 5.0). The gradient value is then divided by the nozzle throat diameter and multiplied by the line labeled "Equilibrium" obtained from Figure 12. This data of H atom gradient vs. area ratio is then cross-plotted against curve labeled "Composite Kinetic (Transition Factor = 1)" of Figure 12. The intersection is the freeze area ratio. In the reference, the kinetic I_{sp} is found directly from a plot of I_{sp} vs. freezing area ratio (freeze) (see Figure 13), but as described previously, the data was modified to yield percent recombination.

Boundary layer or viscous losses: The losses due to the interaction between the gas and the wall, commonly called boundary layer losses, are determined using the approximate methods described in Reference 1. This method is based on the results of a more rigorous calculation method, commonly called TBL (turbulent boundary layer). Data for the various characteristics of the boundary layer, momentum thickness, for example, have been presented in terms of the isentropic exponent, the temperature ratio (gas-to-wall), and Mach number. In the present analysis, the isentropic exponent, γ, was assumed to be 1.2 and the temperature ratio 0.2, since a majority of the wall temperatures are in the 500-2000°F range while the gas temperature is >6,000°F. Figures 14, 15, 16, and 17 show the data used. The effect due to variations in contraction ratio and radius of curvature was assumed constant since the effect is second order. The equation thus used to determine the loss in therms of thrust is

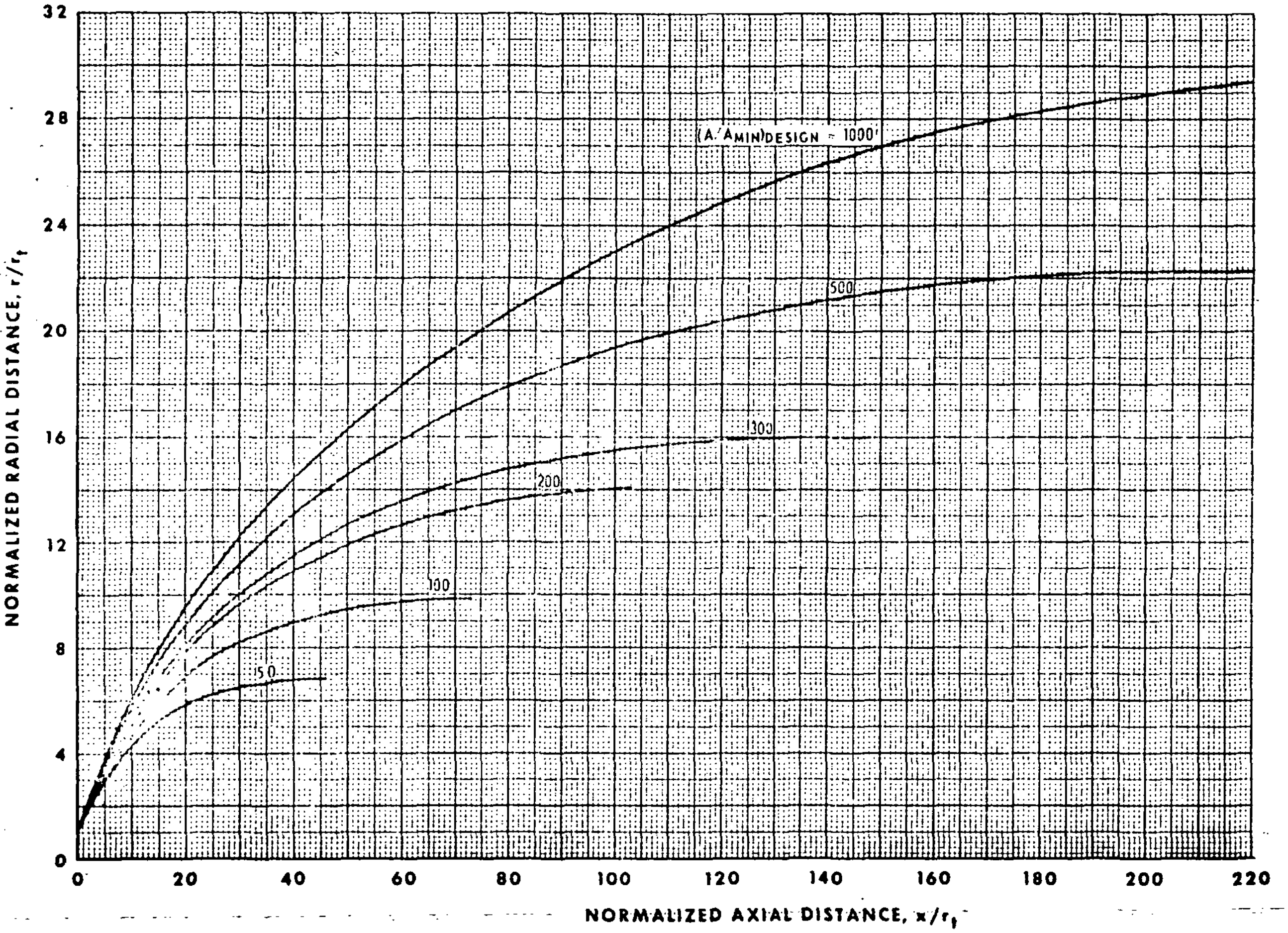
$$\Delta F = 2\pi\, r_e P_o \left(\left(\rho_e u_e^2/P_o \right) \theta_e \cos\alpha_e \right)^{\swarrow\text{Fig. 14}} \left(1 - \left(P/\rho u^2 \right)_e \overset{\swarrow\text{Fig. 15}}{\left(\delta^*/\theta \right)_e} \right) \qquad (65)$$

FIG. 10

AXISYMMETRIC PERFECT NOZZLE CONTOURS

SPECIFIC HEAT RATIO = 1.25

(REF 12)



NOZZLE AREA RATIO GRADIENTS OF SELECTED PERFECT NOZZLE CONTOURS (REF 12)

SPECIFIC HEAT RATIO = 1.25

$r_c/r_t = 1.0$

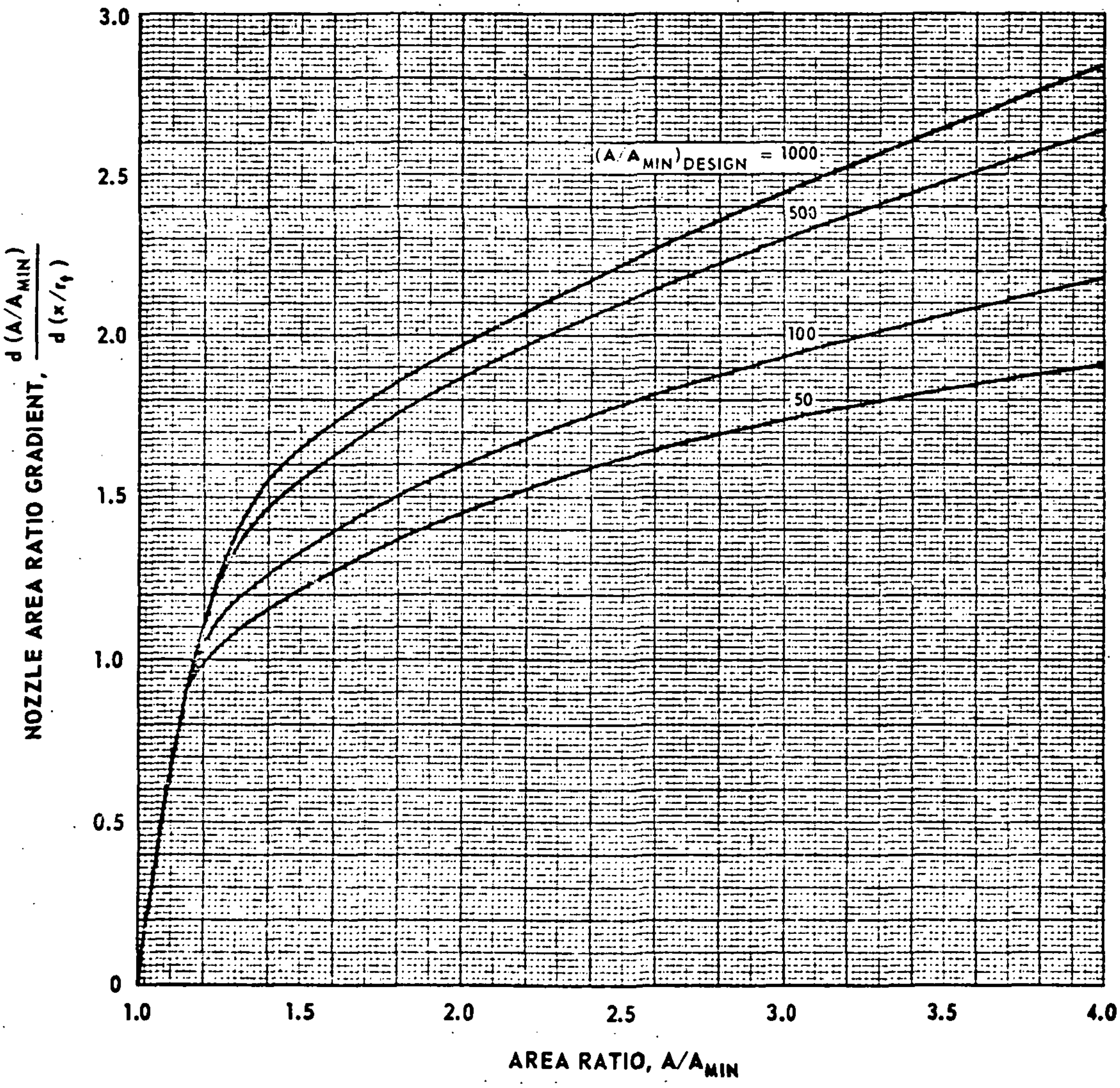


FIG. 12

NORMALIZED GRAPHICAL SOLUTION FOR FREEZING AREA RATIO USING MODIFIED BRAY ANALYSIS (REF 12)

$H_2(\ell) - O_2(\ell)$

P_C = 500 PSIA (3.448×10^6 N/m^2)

O/F = 6.00

NOTE: REACTION RATE CONSTANTS LISTED IN TABLE II-1

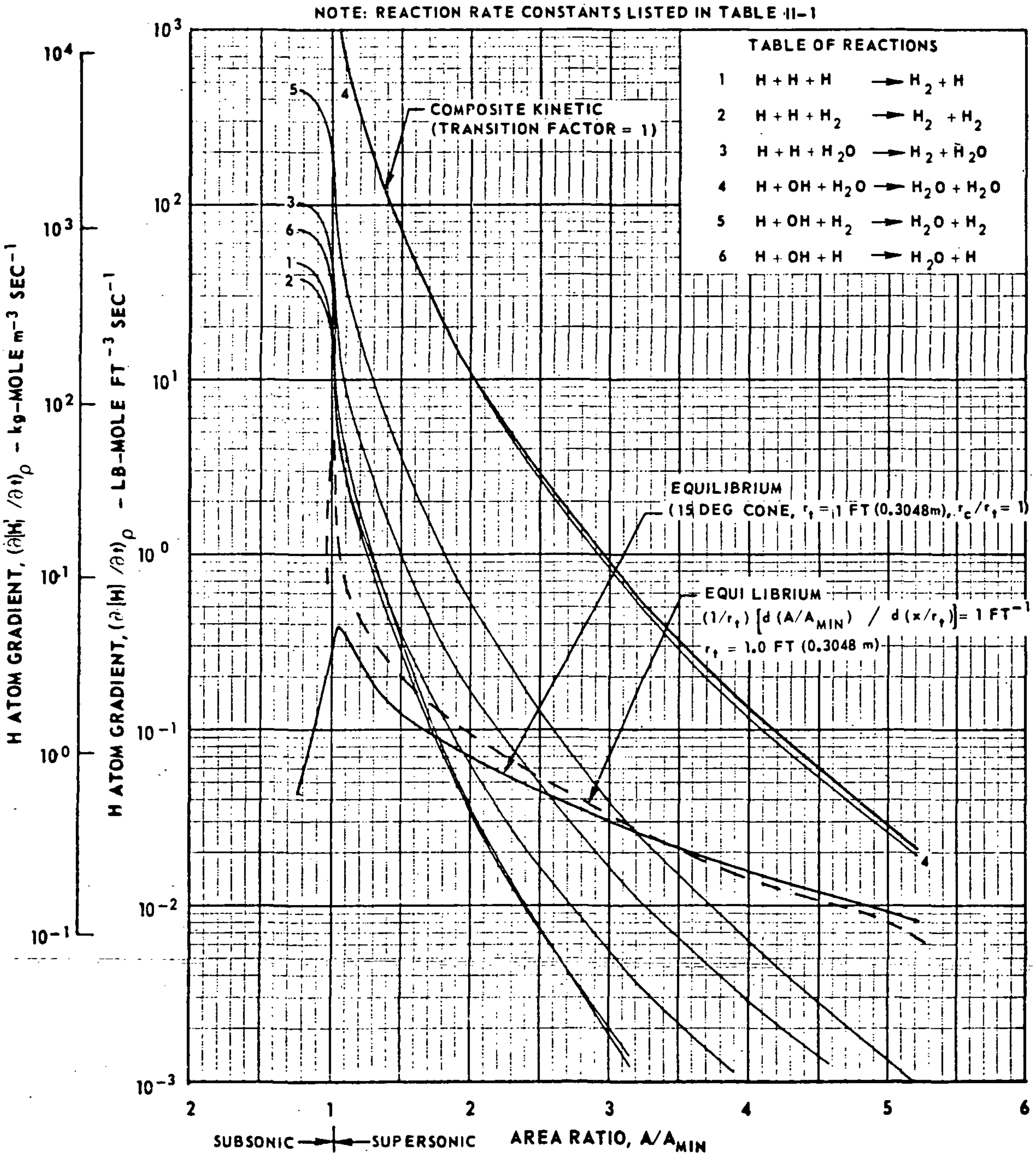


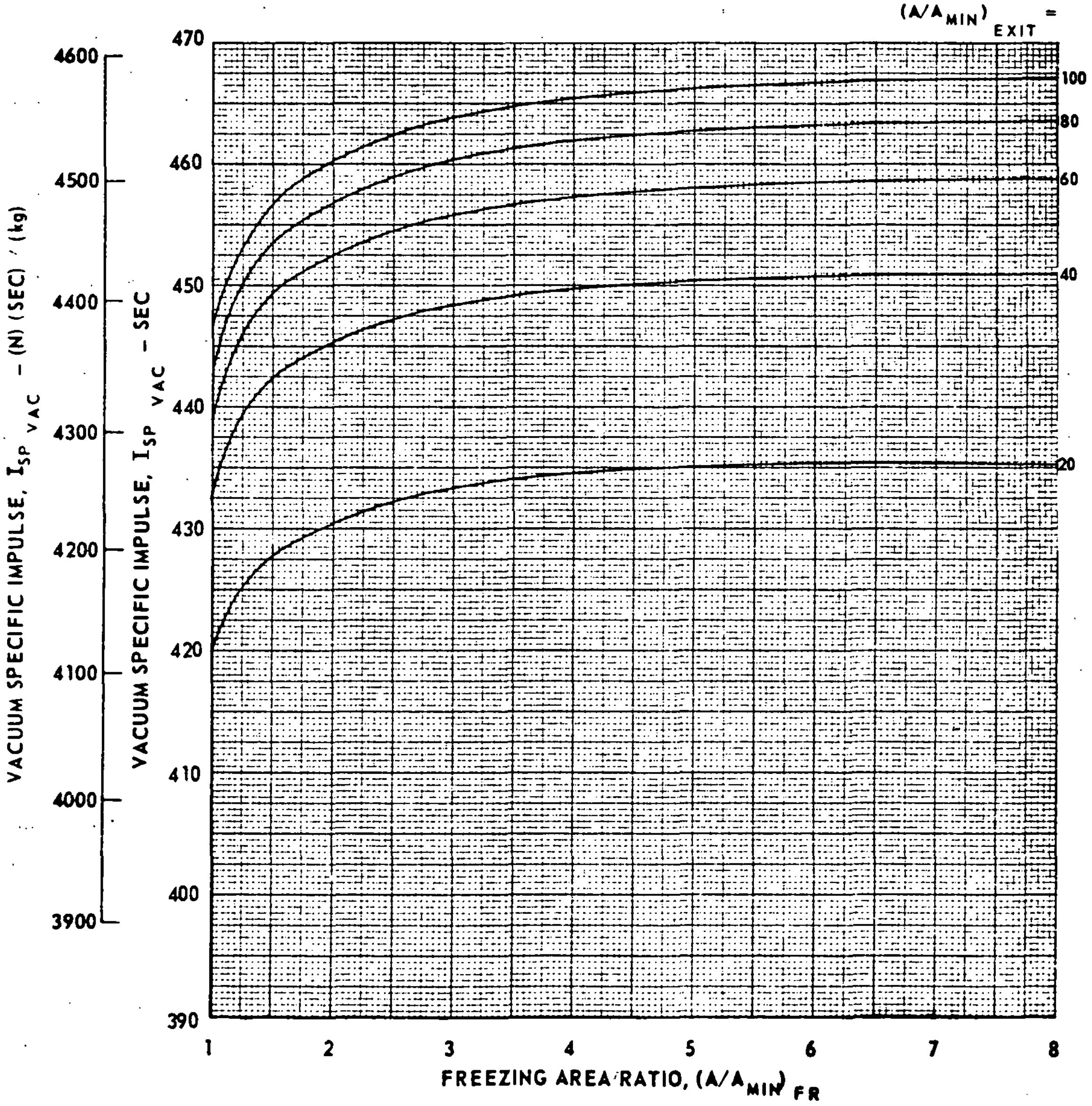
FIG. 13

EFFECT OF FREEZING AREA RATIO ON NONEQUILIBRIUM PERFORMANCE FOR HYDROGEN-OXYGEN PROPELLANT SYSTEM (REF 12)

$H_2(\ell) - O_2(\ell)$

P_C = 500 PSIA ($3.448 \times 10^6 N/m^2$)

O/F = 6.00



EFFECTIVE NOZZLE WALL ANGLE FOR CONTOURED NOZZLE

$\gamma = 1.2$

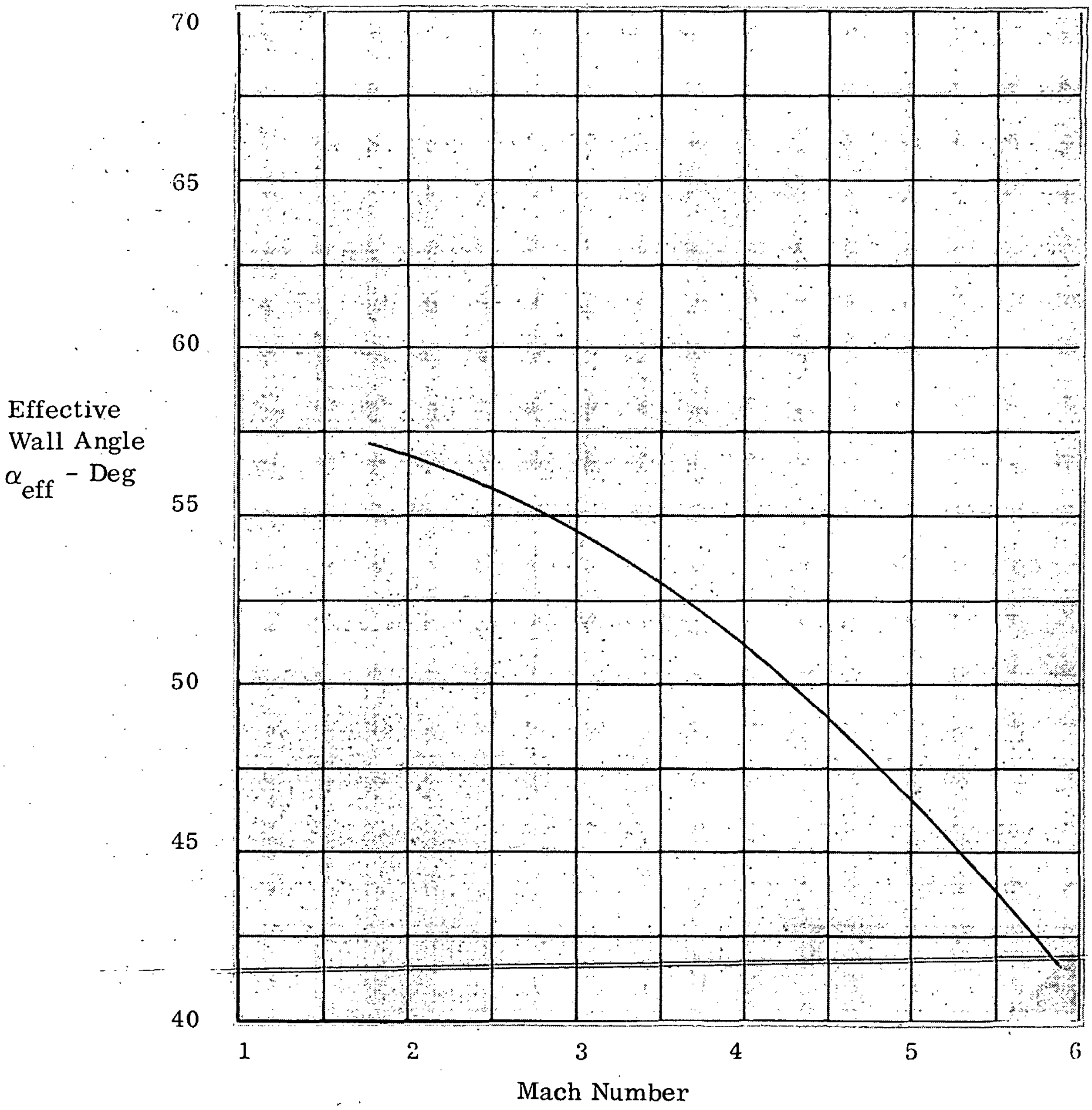
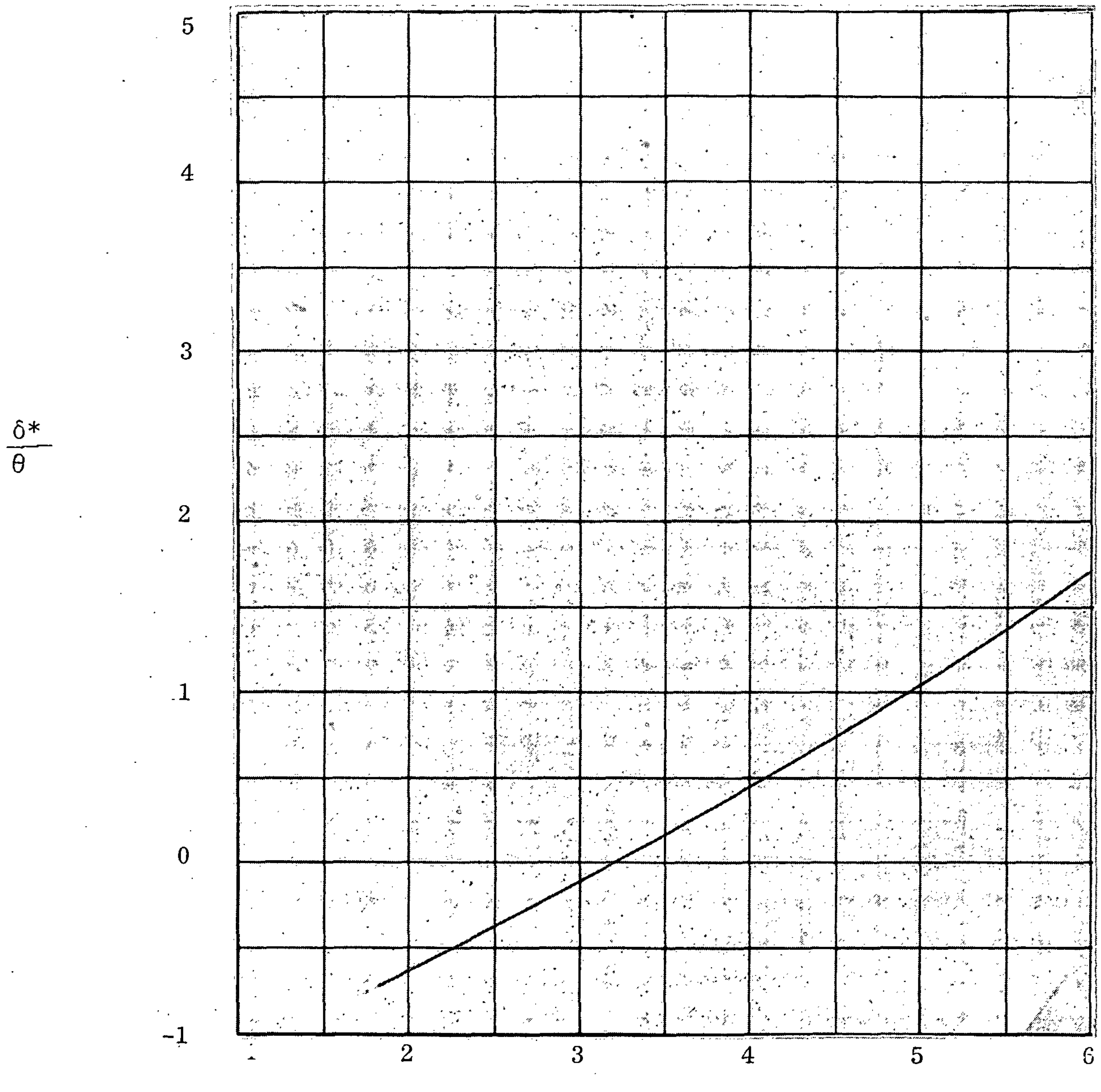


FIG. 15

BOUNDARY LAYER SHAPE FACTOR

γ = 1.2

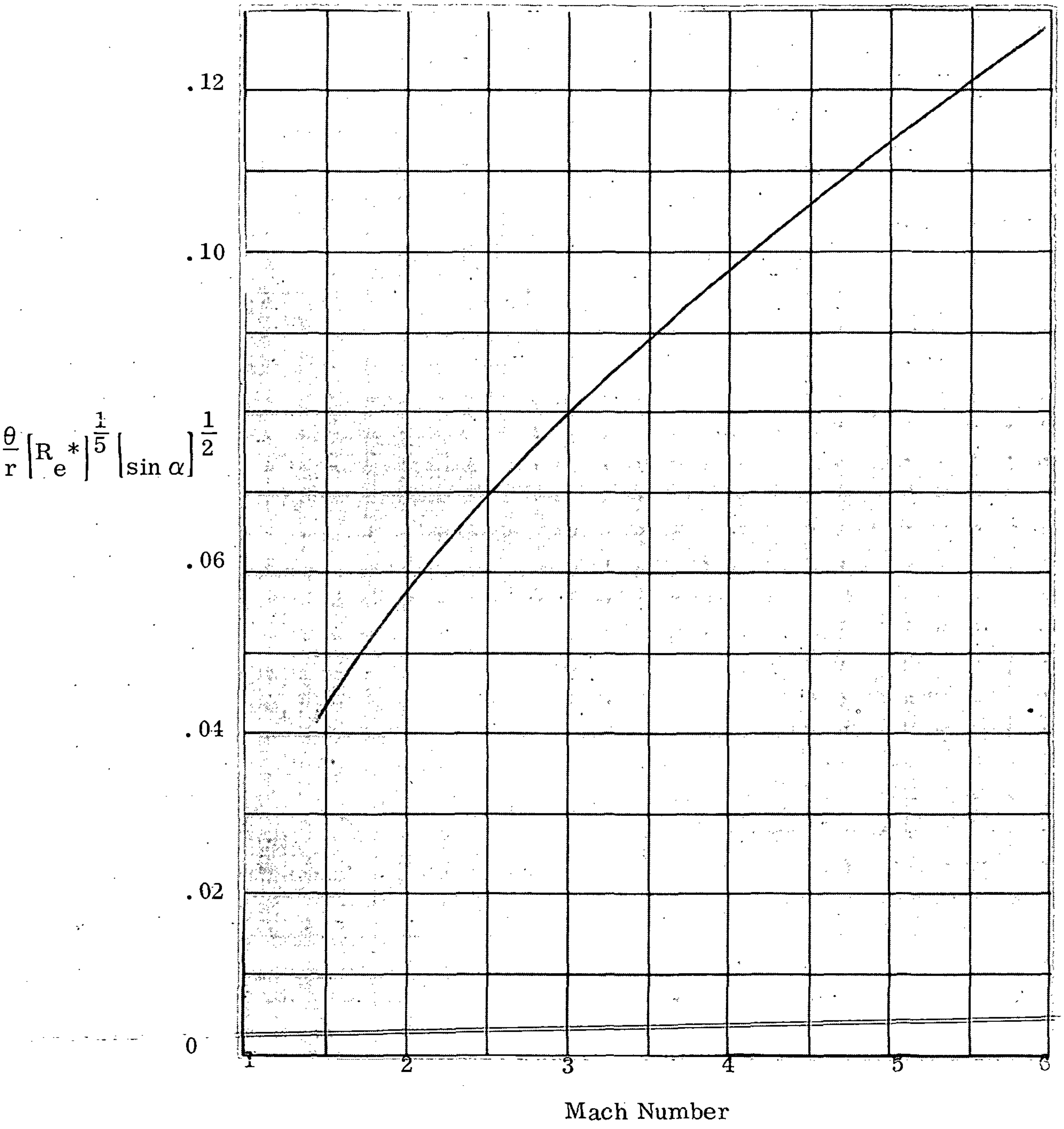


Mach Number

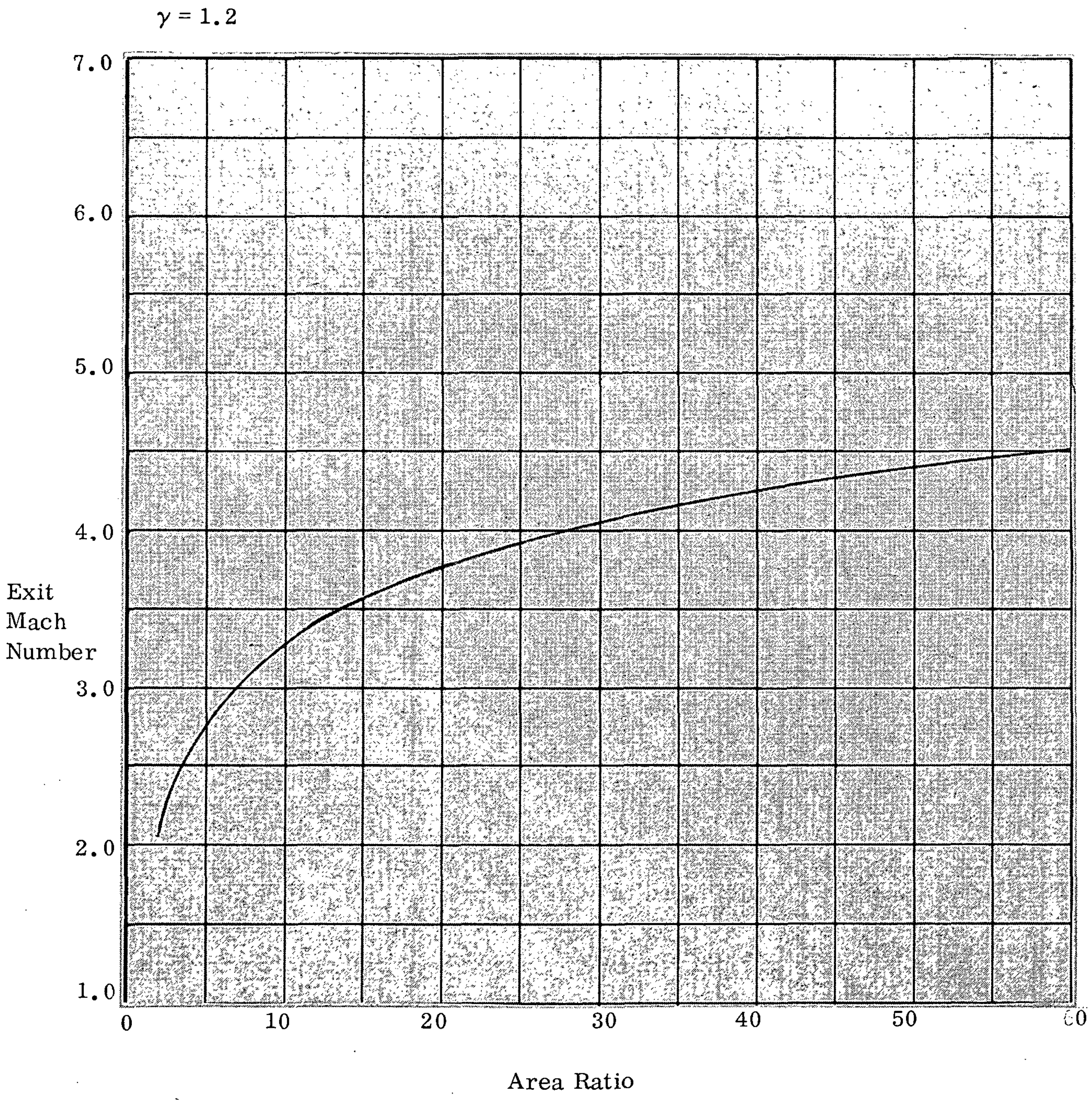
FIG. 16

GROWTH OF THE BOUNDARY LAYER MOMENTUM THICKNESS

$\gamma = 1.2$



NOZZLE 1 - DIMENSIONAL EXIT MACH NUMBER VS. AREA RATIO



where $\frac{P}{\rho u^2} = \left(\frac{1}{\gamma M^2}\right)_e$ and

$$\frac{\rho_e u_e^2}{P_c} = \left(\gamma M^2\right)_e / \left[1 + \frac{\gamma-1}{2} M^2\right]_e^{\frac{\gamma}{\gamma-1}} \qquad (66)$$

$$\theta_e = \left[\text{Fig. } 17\right] \left(\frac{1}{R_e}\right)^{.2} r_e \underset{\nwarrow\text{Fig. } 14}{\frac{1}{(\sin \alpha)}}{}^{.5}$$

Dividing by the total mass flow results in I_{sp} loss. The gas properties were taken to be those of the streamline adjacent to the wall.

Divergence: The loss due to divergence was obtained from Reference 1 and is shown in Figure 18. The pressure ratio was assummed to be infinite.

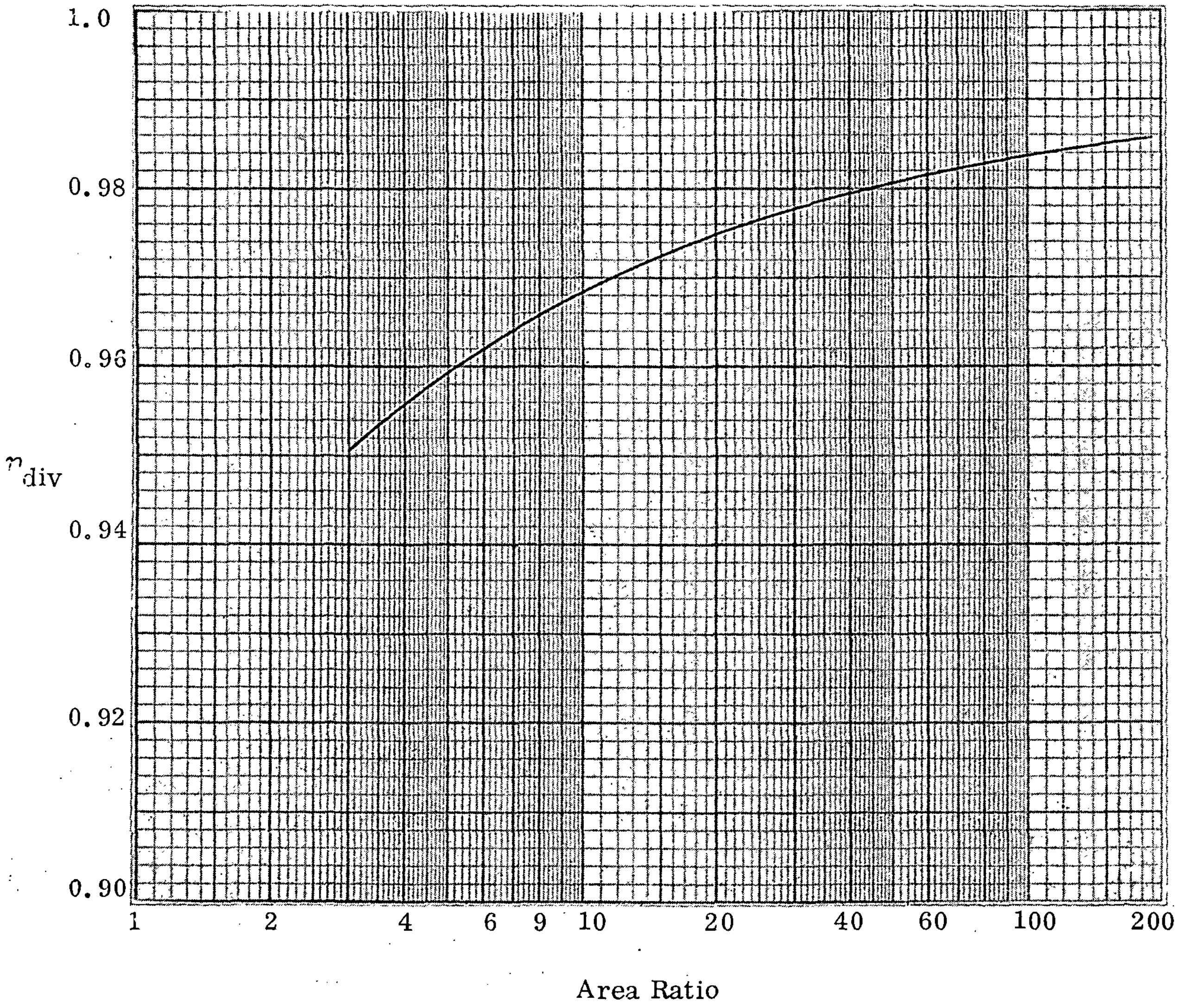
Mixture ratio maldistribution effects: The gas properties at each grid point from the combustion program must be integrated to find an average kinetic specific impulse as described by the first portion of equation 64. The $\dot{w}_i$ for each streamline is a point function which is applied to a flow streamtube. The method used is simply one of linear interpolation between streamlines. For example, the following data

Streamline	Radius	Area (i) of Streamtube
1	$r_1 = 0$	$\pi \left(\left(r_2\right)/2\right)^2$
2	r_2	$\pi\left[\left(\left(r_3 + r_2\right)/2\right)^2 - \left(\left(r_1 + r_2\right)/2\right)^2\right]$
i	r_4	$\pi\left[\left(\left(r_{i+1} + r_i\right)/2\right)^2 - \left(\left(r_{i-1} + r_i\right)/2\right)^2\right]$
⋮	⋮	⋮
n	r_n	$\pi \left(\left(r_n + r_{n-1}\right)/2\right)^2$

FIG. 18

DIVERGENCE EFFICIENCY FOR CONTOURED NOZZLES

$P_e/P_o = 0$
$\gamma = 1.2$
from Ref. 1



where r_n = wall radius, will, when combined with the individual $(\dot{w}/A)_i$ from the combustion program provide the $\dot{w}_i$ required for equation (64).

Heat transfer model. - The heat transfer model predicts combustion, injector, and coolant temperatures for a generalized GO_2/GH_2 rocket engine. The cooling methods which can be used are shown in Figure 19. The user specifies combustor geometry and heat transfer properties, engine operating conditions, and, where applicable, coolant injection conditions. The heat transfer subroutine creates from the input a thermal model of the engine and solves for temperature distribution using a finite difference technique. The heat transfer model is completely unconnected to other characterization program models. It exchanges no information with other models. Unlike the other characterization subroutines, heat transfer input is accepted in the subroutine itself, not passed from the driver program.

The thermal model created by the heat transfer routine consists of mass nodes connected by thermal resistances analogous to an electrical network containing capacitors and resistors. A simple example is given in Figure 20 showing a bar heated by convection, cooled by radiation, and exhibiting internal conduction. It could represent a portion of film cooled combustor wall. Figure 20 also shows how the thermal admittances (admittance = (1 ÷ resistance)) are calculated for conductive, convective, and radiative heat transfer and how the thermal capacities are defined for mass nodes and for boundary nodes. Surface areas, cross-sectional areas, lengths, volumes, view factors, and convective coefficients are automatically computed by the heat transfer sub-program in the process of creating the thermal model.

Heat transfer solution technique: The basic equation describing the continuity of heat flow at node i is

$$\sum_{\substack{\text{all } j\\ \text{connected}\\ \text{to } i}} \left[A_{D_{i,j}} \left(T_j - T_i\right)\right]_t = \frac{C_i\left(T_{i_{t+\Delta t}} - T_{i_t}\right)}{\Delta t} \qquad (67)$$

which can be rearranged to give $T_{i_{t+\Delta t}}$ explicitly.

$$T_{i_{t+\Delta t}} = \frac{\Delta t}{C_i} \sum_{\substack{\text{all } j\\ \text{connected}\\ \text{to } i}} \left(A_{D_{i,j}} T_{j_t}\right) + \left(1 - \frac{\Sigma A_{D_{i,j}} \Delta t}{C_i}\right) T_{i_t} \qquad (68)$$

HEAT TRANSFER MODEL

COOLING METHODS

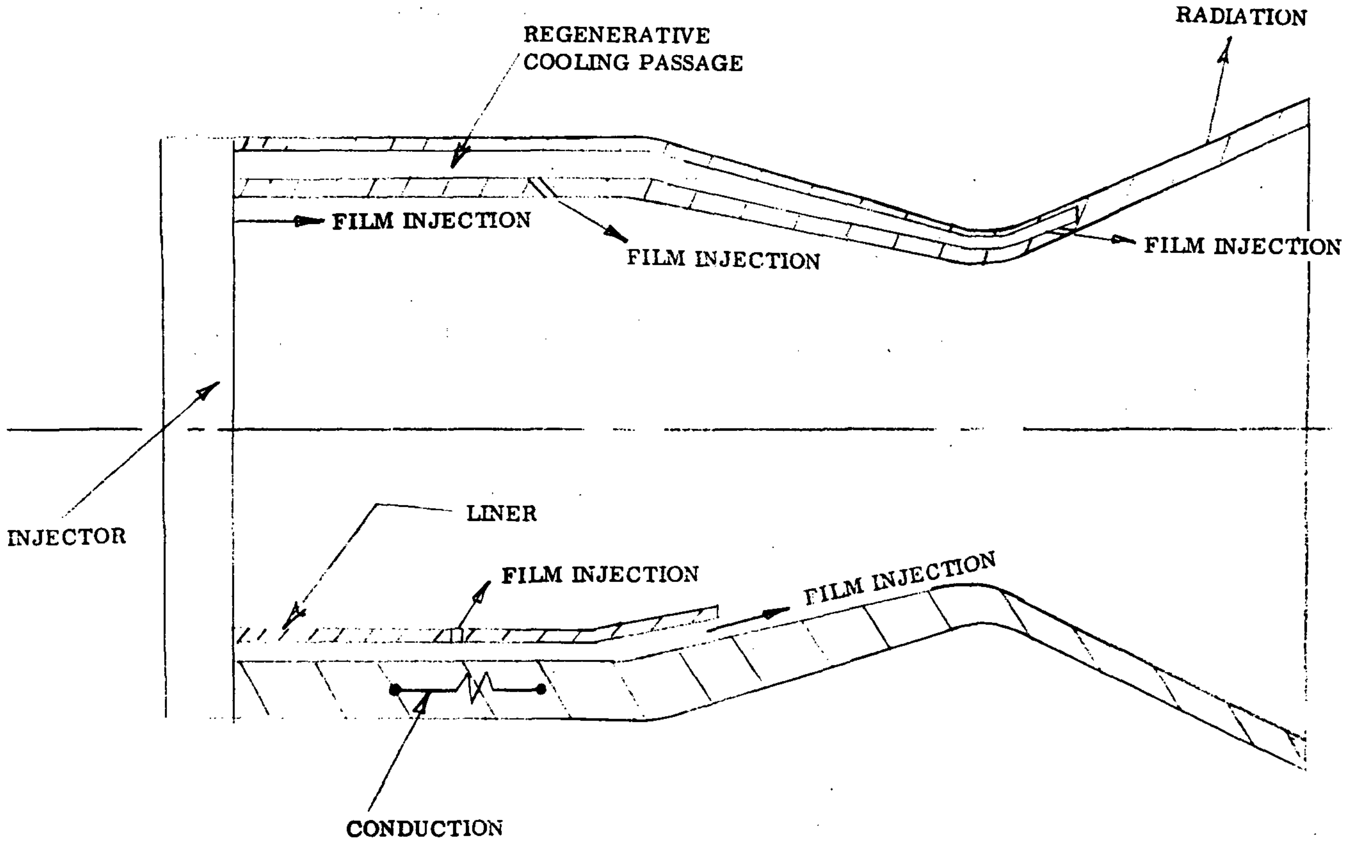
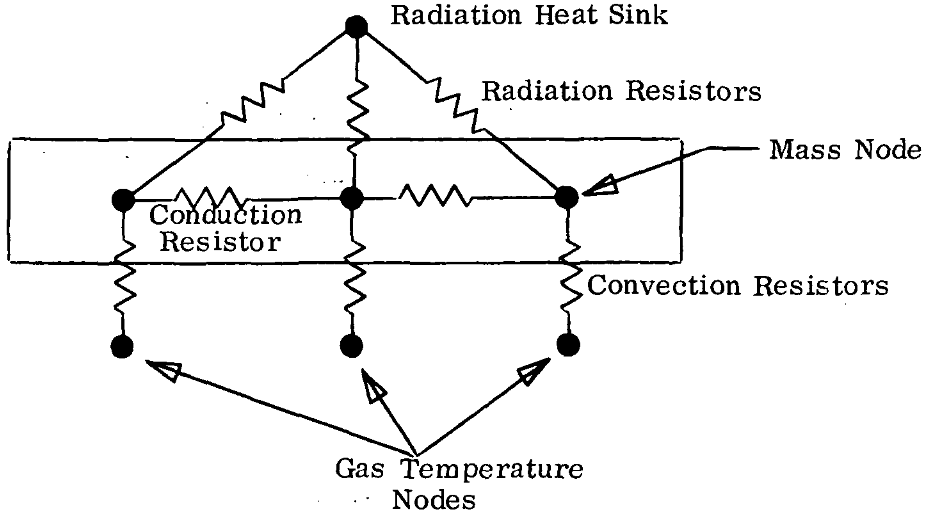


FIG. 20

HEAT TRANSFER MODELING



Conductive Admittance = $\frac{KA}{\ell}$

Convective Admittance = $h_g A$

Radiative Admittance = $A_s F\sigma \left(T_1^3 + T_1^2 T_2 + T_2^2 T_1 + T_2^3\right)$

Node Capacity = $\rho C_p V$ (Mass Nodes)

= 0 (Boundary Nodes)

In other words, the temperature of node i at time (t + Δ t) depends upon the weighted average of the temperatures of nearby nodes and the temperature of i at time t.

Except for the coefficient of T_{i_t}, the weights are inherently positive. In order to assure solution stability, all T_{i_t} coefficients are kept non-negative by selecting time increments, Δ t, such that

$$\Delta t \leq \frac{C_i}{\sum\limits_{\substack{\text{all } j \\ \text{connected} \\ \text{to } i}} A_{D_{i,j}}} \qquad \text{for all } i \tag{69}$$

For steady state calculations, the temperature at node i is simply:

$$T_i = \sum \frac{A_{D_{i,j}}}{\sum A_{D_{i,j}}} T_j \tag{70}$$

$$\text{all } j \text{ connected to } i$$

Calculation of passive cooling heat transfer admittances: Hot side convective coefficients are calculated using the method of Bartz (Reference 7) shown in Equation (55).

Viscosity is calculated using Sutherland's formula

$$\mu = (46.6 \times 10^{-10}) \sqrt{MW} \; (T_o)^{.6} \quad \text{lbm/in sec}$$

Prantl number is calculated using the approximate result

$$Pr = \frac{4\gamma}{9\gamma - 5} \tag{71}$$

Specific heat, Cp, is utilized in the form of a polynomial curve fit of rocket performance data evaluated at a mean film temperature.

Radiation view factors are assumed to be 1.0 on the external surfaces of the engine. The inside of the combustor is assumed to have a negligible view of the engine surroundings. For the purpose of calculating radiation view factors, the inside surface of the expansion bell is assumed to have approximately the shape of a 15° half angle cone.

The local view factor is given by a polynomial curve fit of previously calculated view factor data for 15° cones.

$$\mathcal{F} = \mathcal{F}\,(L/D^*\,,\ X/L)$$

Active cooling models

Regenerative cooling: The coolant side convective heat transfer coefficient is calculated using the equation developed in Reference 13. It is assumed that only hydrogen is used as a coolant. The coefficient is given by

$$h_g = .048\,\frac{k}{D_H}\,Re_b^{.8}\,Pr_b^{.4}\left(\frac{T_w}{T_b}\right)^{-.55} \qquad (72)$$

where:

$$Re_b = \frac{\dot{w}\,D_H}{A\,\mu}$$

Hydrogen conductivity and viscosity are determined using polynomial curve fits of Reference 13 results with bulk coolant temperature as the independent variable.

Coolant temperature rise is calculated using heat transfer resistances connecting the fluid nodes and given by $Res = 1/\dot{w}C_p$. This "heat flow" model is modified to prevent upstream node temperatures from being affected by downstream nodes, much like a cathode follower in electronics. The coolant temperature rise, then, is calculated using the same modeling framework shown in Figure 20.

Film cooling: The film cooling model is based upon experimental and analytical results given in Reference 6 for a variety of injection conditions and described in the section on the combustion model.

Coolant velocity is determined using input parameters and mass continuity. Core gas velocity is calculated from core Mach number which is determined by solving

$$\frac{A}{A^*} = \frac{1}{M}\left[\left(\frac{2}{\gamma+1}\right)\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{\frac{\gamma+1}{2(\gamma-1)}} \qquad (73)$$

for Mach number by Newton's method. The ratio of specific heats, γ, is available as a polynomial equation based on equilibrium chemistry results.

The film cooling model is used for multiple injection cases by using the film temperature produced by upstream injection as driving potential for films injected downstream. Calculations for the film injected farthest upstream are carried to the nozzle exit using core temperature as the driving potential. Calculations for the film injected next downstream are based upon the film temperatures resulting from the first film. Temperature distribution for a third film is influenced by the two upstream films, and so forth.

Liner cooling: The liner cooling model is designed to predict coolant temperature rise, and combustor temperature distribution. The liner is assumed to be a thin metal structure which delivers hydrogen coolant downstream. It is cooled by convection and radiation on the backside, and, in addition, may be film cooled on the combustion gas side.

Backside convective coefficients are calculated using equation (72). Since liners are sometimes run quite hot, radiative heat transfer to the combustor is important. The effective emissivity between liner and combustor is given by:

$$\epsilon = \frac{1}{\frac{1}{\epsilon_{liner}} + \frac{1}{\epsilon_{combustor}} - 1} \qquad (74)$$

which accounts for reflection back to the liner.

Like the regenerative coolant temperature rise model, the backside liner flow model is modified to allow "heat flow" only in the downstream direction. The liner effluent forms a cooling film for the combustor downstream of the liner. This film is treated exactly as described in the film cooling section.

Combined cooling models: The basic cooling models may be combined in a number of ways to describe complicated cooling schemes. The radiation and chamber conduction heat transfer modes are always in effect unless deactivated by setting emissivity or conductivity equal to zero. Film cooling may be used in conjunction with regenerative cooling or liner cooling. There is no provision for combination liner and regenerative cooling nor for transient calculations with active cooling.

Injector heat transfer model: The program contains an option for including a thermal model of the injector. Due to the vast number of possible injector configurations, most heat transfer information is required as input including heat transfer coefficients, surface areas, face heat flux, injector weight, injector-combustor seal resistance, etc. The model considers heat input to the injector from face convection, conduction from the combustor, and radiation from the combustor. Cooling is by radiation and by convection to the flowing propellants. The injector option may be used in both steady state and transient operation and in conjunction with any of the active cooling models.

SAMPLE CASE RESULTS

The characterization program was tested using an engine for which test data was currently available. This engine was fabricated and tested by the Aerojet Corporation and the results were presented in Reference 2. The results of the test case indicated that

a. The performance and dynamic response of the engine could be simulated by the computer program

b. The heat transfer analysis was adequate for predicting wall and coolant temperatures..

Dynamics and Ignition

The Aerojet engine used in the dynamic analysis is defined in Figures 21-23. The initial conditions and the calculated volumes and valve response are shown in Figure 2. The sample case output is shown in Appendix B. The plotted results are shown in Figure 24. The response of the actual engine and the output from the computer simulation are essentially identical.

Injection, Combustion, and Peformance

The results of the performance calculations which use the output of the combustion and injection model indicate that the methods used to analyze performance can predict the specific impulse within the accuracy required. The results of the study are shown in Figure 25 where data from Reference 2 is plotted against the output of the sample case.

Heat Transfer

The heat transfer program was used in a comparison of the engine cooling scheme shown in Figure 26. The engine which is both regenerative and slot film cooled has not been tested and as a result, the comparison could only be with the analytical results reported in Reference 2, shown in Figure 27. As shown in Figure 28, the results of the heat transfer computer program are essentially compatible with the results of Reference 2.

FIG. 21

IMPINGING ELEMENT INJECTOR MANIFOLDING AND FLOW SCHEMATIC (REF 2)

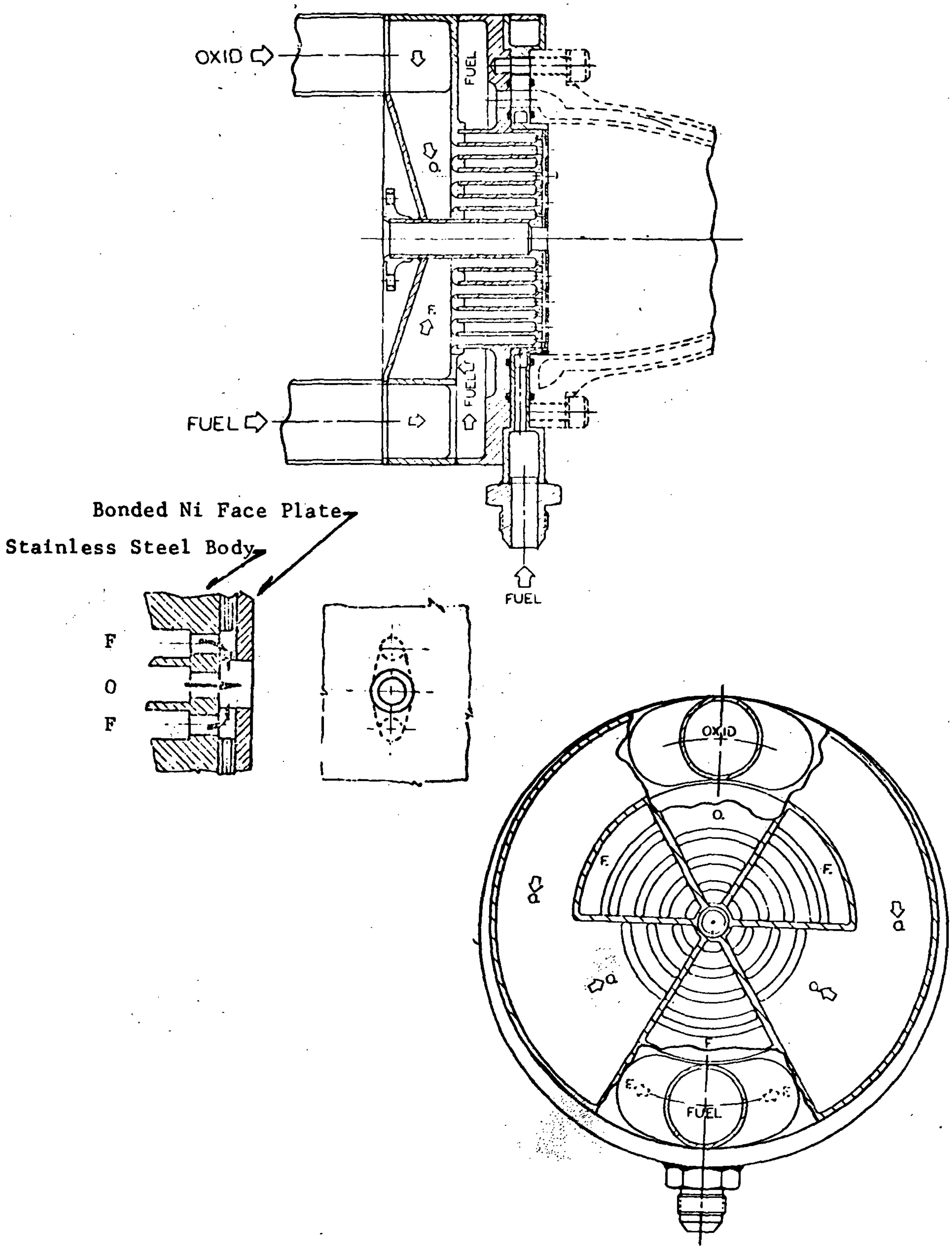
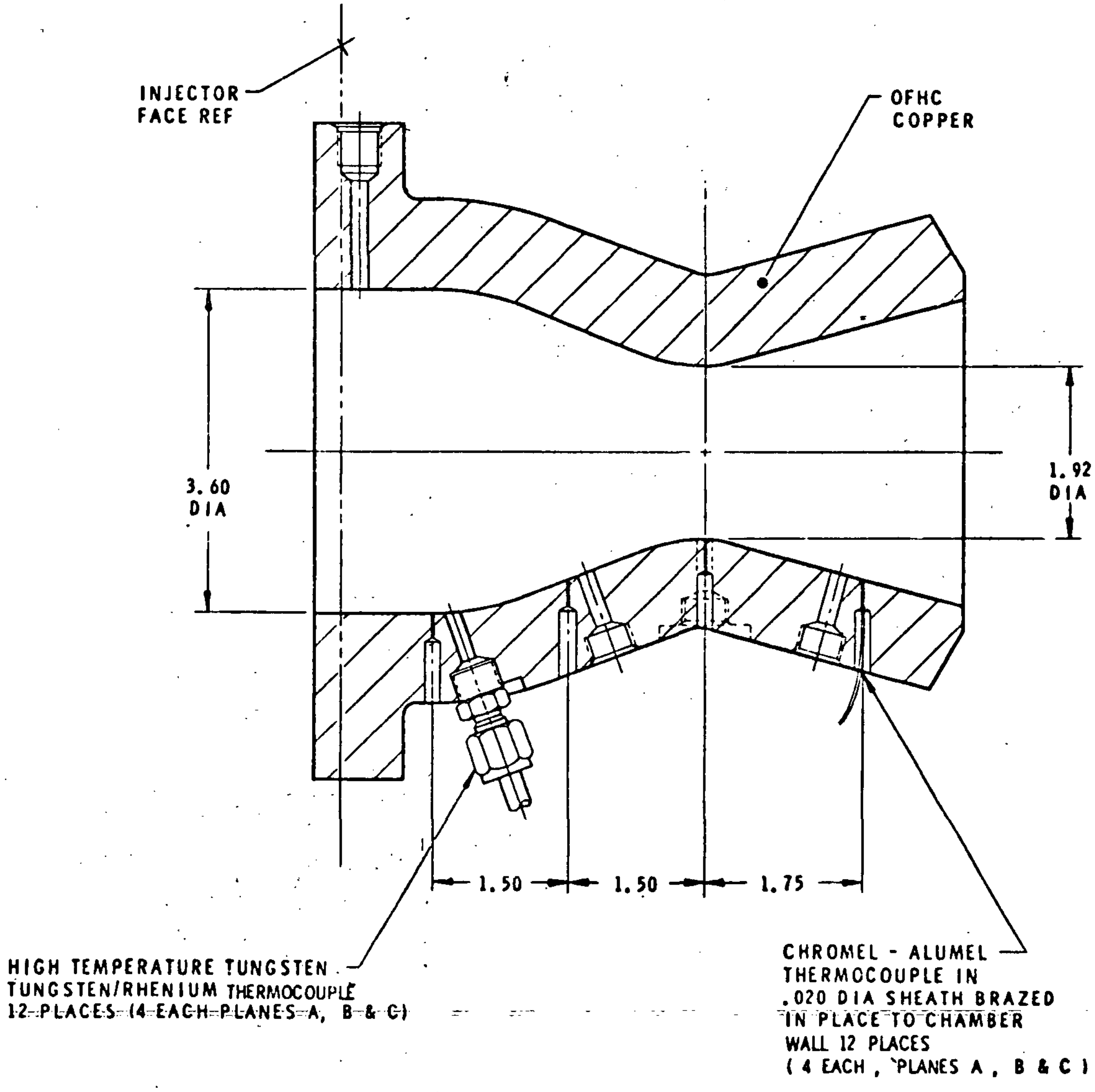


FIG. 22

HEAT SINK CHAMBER
L* = 15 INCHES
(REF 2)


INJECTOR FACE REF
OFHC COPPER
3.60 DIA
1.92 DIA
1.50
1.50
1.75
HIGH TEMPERATURE TUNGSTEN - TUNGSTEN/RHENIUM THERMOCOUPLE 12 PLACES (4 EACH PLANES A, B & C)
CHROMEL - ALUMEL THERMOCOUPLE IN .020 DIA SHEATH BRAZED IN PLACE TO CHAMBER WALL 12 PLACES (4 EACH, PLANES A, B & C)

GO_2/GH_2 ENGINE START SEQUENCE AND PRESSURE HISTORY

(REF 2)

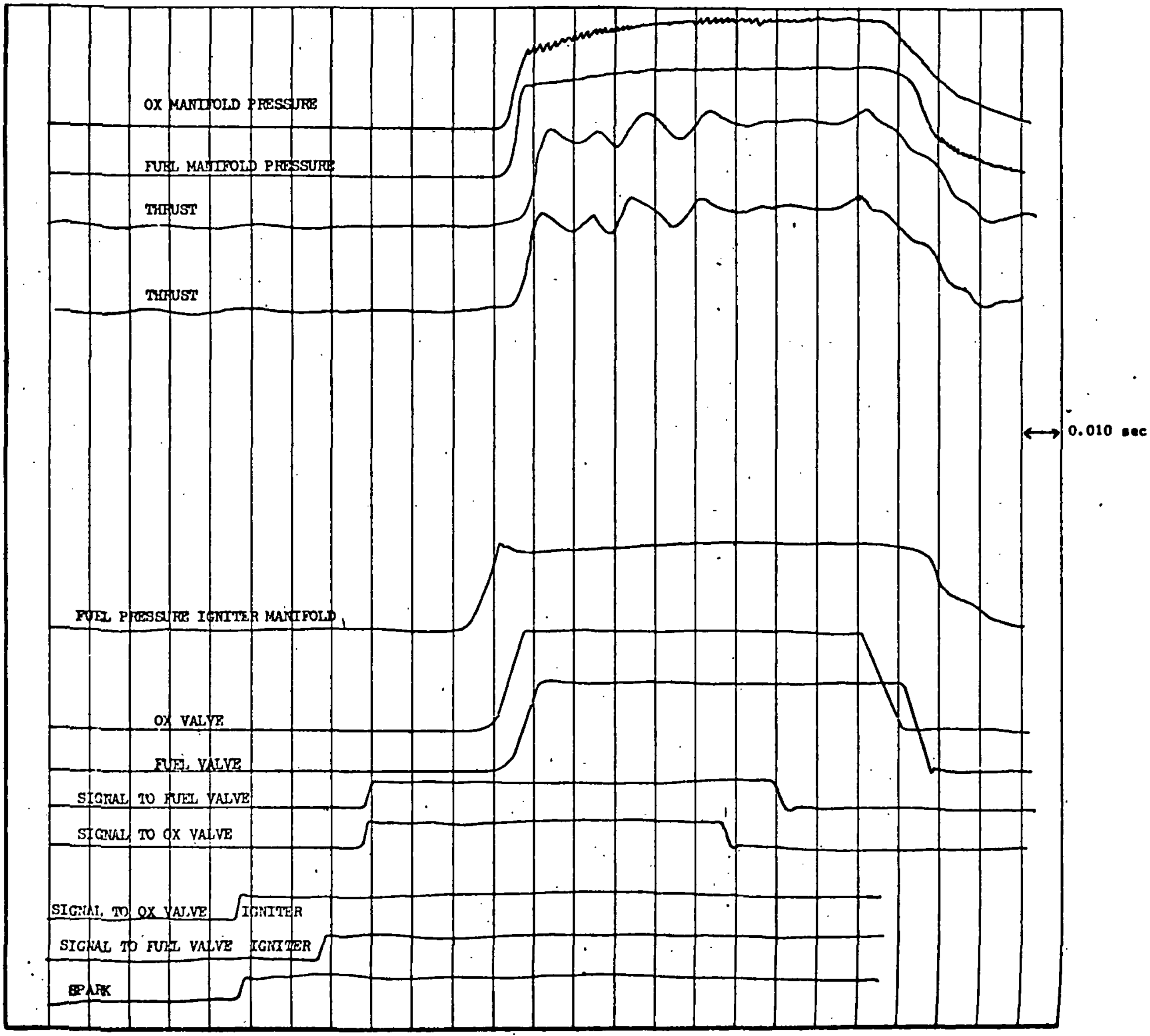


FIG. 24

DYNAMIC RESPONSE - AEROJET GO_2/GH_2 ENGINE
(COMPUTER PROGRAM RESULTS)

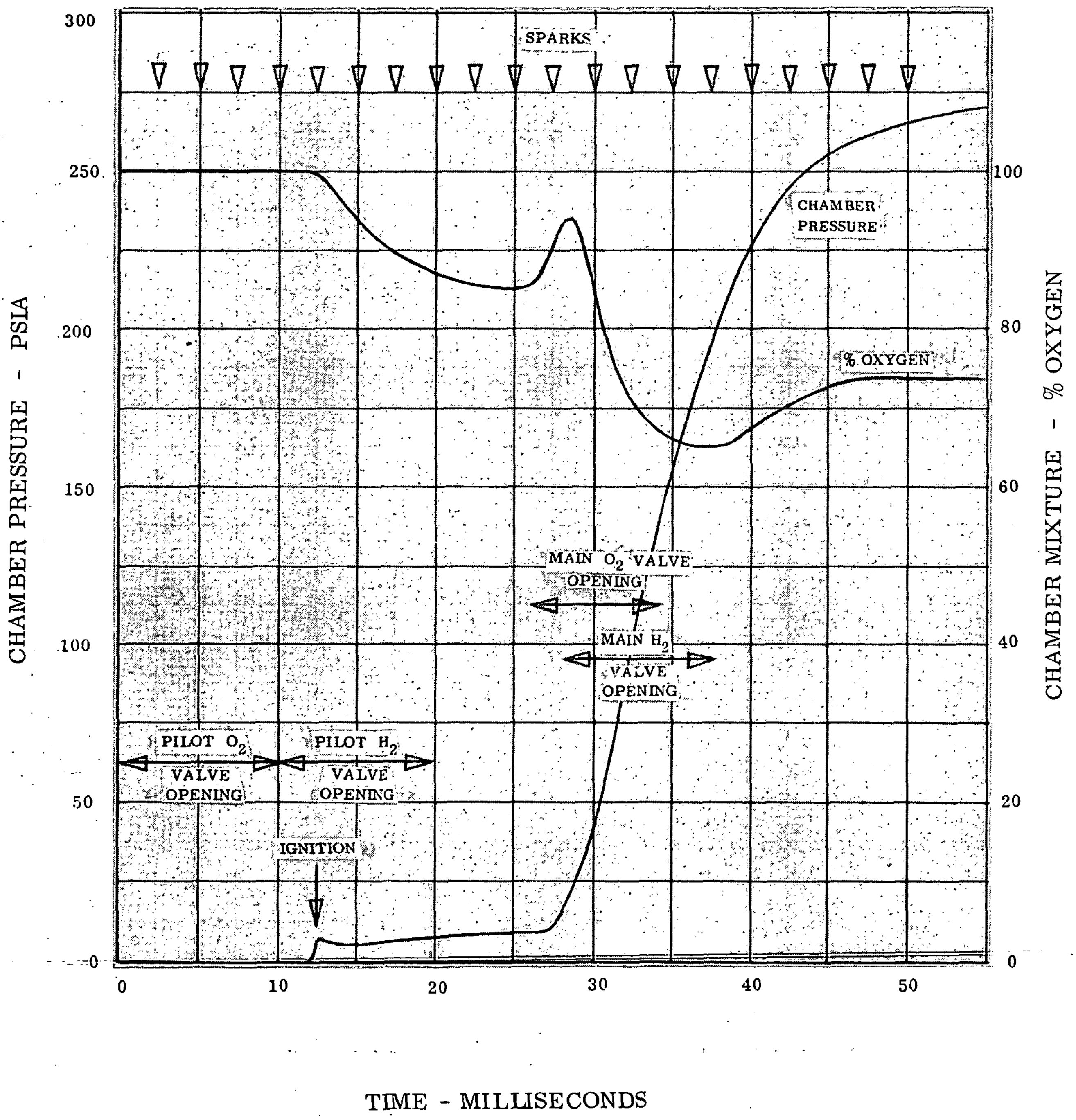


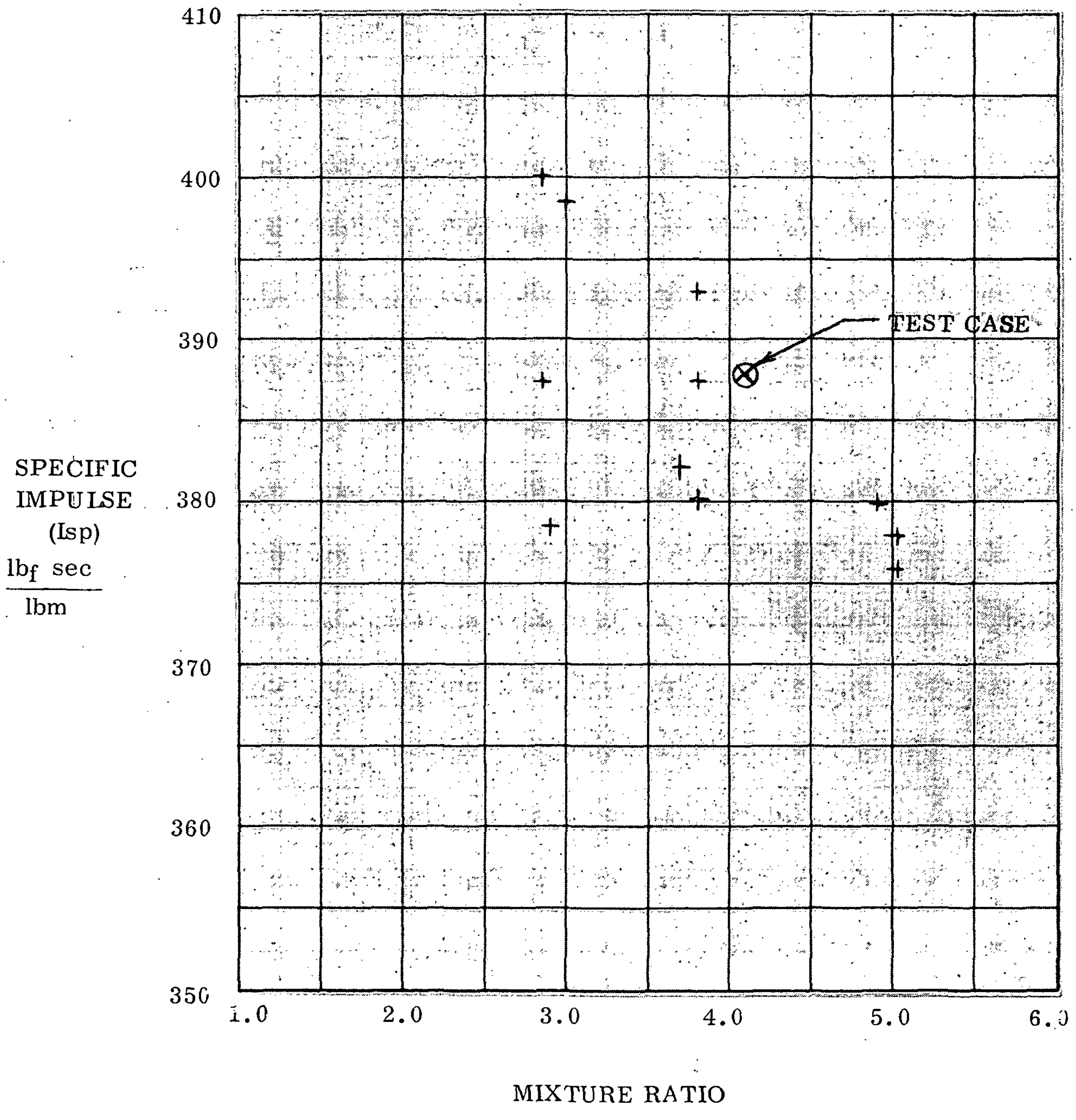
FIG. 25

SPECIFIC IMPULSE VS MIXTURE RATIO

+ Aerojet Data

L* = 15 inches

Data Modified for Area Ratio = 3.7



AEROJET REGENERATIVELY COOLED THRUST CHAMBER (REF 2) (Used for Heat Transfer Analysis)

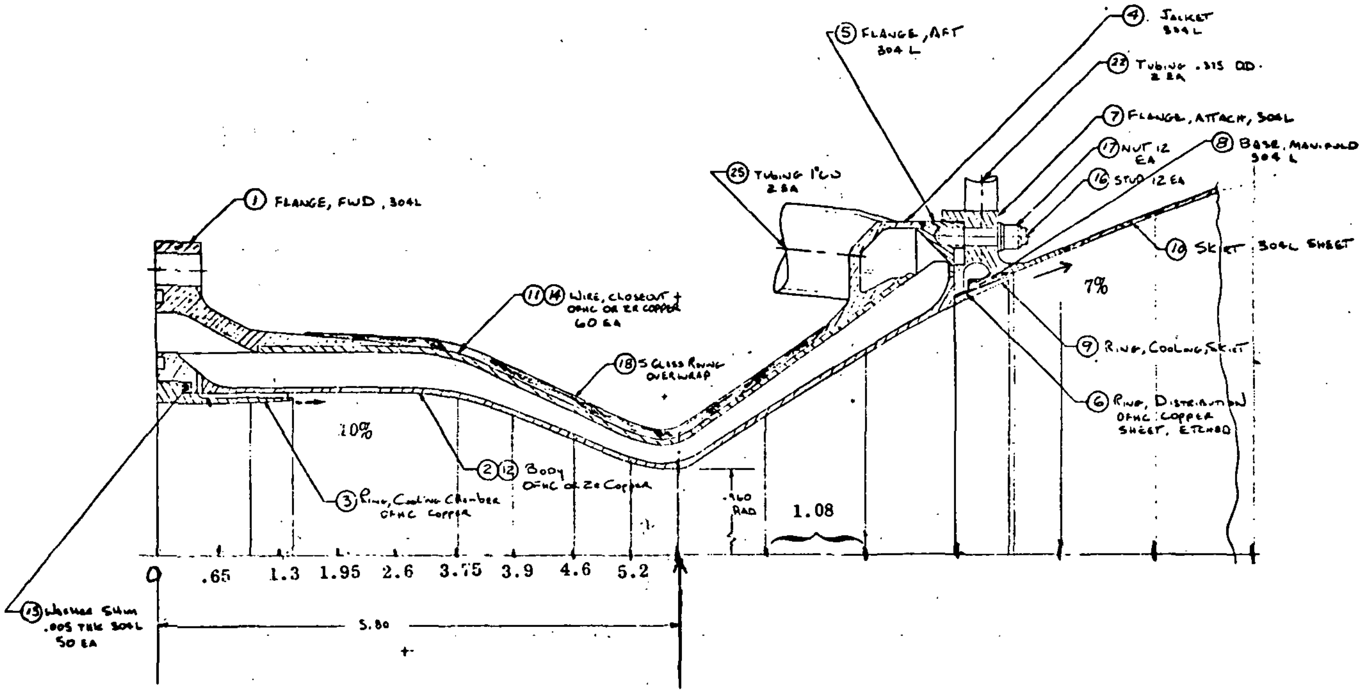


FIG. 27

THERMAL CHARACTERISTICS OF REGENERATIVELY COOLED THRUST CHAMBER DESIGN (REF 2)

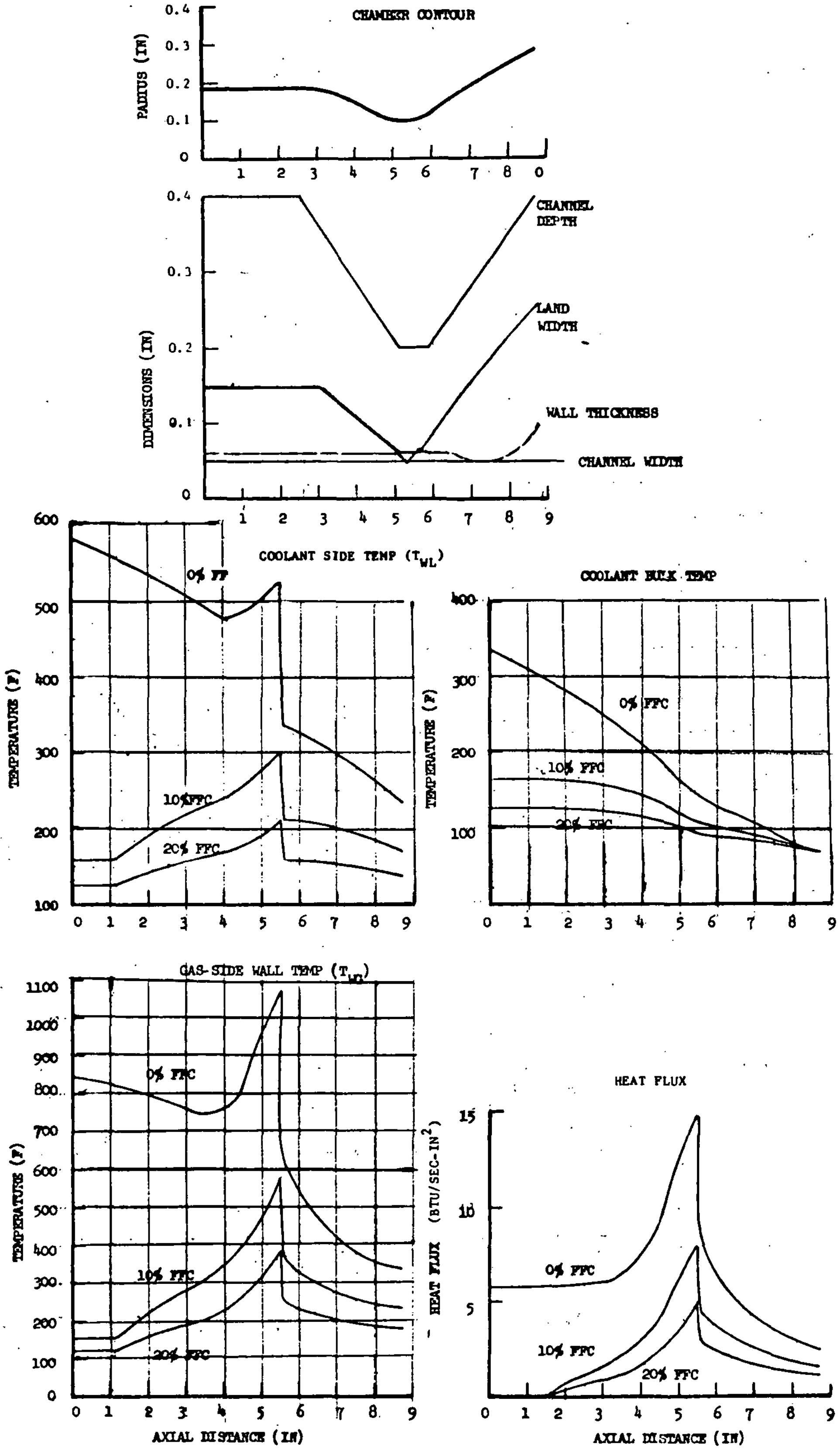
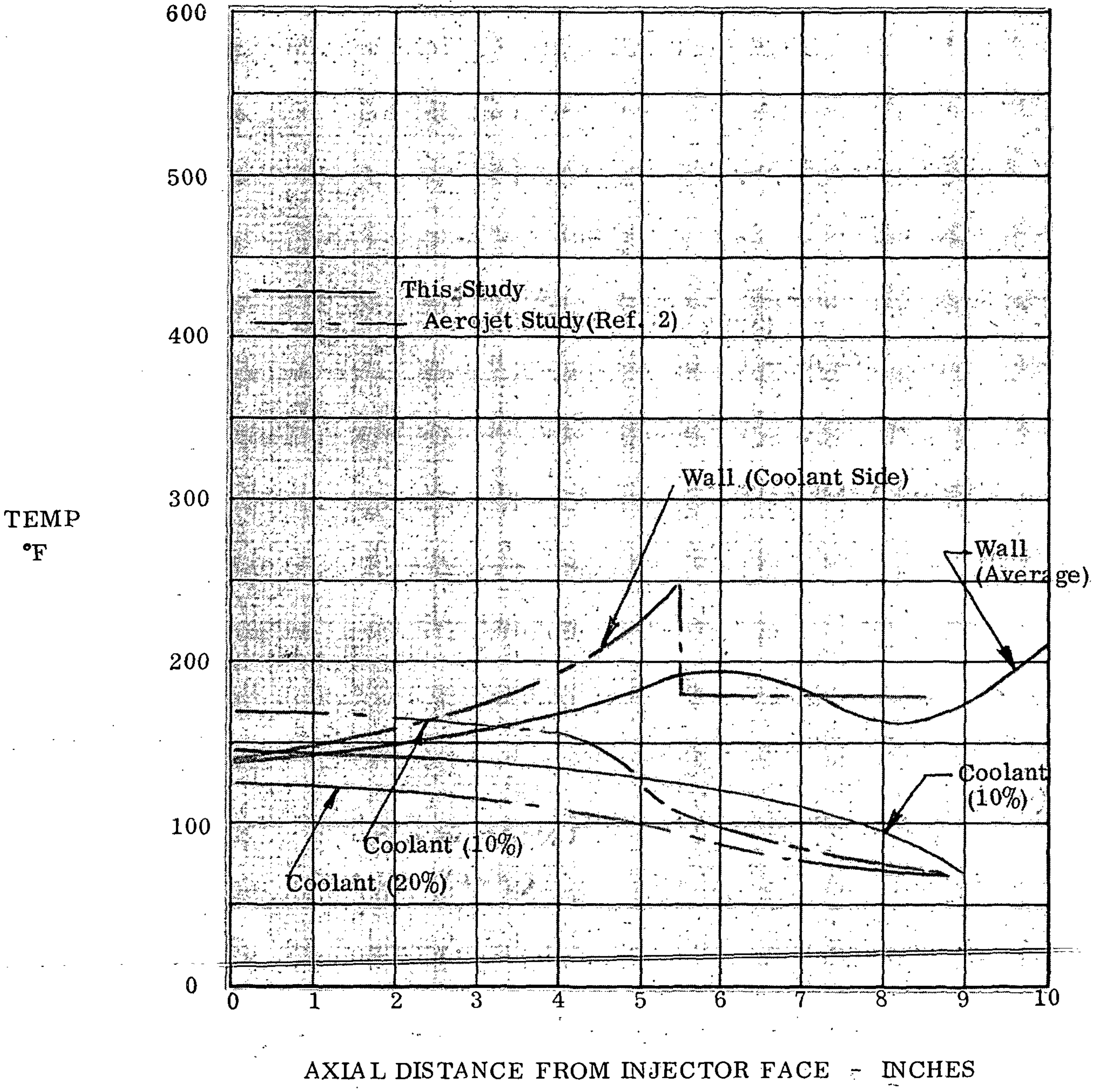


FIG. 28

TEMPERATURE VS LOCATION

AEROJET REGENERATIVELY COOLED THRUST CHAMBER

COMPARISON OF ANALYSIS
(REF 2)



CONCLUSIONS AND RECOMMENDATIONS

A FORTRAN V computer program was written which predicts the performance of GO_2/GH_2 rocket engines during both pulsing and continuous operation. The results of the comparison between the test case used for the computer simulation and the actual test data indicate the validity of the methods used. The adequacy of the comparison is dependent upon the input used. Critical parameters such as injector mixing efficiency and turbulent viscosity (used in the combustion program) must be chosen with care.

It is recommended (1) The program be tested with at least 3-4 engine concepts for which data is now available, and (2) The injection model be modified to include the data and models now being generated under NASA Contract NAS 3-14379 (Investigation of GH_2/GO_2 Combustion).

REFERENCES

1. ICRPG Liquid Propellant Thrust Chamber Performance Evaluation Manual, Chemical Propulsion Information Agency Publication No. 178, September 1968.

2. Schoenman, L.: "Hydrogen-Oxygen High P_c APS Engines", Report 14354-Q-2, Aerojet Liquid Rocket Company.

3. Shapiro, A. H.: Compressible Fluid Flow , 1953.

4. Schoenman, L. :"Hydrogen-Oxygen High P_c APS Engines", Report 14354-Q-1, Aerojet Liquid Rocket Company.

5. Hersch, M.: A Mixing Model for Rocket Engine Combustion. NASA TND-2881, 1965.

6. Hatch, J. E.; and Papell, S. : Use of Theoretical Flow Model to Correlate Data for Film Cooling or Heating an Adiabatic Wall by Tangential Injection of Gases of Different Fluid Properties. NASA TND-130, 1959.

7. Bartz, D. R. : A Simple Equation for Rapid Estimation of Rocket Nozzle Convective Heat Transfer Coefficients. Jet Propulsion, vol. 27, no. 1, January 1957, pp. 49-51.

8. Sellers, J. P. Jr. : Gaseous Film Cooling with Multiple Injection Stations. AIAA J., vol. 1, no. 9, September 1963, pp 2154-2156.

9. Bartle, E. R.; and Leadon, B. M.: The Effectiveness as a Universal Measure of Mass Transfer Cooling for a Turbulent Boundary Layer. Proc. 1962, Heat Transf. and Fluid Mech. Inst., June 1962, pp. 27.

10. Eckert, E. R. G. : Engineering Relations for Friction and Heat Transfer to Surfaces in High Velocity Flow. J. Aero. Sci., vol. 22, 1955, pp. 585-586.

11. Knuth, E. L.; and Dershin, H. : Use of Reference States in Predicting Transport Rates in High Speed Turbulent Flows with Mass Transfer. Int. J. Heat and Mass Transfer, vol. 6, 1963, pp. 999-1018.

12. Bender, L. S., et al. : "Kinetic Performance Handbook" NASA CR 72601.

13. McCarthy, J.; and Wolf, H.: "The Heat Transfer Characteristics of Gaseous Hydrogen and Helium", Rocketdyne R. R. 60-12, 1960.

14. Rupe, J. H.: "The Liquid-Phase Mixing of a Pair of Impinging Streams", PR 20-195, JPL, August 1953.

SYMBOLS

a	speed of sound (total conditions)
A	flow area (geometric)
$A*$	thrust chamber throat area
A_D	mass flow admittance or heat transfer admittance
B	function of (4fL/D)
c_f	skin friction coefficient
C_F	thrust coefficient
c_p	specific heat
C_i	wc_p of Ith node
$C*$	$pA*/\dot{m}$
C_d	orifice coefficient
$D*$	thrust chamber throat diameter
D	diameter
f	Fanning friction coefficient
F	thrust
G^I	mass transfer parameter for Ith slot
G	mass transfer parameter for transpiration
h	static enthalpy of mixture
h_i	static enthalpy of Ith species
h^I	heat transfer coefficient for Ith slot
$h*$	reference state enthalpy
H	total enthalpy
h	empirical constant

SYMBOLS (Continued)

I	impulse
Isp	specific impulse
k	thermal conductivity
$k_{b,p}$ $k_{c,p}$ $k_{f,p}$	backward, equilibrium,and forward rate constants for the pth reaction
L	length
Le	Lewis number
M	Mach number
$\dot{m}_c$	coolant mass flow rate
$\dot{m}$	thrust chamber mass flow rate
MW	molecular weight
m	percent oxygen
N_E	number of injection elements
N_A	number of annuli
p	pressure
P_o	stagnation pressure
Pr	Prandtl number
q_L	heat transfer rate from thrust chamber to wall
r	radial coordinate
r_w	local thrust chamber radius

SYMBOLS (Continued)

R_c^I	$(\rho us/\mu)_c^I$ Reynolds number for Ith slot based on coolant conditions and slot height
R	gas constant
$\bar{R}$	universal gas constant
R^*	thrust chamber throat radius
Re	Reynolds number
Res	resistance to heat flow
s^I	height of Ith slot
St	Stanton number
t	time
T	temperature
T_o	total temperature
u	axial velocity
V	volume
U^I	$\left(u_b / u_c^I\right)$
$\dot{w}$	mass flow rate
$\dot{w}_i$	volumetric rate of production of species i
$\dot{w}_T$	total mass flow
$\dot{w}_i$	molecular weight of species i
W	weight
x	axial coordinate

SYMBOLS (Continued)

x_S^I	axial coordinate of Ith slot
α_i	mass fraction of ith species
$\tilde{\alpha}_j$	mass fraction of jth element
α_i^*	reference state composition
ΔF_{BL}	boundary layer loss
η^I	effectiveness parameter for Ith slot
η_D	divergence effectiveness
μ	turbulent viscosity
μ^*	molecular viscosity based on reference state properties
$\mathfrak{F}$	configuration factor
μ_b	molecular viscosity based on bulk properties
μ_w	molecular viscosity based on wall conditions
ρ	density
ρ_o	stagnation density
ρ^*	density based on reference state properties
σ^I	thermodynamic parameter
Ψ	stream function
γ	specific heat ratio

SYMBOLS (Continued)

Subscripts

BL	boundary layer
e	edge conditions
w	wall conditions
b	bulk conditions
H	hydraulic
aw	adiabatic wall
c	coolant
T	trace
o	zero blowing condition
D	duct
$K_{I,J}$	volume identification (typ)
Δt	time increment
K	kinetic
S	shifting equilibrium,
F	frozen equilibrium

Superscripts

'	fuel rich region
"	oxidizer rich region
—	average
*	refers to throat

Page Intentionally Left Blank

PRECEDING PAGE BLANK NOT FILMED

APPENDIX A

PROGRAM INPUT

Page Intentionally Left Blank

PRECEDING PAGE BLANK NOT FILMED

INPUT FORMS

Preceding page blank

IBM 360 INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7/66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE
	CHARACTERIZATION INPUT				OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

NDYNA	NCOMBU	NINJE	NHEAT	NCAS	NBLFG	RUN CONTROL CARD (REQUIRED)

NNODE	NCONN	NITER	LPRTF	NCHEK	PCHEK	PSST	CD

NCOMCL	DPCOMC

NOINJ (1)	NOINJ (2)	NOINJ (3)

OMIT THESE TWO CARDS IF NO COMBUSTION CALCULATIONS (NCOMBU = 0)

NVOL	ICOMB	VOL	PRES	TEMP	RMIX

"NNODE" OF THESE REQ'D

OMIT IF NDYNA = 0

IBM 3(INPUT FORM - 80 COLUMN

FORM TMC 1949 REV. 7-66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

IADMIT (J,1)	IADMIT (J,2)	IRTYPE	CAREA	CCOEFF or DLEN	TIMØN	TIMØFF	TIMØPN	TIMCLS

"NCONN" OF THESE

SPTIME	SPKTL	SPKF

SPGAP	SPARKP	SPARKE	DC

COMBUSTION PROGRAM TITLE

MPSI	IPRESS	ITURB	LW	NTYPE	ISOBAT	MB	ICHEM	MC	MG	LZ	MA	MY	MH	ISBATY	NSLOT

PRNT	XMAX	X

OMIT IF NCOMBU = 0

IBM 3F INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE OF

Cols 1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80
XLE	SIGMA	DELPSI	XMPS	XK2	PSI(1)	Omit DELPSI value if NINJE = 1	
XK(1)	XK(3)	XK(4)	XK(5)	XK(6)	XK(7)		
TAR	XP(1)	XP(2)	XP(3)	XP(4)	XP(5)	XP(6)	Omit TAR value if NDYNA = 1
CGP(1,J)	CGP(2,J)	CGP(3,J)	CGP(4,J)	CGP(5,J)	CGP(6,J)	CGP(7,J)	← 4 CARDS (J = 1, ..., 4)
TWX(1,J)	TWX(2,J)	TWX(3,J)	TWX(4,J)	TWX(5,J)	TWX(6,J)	TWX(7,J)	4 CARDS (J = 1, 4) OMIT IF MB = 0, ISOBAT ≠ 2 and ISBATY ≠ 1
QLX(1,J)	QLX(2,J)	QLX(3,J)	QLX(4,J)	QLX(5,J)	QLX(6,J)	QLX(7,J)	4 CARDS (J = 1, 4) OMIT IF MB = 0, ISOBAT ≠ 2 and ISBATY ≠ 2
RUCX(1,J)	RUCX(2,J)	RUCX(3,J)	RUCX(4,J)	RUCX(5,J)	RUCX(6,J)	RUCX(7,J)	4 CARDS
TCX(1,J)	TCX(2,J)	TCX(3,J)	TCX(4,J)	TCX(5,J)	TCX(6,J)	TCX(7,J)	4 CARDS

OMIT IF MB = 0, ISOBAT ≠ 3 and ISBATY ≠ 1 or 2

OMIT IF NCOMBU = 0

IBM 3ʗ INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO. | JOB NAME | NAME | EXTENSION | DEPT. | PAGE OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

OMIT IF NCOMBU = 0

TWX(1,J)	TWX(2,J)	TWX(3,J)	TWX(4,J)	TWX(5,J)	TWX(6,J)	TWX(7,J)	4 CARDS	OMIT IF MB=0, ISOBAT ≠ 3 and ISBATY ≠ 3 or 4
TCX(1,J)	TCX(2,J)	TCX(3,J)	TCS(4,J)	TCX(5,J)	TCX(6,J)	TCX(7,J)	4 CARDS	
NSLOT	OMIT IF ISOBAT ≠ 4							
XS(K)	SH(K)	UC(K)	RUCF(K)	TCS(K)	OMIT IF ISOBAT ≠ 4 NSLOT CARDS			
RSTAR	DST	PRC	OMIT IF ISOBAT ≠ 4					
NELEM DCHAMB	TØ	TH	EMR	OMIT IF NINJE = 0				
WT	PØ	FM	OMIT IF NINJE = 0 OR NDYNA = 1					
T(1)	T(2)	T(3)	T(4)...				Omit this card if NINJE - 1 MPSI Values	

IBM 3(INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE OF
1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54	55 56 57 58 59 60 61 62	63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

U(1)	U(2)	U(3)...					Omit this card if NINJE = 1 MPSI VALUES
FIX(1)	FIX(2)	FIX(3)...					Omit this card if ICHEM ≠ 3 MPSI VALUES
TELAP(1)	TLAP(2)	TLAP(3)...					Omit this card if X = 0
YSPEC(1, I)	YSPEC(2, I)	YSPEC(3, I)	YSPEC(4, I)	YSPEC(5, I)	YSPEC(6, I)	YSPEC(7, I)	Omit this card if NINJE = 1 MPSI SETS
YSPEC(8, I)	YSPEC(9, I)						
RT	RE	XN	PERBEL	ENTHØ	ENTHH		
ØFINPT	PER	TTT					
BLANK IF NDYNA = 1	BLANK IF NO SUPERSONIC FILM COOLING						

OMIT IF NCOMBU = 0

IBM 3(INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

TITLE CARD - 72 CHARACTERS OF DESCRIPTIVE INFORMATION

NNODE	NTYP FL	NFLM FL	NRGN FL	NLFL	INJFL
TSTOP or NCOUNT	PRINT INTERVAL				
TWALLF	TSINKF	CAPPA	EPS	PO	OF
RC	RHO	CP	TINJH2	TINJØ2	
X	DI	DO	NNODE Cards		

OMIT IF NHEAT = 0

S-1220

IBM 3ſ INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE OF
1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54	55 56 57 58 59 60 61 62	63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

NINJ

SLOT	HSLOT	WCOOL

NINJ cards

Omit cards on this page if NRGNFL = 0

OMIT IF NHEAT = 0

IBM 3ᶜ INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE
					OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

HRWD	NPASS	XREGEN		
X	HPASS	WPASS	FIN EFFICIENCY	

Omit cards on this page if
NRGNFL = 0

NNODE Cards

OMIT IF NHEAT = 0

IBM 3ᶠ INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO. | JOB NAME | NAME | EXTENSION | DEPT. | PAGE OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

MASS FLOW	EMISSIVITY	LINER LENGTH
X	LINER ID	LINER OD

NNODE Cards

Omit cards on this page if NLFL = 0

OMIT IF NHEAT = 0

IBM 3ſ INPUT FORM - 80 COLUMN

FORM TMC 1549 REV. 7-66

JOB NO.	JOB NAME	NAME	EXTENSION	DEPT.	PAGE OF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

AINJS	EMINJ	ARINJH	HGINJH	ARINJØ	HGINJØ	
RESINJ	QFINJ	WTINJ	CPINJ			

Omit cards on this page if INJFL = 0

OMIT IF NHEAT = 0

PRECEDING PAGE BLANK NOT FILMED

INPUT VARIABLES

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS

NAME	MODEL	FORMAT	DESCRIPTION
NDYNA	Driver	I 3	1 if dynamics calculation, 0 otherwise
NCOMBU	"	"	1 if combustion calculation, 0 otherwise
NINJE	"	"	1 if injection model used, 0 otherwise
NHEAT	"	"	1 if heat transfer calculation, 0 otherwise
NNODE	Dynamics	I 4	No. of nodes (volumes)
NCONN	"	"	No. of connectors (orifices or ducts)
NITER	"	"	Maximum no. of iterations
LPRTF	"	"	Print frequency (print results every LPRTF iterations)
NCHEK	"	"	Time is advanced to the first valve closing time when:
PCHEK	"	E10.2	$\frac{d(PRES(NCHEK))}{d(TIME)} < PCHEK$ (psia/ms)
PSST	"	"	once PRES(NCHEK) $\geq$ PSST (psia)
NCOMCL	"	I 4	Call combustion every NCOMCL dynamics iterations
DPCOMC	"	F10.2	Call combustion every time combustion pressure increases by DPCOMC (psia) *
CD	"	"	Nozzle coefficient * can use either or both
NOINJ(I)	"	24 I 3	Volume numbers feeding injector face (used by injection model)
NVOL(I)	"	I 3	Volume number (consecutive integers, I = 1,....., NNODE)
ICOMB(I)	"	"	ICOMB(I) = 0 Neither pilot nor combustor 1 Unlit combustor 2 Lit combustor 3 Unlit pilot 4 Lit pilot
VOL(I)	"	F10.0	Node volume (in^3)
PRES(I)	"	F10.0	Initial pressure in Node I (psia)

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS
(Continued)

NAME	MODEL	FORMAT	DESCRIPTION
TEMP(I)	Dynamics	F10.0	Initial temperature in Node I (°R)
RMIX(I)	"	"	Initial Massfraction O_2 in Node I
IADMIT(J,I)	"	I 3	Volume no. connected to admittance J (I = 1,2)
IRTYPE(J)	"	"	IRTYPE(J) = 0 orifice = 1 duct = 2 throat
CAREA(J)	"	F10.0	Full open cross sectional area (in^2)
CCOEFF(J)	"	"	Discharge coefficient for I = 0, 2
DLEN(J)	"	"	Length (inches) for I = 1
TIMON(J)	"	"	Time valve starts to open(sec) If all timing parameters are 0.0, area is assumed constant at CAREA
TIMOFF(J)	"	"	Time valve starts to close(sec)
TIMOPN(J)	"	"	Valve opening response time
TIMCLS(J)	"	"	Valve closing response time
SPTIME	"	"	Time of initial spark (sec)
SPKTL	"	"	Time of last spark (sec)
SPKF	"	"	Time between sparks (sec)
SPGAP	"	"	Spark plug gap (inches)
SPARKP	"	"	Spark plug potential (volts)
SPARKE	"	"	Spark energy (millijoules)
DC	"	"	Chamber diameter (inches)
-	Combustion	12A6	Title card - any statement. Will be printed on each radial profile output.
MPSI	"	I 5	Number of grid points at x = x initial .
IPRESS	"	"	Number of grid points at x = 0, used in halving the grid, and program restarts at x > 0.
ITURB	"	"	1 - Use Hersch viscosity model.
			2 - Viscosity is Input
LW	"	"	Specifies the laminar viscosity model used for the wall cooling modes. LW = 1 employs the Sutherland air viscosity model.

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS
(Continued)

NAME	MODEL	FORMAT	DESCRIPTION
NTYPE	Combustion	I 5	0 - Axisymmetric flow field 1 - Plane two-dimensional flow
ISOBAT	"	"	1 - Isoenergetic wall 2 - Regeneratively cooled wall 3 - Transpiration cooling 4 - Slot cooling
MB	"	"	0 - Free jet; P (x) prescribed 1 - Ducted flow; P (x) prescribed 2 - Ducted flow; Wall radius (x) prescribed
ICHEM	"	"	1 - Chemically frozen flow 2 - Equilibrium ("complete combustion") chemistry 3 - Finite rate chemical kinetics
MC	"	"	Printout of the flow field radial profiles is made every MC finite difference steps. The default is 10.
MG	"	"	Specifies the diluent specie used as a tracer in the transpiration model. The diluent may also be present initially in the main stream flows, when other wall cooling models are used. 1 - Diluent is Nitrogen 2 - Diluent is Helium 3 - Diluent is Argon
LZ MA MY MH	"	"	Printout dump controls for various portions of the program. In general, nn = 0 means no dump; nn = 1 yields moderate dump; nn ≥ 2 yields overwhelming printout dump.
ISBATY	"	"	Specifies alternatives in the regeneratively cooled and transpiration cooled wall models: Regenerative Cooling (ISOBAT = 2) ISBATY 1. Wall temperature, $T_W(x)$, (°R), specified. The wall heat transfer, q_W (x) $^{Btu}/_{in^2}$ sec), is computed. 2. q_W (x) specified. T_W is computed.

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS
(Continued)

NAME	MODEL	FORMAT	DESCRIPTION
ISBATY (Cont.)	Combustion	I 5	Transpiration Cooling (ISOBAT = 3) ISBATY 1. Coolant temperature, $T_c(x)$, (°R), and coolant unit area mass flow rate, $(\rho u\ (x))_c$ (lb/in^{2}sec), are specified, and coolant domain "edge" conditions are used in the model. T_w is computed. 2. T_c (x) and $(\rho u\ (x))_c$ are given, and bulk flow "edge" conditions are used. T_w is computed. 3. $T_c(x)$ and T_w (x) are given, with coolant domain "edge" conditions. $(\rho u)_c$ is computed. 4. T_c (x) and T_w (x) are specfied with bulk flow "edge" conditions. $(\rho u)_c$ is computed.
NSLOT	"	"	The number of slots in the slot wall cooling model.
PRNT	"	E10.8	Printout interval Δx (inches).
XMAX	"	"	Axial distance to which calculation is carried out, x_{max} (inches).
X	"	"	Axial station at which calculation is begun, x initial (inches) .
XLE	"	"	Turbulent Lewis number.
SIGMA	"	"	Turbulent Prandtl number.
DELPSI	"	"	The spacing between adjacent flow field grid points. The dimensions of $\Delta\ \Psi$ are $\sqrt{\text{lb/sec}}$ in axisymmetric flows and (lb/sec) in plane two-dimensional flows.
XMPS	"	"	Available for use in reducing the diffusion step size, $\Delta x = \Delta x$/XMPS. Let MPS = 1.
XK2	"	"	Constant employed in the turbulent viscosity models.
PSI(1)	"	"	Value of the flow field grid point, ψ 1, nearest the chamber centerline, with dimensions of $\left[\text{lb/sec}\right]^{\left(\frac{1+\text{NTYPE}}{2}\right)}$.

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS
(Continued)

NAME	MODEL	FORMAT	DESCRIPTION
XK(1)	Combustion	E10.8	α in the Hersch viscosity model. Let $\alpha = 1$.
XK(3)	"	"	The spacing of initial fuel and oxidizer "rings" in the Hersch model, s (inches).
XK(4)	"	"	Parameter in the Hatch-Papell film cooling model. Input as .04.
XK(5)	"	"	Mean maximum number of finite rate chemistry steps per flow field diffusion step. Default value is 10.
XK(6)	"	"	$C_f/2$ in the momentum eq. wall boundary condition. Recommend using $1x10^{-3}$.
XK(7)	"	"	D, for x/D printout, (inches).
TAR	"	"	Initial wall radius (inches) for MB = 0 and 1; initial static pressure (lb/in^2) for MB = 2.
XP(1) thru XP(4)	"	"	End points of domains of polynomials for wall radius, or static pressure, and other wall boundary conditions (inches), which are imput below.
XP(5)	"	"	Maximum finite rate chemistry time step(seconds). Default is 1×10^{-5} seconds.
XP(6)	"	"	Lower tolerance for changes in finite rate chemistry time step. Default is 5×10^{-3}.
CGP(I, J)	"	7E10.8	Coefficients of polynomial J for static pressure (MB=0, 1) or chamber wall radius (MB=2). J =1, 4 (four cards). $F_J(X) = \sum_{I=1}^{6} a_j \left(X - X_I^*\right)^{j-1}$ for $X < XP(I)$ where $CGP(I,J) = X_I^*$
TWX(I, J)	"	"	Four sets of polynomial coefficients for $T_W(X)$ (°R) For regenerative cooling if ISOBAT = 2. For transpiration cooling if ISOBAT = 3.
QLX(I, J)	"	"	Four sets of polynomial coefficients for $q_W(X)$ ($Btu/in^2 sec$) (Regenerative cooling).

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS
(Continued)

NAME	MODEL	FORMAT	DESCRIPTION
RUCX(I, J)	Combustion	7E10.8	Four sets of polynomial coefficients for $(\rho u)_c$ (lb/in^2 sec) (Transpiration cooling).
TCX(I, J)	"	"	Four sets of polynomial coefficients for T_c (X) (°R) (Transpiration cooling).
NSLOT	"	E10.8	Number of slots in the chamber wall (≤ 21).
XS(K)	"	"	Axial location of first slot (inches). K = 1, 2, . . . , NSLOT
SH(K)	"	"	Height of first slot (inches)
UC(K)	"	"	Coolant velocity (in/sec)
RUCF(K)	"	"	Coolant mass flux (lb/in^2 sec)
TCS(K)	"	"	Coolant temperature (°R)
RSTAR	"	"	Radius of curvature of engine throat (inches)
PST	"	"	Throat diameter (inches)
PRC	"	"	Coolant Prandtl number.
NELEM	Injection	I 3	No. of injector elements
DCHAMB	"	F10.0	Chamber diameter at injector plane (inches)
TO	"	"	Oxygen total temperature (°R)
TH	"	"	Hydrogen total temperature (°R)
EMR	"	"	Rupe mixing factor
WT	"	"	Mass flow through injector (lb/sec)
PO	"	"	Chamber total pressure (psia)
FM	"	"	Mass fraction oxygen thru injector
T(I)	Combustion	E10.8	Static temperature (°R) at grid point I
U(I)	"	"	Axial velocity (in/sec) at each grid point
FIX(I)	"	"	Chemistry time step (seconds) at each grid point
TELAP(I)	"	"	Integrated streamline residence times (seconds) for each grid point
YSPEC(1, I)	"	"	H mass fraction at grid point I
YSPEC(2, I)	"	"	O mass fraction at grid point I
YSPEC(3, I)	"	"	H_2O mass fraction at grid point I

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS
(Continued)

NAME	MODEL	FORMAT	DESCRIPTION
YSPEC(4,I)	Combustion	E10.8	H_2 mass fraction at grid point I
YSPEC(5,I)	"	"	O_2 mass fraction at grid point I
YSPEC(6,I)	"	"	OH mass fraction at grid point I
YSPEC(7,I)	"	"	HO_2 mass fraction at grid point I
YSPEC(8,I)	"	"	H_2O_2 mass fraction at grid point I
YSPEC(9,I)	"	"	Diluent mass fraction at grid point I
RT	Performance	F10.0	Throat radius (inches)
RE	"	"	Exit radius (inches)
XN	"	"	Nozzle length (inches)
PERBEL	"	"	Percent bell
ENTHO	"	"	Oxygen enthalpy (Btu/lb)
ENTHH	"	"	Hydrogen enthalpy (Btu/lb)
OFINPT	"	"	Mixture ratio
PER	"	"	Percent fuel injected in supersonic region
TTT	"	"	Temperature of fuel injected in supersonic region
NNODE	Heat Transfer	I 3	Number of nodes
NTYPFL	"	"	1 if transient, 0 if steady state
NFLMFL	"	"	1 if film cooling, 0 if not
NRGNFL	"	"	1 if regen cooling, 0 if not
NLFL	"	"	1 if liner, 0 if not
INJFL	"	"	1 if injector, 0 if not
TSTOP	"	F12.0	Cut-off time for transient case (sec)
NCOUNT	"	"	Number of iterations for steady state case (input as a real number)
PRINT INTERVAL	"	"	Print interval in seconds for transient and in number if iterations for steady state
TWALLF	"	"	Initial wall temperature (°F)

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS (Continued)

NAME	MODEL	FORMAT	DESCRIPTION
TSINKF	Heat Transfer	F12.0	Sink temperature (°F)
CAPPA	"	"	Conductivity (Btu/in sec °R)
EPS	"	"	Wall emissivity (none)
PO	"	"	Combustion chamber pressure (lb/in^2)
OF	"	"	Mixture ratio (none)
RC	"	"	Radius of curvature at throat (inches)
RHO	"	"	Density of wall material (lbs/in^3)
CP	"	"	Specific heat capacity of wall material (Btu/lb °R)
TINJH2	"	"	Hydrogen injection temperature (°R)
TINJO2	"	"	Oxygen injection temperature (°R)
X	"	"	Axial distance of node (in)
DI	"	"	Wall inner diameter (in)
DO	"	"	Wall outer diameter (in)
NINJ	"	I 3	Number of film injection stations
SLOT	"	F12.0	Axial distance of film injection station (in)
HSLOT	"	"	Slot height (in)
WCOOL	"	"	Coolant weight flow (lb/sec)
HRWD	"	"	Hydrogen regen weight flow (lb/sec)
NPASS	"	I 12	Number of regen passages
XREGEN	"	F12.0	Regen injection station (in)
X	"	"	Axial distance of node (in) (must correspond to nodes)
HPASS	"	"	Regen passage height (in)
WPASS	"	"	Regen passage width (in)
FIN EFFICIENCY	"	"	Fin efficiency (none)
MASS FLOW	"	"	Mass flow between liner and wall (lb/sec)
EMISSIVITY	"	"	Liner emissivity (none)
LINER LENGTH	"	"	Maximum axial distance of liner (in)

O_2/H_2 CHARACTERIZATION PROGRAM INPUT PARAMETERS
(Continued)

NAME	MODEL	FORMAT	DESCRIPTION
X	Heat Transfer	F12.0	Axial distance of node (in) (must correspond to a node)
LINER ID	"	"	Liner inner diameter (in)
LINER OD	"	"	Liner outer diameter (in)
AINJS	"	"	Injector surface area (in^2)
EMINJ	"	"	Injector emissivity
ARINJH	"	"	Injector H_2 Convection Area (in^2)
HGINJH	"	"	Injector H_2 Convective Coefficient $\left(\frac{Btu}{in^2 sec\ °R}\right)$
ARINJO	"	"	Injector O_2 Convection Area (in^2)
HGINJO	"	"	Injector O_2 Convective Coefficient $\left(\frac{Btu}{in^2\ sec\ °R}\right)$
RESINJ	"	"	Injector - Combustor thermal resistance (sec°R/Btu)
OFINJ	"	"	Injector face heat flux (But/in^2 sec)
WTINJ	"	"	Injector weight (lbs)
CPINJ	"	"	Injector specific heat (But/lb °R)

APPENDIX B

SAMPLE CASES

Sample Case I

Sample Case I is a combustion and performance computation utilizing the injection model to provide starting profiles. The first data card required is the run control card with NCOMBU and NINJE set equal to 1 (all others zero or blank). All dynamics input is omitted. The second data card, then, is the combustion program title card followed by the combustion control card (MPSI thru NSLOT). The fourth, fifth, and sixth cards are (PRNT thru X), (XLE thru PSI(1)), and (XK(1) thru XK(7)). The next five cards define the wall contour. The seventh card (TAR thru XP(6)) defines the polynomial limits, and the next four are the polynomial coefficients (CGP(1,J) thru CGP(7,J) for J = 1,..,4). The next data cards are the two injection model cards, (NELEM thru EMR) and (WT thru FM). All intervening cards shown on the input sheet are omitted. The final cards are the two performance input cards (RT thru ENTHH) and (OFINPT thru TTT).

The following page is a copy of the Sample Case I input report printed by the program followed by a sample of the combustion output and the final performance report. Note that injection model results (species, velocity, and temperature profiles) are printed as part of the combustion input report.

```
                                     * * * COMBUSTION INPUT * * *

TEST CASE  AEROJET H2/02 1500 LB THRUST ENGINE
 50  50   1   1   0   1   2   2  50   1   0   0   0   0   0   0
    .50000000-00    .52000000+01    .00000000
    .10000000+01    .10000000+01    .10000000+01    .10000000+01    .18000000-01    .00000000
    .10000000+01    .00000000       .36000000-00    .40000000-01    .10000000+02    .99999999-03    .36500000+01
                                         INITIAL YW OR P.POLYNOMIAL LIMITS
    .30000000+03    .27500000+01    .40000001+01    .53500000+01    .26416000-00    .99999999-05    .49999999-02
                                       WALL RADIUS(INCHES) POLYNOMIALS
    .18250000+01    .00000000       .00000000       .00000000       .00000000       .00000000       .00000000
    .69653799-00    .60629699-00    .96551999-01   -.10684000+00    .21405500-01   -.17283500-02    .00000000
    .32461800+01   -.42677000-00    .00000000      -.00000000       .00000000       .00000000       .00000000
    .96343600-00   -.55418400-00    .89985400-00   -.62323440-00    .67604800-00   -.31049930-00    .53500000+01
                                            TEMPERATURE(R) ARRAY
    .52971962+03    .52971962+03    .52971962+03    .52971962+03    .52971962+03    .52971962+03    .52971960+03
    .52971960+03    .52971960+03    .52971960+03    .52971960+03    .52971960+03    .52971962+03    .52971962+03
    .52971962+03    .52971960+03    .52971960+03    .52971960+03    .52971960+03    .52971960+03    .52971962+03
    .52971962+03    .52971962+03    .52971960+03    .52971960+03    .52971960+03    .52971960+03    .52971960+03
    .52971962+03    .52971962+03    .52971962+03    .52971960+03    .52971960+03    .52971960+03    .52971960+03
    .52971960+03    .52971962+03    .52971962+03    .52971962+03    .52971960+03    .52971960+03    .52971960+03
    .52971960+03    .52971960+03    .52971962+03    .52971962+03    .52971962+03    .52971960+03    .52971960+03
    .52971960+03

                                       VELOCITY PROFILES (IN/S)
    .15844788+04    .15844788+04    .15844788+04    .15844788+04    .15844788+04    .15844788+04    .11029009+04
    .11029009+04    .11029009+04    .11029009+04    .11029009+04    .11029009+04    .15844788+04    .15844788+04
    .15844788+04    .11029009+04    .11029009+04    .11029009+04    .11029009+04    .11029009+04    .15844788+04
    .15844788+04    .15844788+04    .11029009+04    .11029009+04    .11029009+04    .11029009+04    .11029009+04
    .15844788+04    .15844768+04    .15844788+04    .11029009+04    .11029009+04    .11029009+04    .11029009+04
    .11029009+04    .15844788+04    .15844788+04    .15844788+04    .11029009+04    .11029009+04    .11029009+04
    .11029009+04    .11029009+04    .15844788+04    .15844788+04    .15844788+04    .11029009+04    .11029009+04
    .11029009+04
                                       SPECIE MASS FRACTION PROFILES
    .00000000       .00000000       .00000000       .31674202-00    .68325799-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .31674202-00    .68325799-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .31674202-00    .68325799-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .31674202-00    .68325799-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .31674202-00    .68325799-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .31674202-00    .68325799-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .11880413+00    .38119587-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .11880413+00    .88119587-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .11880413+00    .88119587-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .11880413+00    .88119587-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .11880413+00    .88119587-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .11880413+00    .88119587-00    .00000000       .00000000
    .00000000       .00000000
    .00000000       .00000000       .00000000       .31674202-00    .68325799-00    .00000000       .00000000
    .00000000       .00000000
```

```
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .31674202-00   .68325799-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
.00000000   .00000000   .00000000   .11880413+00   .88119587-00   .00000000   .00000000
.00000000   .00000000
```

```
.00000000      .00000000      .00000000      .11880413+00   .88119587-00   .00000000      .00000000
.00000000      .00000000
.00000000      .00000000      .00000000      .31674202-00   .68325799-00   .00000000      .00000000
.0000 000      .00000000
.00000000      .00000000      .00000000      .31674202-00   .68325799-00   .00000000      .00000000
.00000000      .00000000
.00000000      .00000000      .00000000      .31674202-00   .68325799-00   .00000000      .00000000
.0000 000      .00000000
.00000000      .00000000      .00000000      .11880413+00   .88119587-00   .00000000      .00000000
.00000000      .00000000
.00000000      .00000000      .00000000      .11880413+00   .88119587-00   .00000000      .00000000
.00000000      .00000000
.00000000      .00000000      .00000000      .11880413+00   .88119587-00   .00000000      .00000000
.00000000      .00000000
                              ELEMENT MASS FRACTION PROFILES
.31674202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.3167 202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.3167 202-00   .68325799-00   .00000000
.11880413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.31674202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.3167 202-00   .68325799-00   .00000000
.11880413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.31674202-00   .68325799-00   .00000000
.3167 202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.1188 413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.31674202-00   .68325799-00   .00000000
.3167 202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.1188 413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.3167 202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.31674202-00   .68325799-00   .00000000
.1188 413+00   .88119587-00   .00000000
.11880413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.1188 413+00   .88119587-00   .00000000
.3167 202-00   .68325799-00   .00000000
```

```
.31674202-00      .68325799-00      .00000000
.3167 202-00      .68325799-00      .00000000
.11880413+00      .88119587-00      .00000000
.11880413+00      .88119587-00      .00000000
.1188 413+00      .88119587-00      .00000000




                                                      * * * INJECTION INPUT * * *

  NELRM    DCHAMB      TO        TH          EM
   72      3.650    530.00    530.00       .700

 WT        PO          FM
  3.4500  300.0000     .8000
                                                      * * * PERFORMANCE INPUT * * *

  RT         RE        XN      PERBEL     ENTHO      ENTHH
  .8600    1.6550    2.7000  100.0000     .0000      .0000

 OFINPT     PEF       TTT
 4.0000    -.0000    -.0000
```

```
          HYDROGEN/OXYGEN THRUST CHAMBER(K.M.S. UNITS)     EQUILB  CHEMISTRY     AXISYM  FLOW     RW SPC WALL
          X=    .1322397-00 METERS         TES  T CA  SE    AERO  JET   H2/O  2 15   00 L  B TH  RUST   ENG  INE          PAGE  16
       P(N/M**2)      LEWIS NUMBER PRANDTL NUMBER  VISC(KG/M-SC)   STEP SIZE(M)      STEPS
       .1609288+07      .1000000+01     .1000000+01     .4455536-01     .3880262-03        585
  PT      PSI            VEL(M/S)        T(DEG K)        CP J/KG-K      RHO KG/M3       R(M)            RH/-U         MOLECULAR WT   PT
   1    -.000000        .118923+C4      .321895+04      .450174+04      .617910-00      .000000         .734835+03      .102762+02     1
   2    .144036-01      .118922+C4      .321928+04      .450137+04      .617923-00      .531341-03      .734844+03      .102774+02     2
   3    .288072-01      .118918+C4      .322025+04      .450026+04      .617961-00      .106267-02      .734869+03      .102812+02     3
   4    .432108-01      .118913+C4      .322182+04      .449845+04      .618024-00      .159399-02      .734910+03      .102872+02     4
   5    .576144-01      .118906+C4      .322393+04      .449603+04      .618108-00      .212528-02      .734965+03      .102954+02     5
   6    .720180-01      .118897+C4      .322651+04      .449307+04      .618211-00      .265653-02      .735033+03      .103053+02     6
   7    .864216-01      .118887+C4      .322946+04      .448970+04      .618329-00      .318775-02      .735110+03      .103167+02     7
   8    .100825+00      .118875+C4      .323268+04      .448602+04      .618450-00      .371892-02      .735195+03      .103292+02     8
   9    .115229+00      .118864+C4      .323606+04      .448216+04      .618594-00      .425005-02      .735284+03      .103422+02     9
  10    .129632-00      .118852+C4      .323950+04      .447824+04      .618732-00      .478114-02      .735375+03      .103555+02    10
  11    .144036-00      .118840+C4      .324290+04      .447438+04      .618869-00      .531218-02      .735464+03      .103687+02    11
  12    .158440-00      .118829+C4      .324617+04      .447067+04      .619001-00      .584317-02      .735551+03      .103814+02    12
  13    .172843-00      .118818+C4      .324924+04      .446720+04      .619125-00      .637412-02      .735633+03      .103933+02    13
  14    .187247-00      .118808+C4      .325204+04      .446402+04      .619239-00      .690504-02      .735707+03      .104041+02    14
  15    .201650-00      .118800+C4      .325455+04      .446119+04      .619340-00      .743593-02      .735774+03      .104139+02    15
  16    .216054-00      .118792+C4      .325673+04      .445873+04      .619429-00      .796679-02      .735832+03      .104223+02    16
  17    .230458-00      .118786+C4      .325858+04      .445665+04      .619504-00      .849763-02      .735882+03      .104295+02    17
  18    .244861-00      .118780+C4      .326010+04      .445493+04      .619566-00      .902845-02      .735922+03      .104354+02    18
  19    .259265-00      .118776+C4      .326132+04      .445357+04      .619615-00      .955927-02      .735955+03      .104402+02    19
  20    .273668-00      .118773+C4      .326226+04      .445251+04      .619653-00      .100901-01      .735980+03      .104438+02    20
  21    .288072-00      .118770+C4      .326295+04      .445174+04      .619682-00      .106209-01      .735998+03      .104465+02    21
  22    .302476-00      .118769+C4      .326343+04      .445120+04      .619701-00      .111517-01      .736011+03      .104484+02    22
  23    .316879-00      .118768+C4      .326374+04      .445085+04      .619714-00      .116825-01      .736019+03      .104496+02    23
  24    .331283-00      .118767+C4      .326391+04      .445066+04      .619721-00      .122133-01      .736023+03      .104502+02    24
  25    .345686-00      .118767+C4      .326398+04      .445058+04      .619724-00      .127441-01      .736025+03      .104505+02    25
  26    .360090-00      .118766+C4      .326398+04      .445058+04      .619724-00      .132750-01      .736024+03      .104505+02    26
  27    .374494-00      .118766+C4      .326394+04      .445062+04      .619722-00      .138058-01      .736022+03      .104503+02    27
  28    .388897-00      .118766+C4      .326388+04      .445069+04      .619719-00      .143366-01      .736018+03      .104501+02    28
  29    .403301-00      .118766+C4      .326381+04      .445077+04      .619716-00      .148675-01      .736014+03      .104498+02    29
  30    .417704-00      .118766+C4      .326376+04      .445083+04      .619714-00      .153983-01      .736010+03      .104496+02    30
  31    .432108-00      .118766+C4      .326373+04      .445087+04      .619712-00      .159292-01      .736005+03      .104495+02    31
  32    .446512-00      .118765+C4      .326372+04      .445087+04      .619712-00      .164601-01      .735999+03      .104495+02    32
  33    .460915-00      .118764+C4      .326375+04      .445084+04      .619712-00      .169910-01      .735993+03      .104496+02    33
  34    .475319-00      .118762+C4      .326382+04      .445077+04      .619714-00      .175219-01      .735985+03      .104498+02    34
  35    .489723-00      .118760+C4      .326392+04      .445067+04      .619717-00      .180528-01      .735976+03      .104502+02    35
  36    .504126-00      .118757+C4      .326405+04      .445052+04      .619721-00      .185837-01      .735964+03      .104507+02    36
  37    .518530-00      .118754+C4      .326422+04      .445035+04      .619726-00      .191146-01      .735948+03      .104513+02    37
  38    .532933-00      .118749+C4      .326441+04      .445015+04      .619731-00      .196455-01      .735926+03      .104520+02    38
  39    .547337-00      .118744+C4      .326463+04      .444992+04      .619737-00      .201765-01      .735898+03      .104528+02    39
  40    .561741-00      .118736+C4      .326486+04      .444968+04      .619742-00      .207074-01      .735858+03      .104536+02    40
```

```
41   .576144-00   .118727+04   .326511+04   .444944+04   .619747-00   .212385-01   .735004+03   .104545+02  41
42   .5 0548-00   .118714+04   .326536+04   .444919+04   .619750-00   .217695-01   .735727+03   .104554+02  42
43   .604951-00   .118695+04   .326563+04   .444895+04   .619750-00   .223006-01   .735615+03   .104562+02  43
44   .619355-00   .118669+04   .326590+04   .444873+04   .619745-00   .228318-01   .735445+03   .104570+02  44
45   .6 3759-00   .118628+04   .326619+04   .444854+04   .619733-00   .233632-01   .735179+03   .104577+02  45
46   .648162-00   .118563+04   .326652+04   .444838+04   .619706-00   .238948-01   .734743+03   .104583+02  46
47   .6 2566-00   .118453+04   .326694+04   .444828+04   .619656-00   .244268-01   .734001+03   .104588+02  47
48   .676969-00   .118260+04   .326755+04   .444824+04   .619563-00   .249596-01   .732693+03   .104592+02  48
49   .691373-00   .117906+04   .326854+04   .444829+04   .619387-00   .254936-01   .730296+03   .104594+02  49
50   .705777-00   .117231+04   .327035+04   .444849+04   .619050-00   .260301-01   .725716+03   .104595+02  50
51   .720180-00   .115865+04   .327391+04   .444897+04   .618372-00   .265714-01   .716474+03   .104594+02  51

PT  STATIC H     TOTAL H      TOTAL SEN. H  GAMMA       STOICH 02    MACH NO     P TOTAL      T TOTAL     PT
 1   .343362+06   .105049+07   .130534+08   .121911+01   .193702+01   .667405-00   .209780+07   .337603+04   1
 2   .343280+06   .105040+07   .130536+08   .121910+01   .193672+01   .667409-00   .209780+07   .337637+04   2
 3   .3 3036+06   .105011+07   .130544+08   .121906+01   .193585+01   .667419-00   .209780+07   .337737+04   3
 4   .342640+06   .104965+07   .130557+08   .121901+01   .193443+01   .667437-00   .209781+07   .337898+04   4
 5   .342108+06   .104903+07   .130574+08   .121895+01   .193253+01   .667460-00   .209782+07   .338116+04   5
 6   .341460+06   .104828+07   .130594+08   .121886+01   .193021+01   .667488-00   .209784+07   .338382+04   6
 7   .340720+06   .104742+07   .130618+08   .121877+01   .192756+01   .667521-00   .209785+07   .338687+04   7
 8   .339914+06   .104648+07   .130644+08   .121867+01   .192467+01   .667556-00   .209787+07   .339019+04   8
 9   .339069+06   .104550+07   .130671+08   .121856+01   .192165+01   .667593-00   .209789+07   .339367+04   9
10   .338212+06   .104450+07   .130698+08   .121845+01   .191858+01   .667631-00   .209790+07   .339722+04  10
11   .3 7367+06   .104352+07   .130725+08   .121834+01   .191556+01   .667668-00   .209792+07   .340072+04  11
12   .336556+06   .104257+07   .130751+08   .121824+01   .191266+01   .667704-00   .209794+07   .340409+04  12
13   .3 5797+06   .104160+07   .130776+08   .121814+01   .190995+01   .667738-00   .209796+07   .340725+04  13
14   .335104+06   .104088+07   .130798+08   .121805+01   .190747+01   .667769-00   .209797+07   .341015+04  14
15   .334486+06   .104015+07   .130818+08   .121797+01   .190526+01   .667797-00   .209798+07   .341273+04  15
16   .333949+06   .103953+07   .130835+08   .121790+01   .190334+01   .667821-00   .209800+07   .341498+04  16
17   .333495+06   .103900+07   .130849+08   .121784+01   .190171+01   .667841-00   .209801+07   .341688+04  17
18   .333120+06   .103856+07   .130861+08   .121779+01   .190030+01   .667858-00   .209801+07   .341845+04  18
19   .332822+06   .103821+07   .130871+08   .121776+01   .189931+01   .667872-00   .209802+07   .341971+04  19
20   .332592+06   .103794+07   .130878+08   .121773+01   .189849+01   .667882-00   .209803+07   .342067+04  20
21   .3 2423+06   .103774+07   .130884+08   .121770+01   .189789+01   .667890-00   .209803+07   .342138+04  21
22   .332306+06   .103761+07   .130888+08   .121769+01   .189747+01   .667895-00   .209803+07   .342188+04  22
23   .332231+06   .103752+07   .130890+08   .121768+01   .189720+01   .667898-00   .209803+07   .342220+04  23
24   .332189+06   .103747+07   .130891+08   .121767+01   .189705+01   .667900-00   .209803+07   .342237+04  24
25   .332173+06   .103745+07   .130892+08   .121767+01   .189698+01   .667900-00   .209803+07   .342245+04  25
26   .332174+06   .103745+07   .130892+08   .121767+01   .189698+01   .667899-00   .209803+07   .342245+04  26
27   .332187+06   .103746+07   .130892+08   .121767+01   .189702+01   .667898-00   .209803+07   .342241+04  27
28   .332205+06   .103748+07   .130891+08   .121768+01   .189707+01   .667896-00   .209803+07   .342234+04  28
29   .332225+06   .103750+07   .130891+08   .121765+01   .189713+01   .667893-00   .209802+07   .342227+04  29
30   .332244+06   .103751+07   .130890+08   .121768+01   .189718+01   .667890-00   .209802+07   .342222+04  30
31   .332259+06   .103752+07   .130890+08   .121768+01   .189721+01   .667886-00   .209801+07   .342218+04  31
32   .332270+06   .103752+07   .130890+08   .121768+01   .189721+01   .667881-00   .209801+07   .342218+04  32
33   .332276+06   .103752+07   .130890+06   .121768+01   .189719+01   .667875-00   .209800+07   .342220+04  33
34   .332277+06   .103750+07   .130891+08   .121768+01   .189713+01   .667868-00   .209798+07   .342227+04  34
35   .332274+06   .103747+07   .130891+08   .121767+01   .189705+01   .667859-00   .209797+07   .342237+04  35
36   .332270+06   .103743+07   .130892+08   .121767+01   .189694+01   .667846-00   .209795+07   .342250+04  36
37   .332266+06   .103739+07   .130894+08   .121766+01   .189680+01   .667831-00   .209792+07   .342266+04  37
38   .332268+06   .103734+07   .130895+08   .121766+01   .189664+01   .667810-00   .209788+07   .342285+04  38
39   .332278+06   .103728+07   .130897+08   .121765+01   .189646+01   .667783-00   .209783+07   .342306+04  39
40   .332304+06   .103722+07   .130898+08   .121765+01   .189628+01   .667746-00   .209777+07   .342328+04  40
41   .332355+06   .103715+07   .130900+08   .121764+01   .189608+01   .667696-00   .209769+07   .342351+04  41
42   .332445+06   .103709+07   .130902+06   .121763+01   .189589+01   .667627-00   .209757+07   .342374+04  42
43   .332598+06   .103703+07   .130903+08   .121762+01   .189570+01   .667527-00   .209741+07   .342396+04  43
44   .3 2854+06   .103697+07   .130905+08   .121762+01   .189552+01   .667377-00   .209716+07   .342417+04  44
45   .333283+06   .103692+07   .130907+06   .121761+01   .189536+01   .667144-00   .209678+07   .342436+04  45
46   .334012+06   .103687+07   .130908+08   .121760+01   .189522+01   .666765-00   .209616+07   .342452+04  46
47   .335279+06   .103684+07   .130909+08   .121760+01   .189511+01   .666121-00   .209511+07   .342465+04  47
```

```
 48   .337540+06   .103681+07   .130910+08   .121759+01   .189503+01   .664985-00   .209327+07   .342475+04  48
 49   .341698+06   .103679+07   .130910+08   .121758+01   .189498+01   .662906-00   .208991+07   .342480+04  49
 50   .339635+06   .103679+07   .130910+08   .121757+01   .189496+01   .658932-00   .208354+07   .342481+04  50
 51   .335562+06   .103679+07   .130910+08   .121754+01   .189497+01   .650904-00   .207082+07   .342478+04  51
P=    .15 8244+02 ATM   X/D=    .1426380+01   YW=    .2602974-01  YW=    .2602973-01  DYWDX=   -.4267700-00
                                      BULK MASS'MEAN AVERAGED QUANTITIES

    TBAR            UBAR           WBAR           PHIBAR          GAMMA-BAR       MBAR            ELEM H          CPBAR
  3.2626700+03    1.1865205+03   1.0444169+01   1.8984094+00    1.2177211+00    6.6717359-01    1.9302636-01    4.4524502+03
    VISC             HT             O/F RATIO       HE              RHO'U           TOTAL P         TOTAL T
  .4455536-01     .1037915+07    .4180609+01     .3339998+06     .7351637+03     .2096872+07     .3420766+04
ELEM O=   .80696777-00  HS=    .13087908+08  RHO=    .61959632-00  VISC=    .4455536-01  GAMMA=    .1217721+01
IMP FUNC=     .5282268+04  NET THRUST=    -.8732670+04
 PT      H              O              H20            H2             O2             OH             HO2            H2O2        PT
  1   .000000        .000000        .905093-00     .949015-01     .000000        .000000        .000000        .000000        1
  2   .000000        .000000        .905120-00     .948745-01     .000000        .000000        .000000        .000000        2
  3   .000000        .000000        .905200-00     .947944-01     .000000        .000000        .000000        .000000        3
  4   .000000        .000000        .905330-00     .946644-01     .000000        .000000        .000000        .000000        4
  5   .000000        .000000        .905504-00     .944897-01     .000000        .000000        .000000        .000000        5
  6   .000000        .000000        .905717-00     .942767-01     .000000        .000000        .000000        .000000        6
  7   .000000        .000000        .905961-00     .940334-01     .000000        .000000        .000000        .000000        7
  8   .000000        .000000        .906226-00     .937684-01     .000000        .000000        .000000        .000000        8
  9   .000000        .000000        .906504-00     .934906-01     .000000        .000000        .000000        .000000        9
 10   .000000        .000000        .906786-00     .932085-01     .000000        .000000        .000000        .000000       10
 11   .000000        .000000        .907064-00     .929303-01     .000000        .000000        .000000        .000000       11
 12   .000000        .000000        .907331-00     .926632-01     .000000        .000000        .000000        .000000       12
 13   .000000        .000000        .907581-00     .924130-01     .000000        .000000        .000000        .000000       13
 14   .000000        .000000        .907809-00     .921845-01     .000000        .000000        .000000        .000000       14
 15   .000000        .000000        .908013-00     .919808-01     .000000        .000000        .000000        .000000       15
 16   .000000        .000000        .908190-00     .918037-01     .000000        .000000        .000000        .000000       16
 17   .000000        .000000        .908340-00     .916537-01     .000000        .000000        .000000        .000000       17
 18   .000000        .000000        .908464-00     .915302-01     .000000        .000000        .000000        .000000       18
 19   .000000        .000000        .908562-00     .914317-01     .000000        .000000        .000000        .000000       19
 20   .000000        .000000        .908638-00     .913558-01     .000000        .000000        .000000        .000000       20
 21   .000000        .000000        .908694-00     .913000-01     .000000        .000000        .000000        .000000       21
 22   .000000        .000000        .908733-00     .912612-01     .000000        .000000        .000000        .000000       22
 23   .000000        .000000        .908758-00     .912363-01     .000000        .000000        .000000        .000000       23
 24   .000000        .000000        .908772-00     .912224-01     .000000        .000000        .000000        .000000       24
 25   .000000        .000000        .908778-00     .912166-01     .000000        .000000        .000000        .000000       25
 26   .000000        .000000        .908778-00     .912165-01     .000000        .000000        .000000        .000000       26
 27   .000000        .000000        .908774-00     .912199-01     .000000        .000000        .000000        .000000       27
 28   .000000        .000000        .908769-00     .912250-01     .000000        .000000        .000000        .000000       28
 29   .000000        .000000        .908764-00     .912303-01     .000000        .000000        .000000        .000000       29
 30   .000000        .000000        .908759-00     .912347-01     .000000        .000000        .000000        .000000       30
 31   .000000        .000000        .908756-00     .912374-01     .000000        .000000        .000000        .000000       31
 32   .000000        .000000        .908756-00     .912378-01     .000000        .000000        .000000        .000000       32
 33   .000000        .000000        .908758-00     .912356-01     .000000        .000000        .000000        .000000       33
 34   .000000        .000000        .908763-00     .912306-01     .000000        .000000        .000000        .000000       34
 35   .000000        .000000        .908771-00     .912223-01     .000000        .000000        .000000        .000000       35
 36   .000000        .000000        .908782-00     .912124-01     .000000        .000000        .000000        .000000       36
 37   .000000        .000000        .908794-00     .911997-01     .000000        .000000        .000000        .000000       37
 38   .000000        .000000        .908809-00     .911859-01     .000000        .000000        .000000        .000000       38
 39   .000000        .000000        .908826-00     .911687-01     .000000        .000000        .000000        .000000       39
 40   .000000        .000000        .908843-00     .911513-01     .000000        .000000        .000000        .000000       40
 41   .000000        .000000        .908861-00     .911333-01     .000000        .000000        .000000        .000000       41
 42   .000000        .000000        .908879-00     .911153-01     .000000        .000000        .000000        .000000       42
 43   .000000        .000000        .908897-00     .910979-01     .000000        .000000        .000000        .000000       43
 44   .000000        .000000        .908913-00     .910814-01     .000000        .000000        .000000        .000000       44
 45   .000000        .000000        .908928-00     .910666-01     .000000        .000000        .000000        .000000       45
 46   .000000        .000000        .908941-00     .910533-01     .000000        .000000        .000000        .000000       46
```

```
47   .000000   .000000   .908951-00  .910434-01  .000000   .000000   .000000   .000000   47
48   .000000   .000000   .908959-00  .910358-01  .000000   .000000   .000000   .000000   48
49   .000000   .000000   .908963-00  .910312-01  .000000   .000000   .000000   .000000   49
50   .000000   .000000   .908965-00  .910296-01  .000000   .000000   .000000   .000000   50
51   .000000   .000000   .908963-00  .910312-01  .000000   .000000   .000000   .000000   51
 1   .000000   .000000   .516257-00  .483743-00  .000000   .000000   .000000   .000000    1
 2   .000000   .000000   .516336-00  .483664-00  .000000   .000000   .000000   .000000    2
 3   .000000   .000000   .516569-00  .483431-00  .000000   .000000   .000000   .000000    3
 4   .000000   .000000   .516947-00  .483053-00  .000000   .000000   .000000   .000000    4
 5   .000000   .000000   .517457-00  .482543-00  .000000   .000000   .000000   .000000    5
 6   .000000   .000000   .518079-00  .481921-00  .000000   .000000   .000000   .000000    6
 7   .000000   .000000   .518791-00  .481209-00  .000000   .000000   .000000   .000000    7
 8   .000000   .000000   .519569-00  .480431-00  .000000   .000000   .000000   .000000    8
 9   .000000   .000000   .520386-00  .479614-00  .000000   .000000   .000000   .000000    9
10   .000000   .000000   .521218-00  .478782-00  .000000   .000000   .000000   .000000   10
11   .000000   .000000   .522040-00  .477960-00  .000000   .000000   .000000   .000000   11
12   .000000   .000000   .522832-00  .477168-00  .000000   .000000   .000000   .000000   12
13   .000000   .000000   .523575-00  .476425-00  .000000   .000000   .000000   .000000   13
14   .000000   .000000   .524255-00  .475745-00  .000000   .000000   .000000   .000000   14
15   .000000   .000000   .524863-00  .475137-00  .000000   .000000   .000000   .000000   15
16   .000000   .000000   .525392-00  .474608-00  .000000   .000000   .000000   .000000   16
17   .000000   .000000   .525841-00  .474159-00  .000000   .000000   .000000   .000000   17
18   .000000   .000000   .526211-00  .473789-00  .000000   .000000   .000000   .000000   18
19   .000000   .000000   .526507-00  .473493-00  .000000   .000000   .000000   .000000   19
20   .000000   .000000   .526734-00  .473265-00  .000000   .000000   .000000   .000000   20
21   .000000   .000000   .526902-00  .473098-00  .000000   .000000   .000000   .000000   21
22   .000000   .000000   .527019-00  .472981-00  .000000   .000000   .000000   .000000   22
23   .000000   .000000   .527094-00  .472906-00  .000000   .000000   .000000   .000000   23
24   .000000   .000000   .527136-00  .472864-00  .000000   .000000   .000000   .000000   24
25   .000000   .000000   .527153-00  .472847-00  .000000   .000000   .000000   .000000   25
26   .000000   .000000   .527153-00  .472847-00  .000000   .000000   .000000   .000000   26
27   .000000   .000000   .527143-00  .472857-00  .000000   .000000   .000000   .000000   27
28   .000000   .000000   .527128-00  .472872-00  .000000   .000000   .000000   .000000   28
29   .000000   .000000   .527112-00  .472888-00  .000000   .000000   .000000   .000000   29
30   .000000   .000000   .527098-00  .472902-00  .000000   .000000   .000000   .000000   30
31   .000000   .000000   .527090-00  .472910-00  .000000   .000000   .000000   .000000   31
32   .000000   .000000   .527089-00  .472911-00  .000000   .000000   .000000   .000000   32
33   .000000   .000000   .527096-00  .472904-00  .000000   .000000   .000000   .000000   33
34   .000000   .000000   .527111-00  .472889-00  .000000   .000000   .000000   .000000   34
35   .000000   .000000   .527134-00  .472866-00  .000000   .000000   .000000   .000000   35
36   .000000   .000000   .527165-00  .472835-00  .000000   .000000   .000000   .000000   36
37   .000000   .000000   .527204-00  .472796-00  .000000   .000000   .000000   .000000   37
38   .000000   .000000   .527248-00  .472752-00  .000000   .000000   .000000   .000000   38
39   .000000   .000000   .527297-00  .472703-00  .000000   .000000   .000000   .000000   39
40   .000000   .000000   .527349-00  .472651-00  .000000   .000000   .000000   .000000   40
41   .000000   .000000   .527403-00  .472597-00  .000000   .000000   .000000   .000000   41
42   .000000   .000000   .527458-00  .472542-00  .000000   .000000   .000000   .000000   42
43   .000000   .000000   .527510-00  .472490-00  .000000   .000000   .000000   .000000   43
44   .000000   .000000   .527560-00  .472440-00  .000000   .000000   .000000   .000000   44
45   .000000   .000000   .527604-00  .472396-00  .000000   .000000   .000000   .000000   45
46   .000000   .000000   .527643-00  .472357-00  .000000   .000000   .000000   .000000   46
47   .000000   .000000   .527674-00  .472326-00  .000000   .000000   .000000   .000000   47
48   .000000   .000000   .527697-00  .472303-00  .000000   .000000   .000000   .000000   48
49   .000000   .000000   .527711-00  .472289-00  .000000   .000000   .000000   .000000   49
50   .000000   .000000   .527716-00  .472284-00  .000000   .000000   .000000   .000000   50
51   .000000   .000000   .527711-00  .472289-00  .000000   .000000   .000000   .000000   51
PT  TRACE(MASS) TRACE(MOLE) FLEM H     ELEM O      O/F RATIO   SUM SPECIE  TIME(SEC)   MDOT(KG/SC) PT
 1   .000000   .000000   .196182-00  .803817-00  .407728+01  .999994-00  .372628-03  .000000    1
 2   .000000   .000000   .196158-00  .803836-00  .409790+01  .999994-00  .372573-03  .651767-03  2
 3   .000000   .000000   .196087-00  .803907-00  .409975+01  .999994-00  .372411-03  .240707-02  3
 4   .000000   .000000   .195971-00   .804023-00   .410276+01   .999994-00   .372148-03   .586590-02    4
 5   .000000   .000000   .195816-00   .804178-00   .410680+01   .999994-00   .371798-03   .104283-01    5
 6   .000000   .000000   .195627-00   .804367-00   .411174+01   .999994-00   .371375-03   .162942-01    6
 7   .000000   .000000   .195411-00   .804583-00   .411739+01   .999994-00   .371637-03   .234636-01    7
 8   .000000   .000000   .195176-00   .804819-00   .412356+01   .999994-00   .371317-03   .319366-01    8
 9   .000000   .000000   .194929-00   .805065-00   .413005+01   .999994-00   .371111-03   .417131-01    9
10   .000000   .000000   .194678-00   .805316-00   .413665+01   .999994-00   .371005-03   .527931-01   10
11   .000000   .000000   .194431-00   .805563-00   .414318+01   .999994-00   .370994-03   .651767-01   11
12   .000000   .000000   .194194-00   .805800-00   .414946+01   .999994-00   .371072-03   .788638-01   12
13   .000000   .000000   .193972-00   .806022-00   .415536+01   .999994-00   .370525-03   .938544-01   13
14   .000000   .000000   .193769-00   .806225-00   .416076+01   .999994-00   .370558-03   .110149+00   14
15   .000000   .000000   .193588-00   .806406-00   .416558+01   .999994-00   .370443-03   .127746-00   15
16   .000000   .000000   .193431-00   .806563-00   .416978+01   .999994-00   .370939-03   .146647-00   16
17   .000000   .000000   .193297-00   .806697-00   .417334+01   .999994-00   .370822-03   .166852-00   17
18   .000000   .000000   .193188-00   .806806-00   .417628+01   .999994-00   .370793-03   .188361-00   18
19   .000000   .000000   .193100-00   .806894-00   .417863+01   .999994-00   .370844-03   .211172-00   19
20   .000000   .000000   .193033-00   .806961-00   .418043+01   .999994-00   .370973-03   .235288-00   20
21   .000000   .000000   .192983-00   .807011-00   .418176+01   .999994-00   .370468-03   .260707-00   21
22   .000000   .000000   .192949-00   .807045-00   .418269+01   .999994-00   .370541-03   .287429-00   22
23   .000000   .000000   .192927-00   .807067-00   .418328+01   .999994-00   .370454-03   .315455-00   23
24   .000000   .000000   .192914-00   .807080-00   .418362+01   .999994-00   .370966-03   .344784-00   24
25   .000000   .000000   .192909-00   .807085-00   .418375+01   .999994-00   .370856-03   .375418-00   25
26   .000000   .000000   .192909-00   .807085-00   .418376+01   .999994-00   .370829-03   .407354-00   26
27   .000000   .000000   .192912-00   .807082-00   .418367+01   .999994-00   .370878-03   .440594-00   27
28   .000000   .000000   .192917-00   .807077-00   .418355+01   .999994-00   .371006-03   .475138-00   28
29   .000000   .000000   .192921-00   .807072-00   .418343+01   .999994-00   .370501-03   .510985-00   29
30   .000000   .000000   .192925-00   .807068-00   .418332+01   .999994-00   .370577-03   .548136-00   30
31   .000000   .000000   .192928-00   .807066-00   .418326+01   .999994-00   .370495-03   .586590-00   31
32   .000000   .000000   .192928-00   .807066-00   .418325+01   .999994-00   .371011-03   .626348-00   32
33   .000000   .000000   .192926-00   .807068-00   .418330+01   .999994-00   .370905-03   .667409-00   33
34   .000000   .000000   .192922-00   .807072-00   .418342+01   .999994-00   .370886-03   .709774-00   34
35   .000000   .000000   .192915-00   .807079-00   .418360+01   .999994-00   .370948-03   .753442-00   35
36   .000000   .000000   .192906-00   .807088-00   .418385+01   .999994-00   .371096-03   .798414-00   36
37   .000000   .000000   .192894-00   .807100-00   .418416+01   .999994-00   .370621-03   .844689-00   37
38   .000000   .000000   .192881-00   .807113-00   .418451+01   .999994-00   .370739-03   .892268-00   38
39   .000000   .000000   .192867-00   .807128-00   .418490+01   .999994-00   .370713-03   .941151-00   39
40   .000000   .000000   .192851-00   .807143-00   .418531+01   .999994-00   .371300-03   .991337-00   40
41   .000000   .000000   .192835-00   .807159-00   .418574+01   .999995-00   .371288-03   .104283+01   41
42   .000000   .000000   .192819-00   .807175-00   .418617+01   .999995-00   .371391-03   .109562+01   42
43   .000000   .000000   .192804-00   .807191-00   .418659+01   .999995-00   .371613-03   .114972+01   43
44   .000000   .000000   .192789-00   .807205-00   .418698+01   .999995-00   .371968-03   .120512+01   44
45   .000000   .000000   .192776-00   .807218-00   .418734+01   .999995-00   .371763-03   .126182+01   45
46   .000000   .000000   .192765-00   .807230-00   .418764+01   .999995-00   .372235-03   .131983+01   46
47   .000000   .000000   .192756-00   .807239-00   .418789+01   .999994-00   .372671-03   .137914+01   47
48   .000000   .000000   .192749-00   .807246-00   .418807+01   .999994-00   .373871-03   .143975+01   48
49   .000000   .000000   .192745-00   .807250-00   .418818+01   .999994-00   .374687-03   .150167+01   49
50   .000000   .000000   .192743-00   .807251-00   .418822+01   .999994-00   .375946-03   .156489+01   50
51   .000000   .000000   .192745-00   .807250-00   .418819+01   .999994-00   .377811-03   .162942+01   51
```

```
THE PERFORMANCE OF THE ENGINE AT TIME=   .000 SEC IS 386.48

THE THRUST IS  1387.96 LBS

THE MIXTURE RATIO IS  4.18
THE ISP AFTER COMBUSTION BUT WITHOUT NOZZLE LOSSES IS =  407.65
THEORETICAL ISP IS = 411.22
KINETIC ISP = 407.30
LOSS DUE TO BOUNDARY LAYER =   2.13
LOSS DUE TO DIVERGENCE     =  18.69
```

Sample Case II

Sample Case II is a dynamics computation for the baseline Aerojet engine. The first data card required is the run control card with NDYNA = 1 and NBLFG = 1, all others zero. This card is followed by the dynamics control card (NNODE thru CD). Since a blowdown calculation is specified by NBLFG = 1, the next card defines the blowdown volumes (NBLOW(I)). Then, NNODE cards are required to define the volumes followed by NCONN cards to define the flow passages. The last two cards are the ignition cards (SPTIME thru SPKF) and (SPGAP thru DC).

On the next page is a copy of the Sample Case II input report followed by several samples of the dynamics output.

```
                                                     * * * DYNAMICS INPUT * * *

 NNODE    NCONN    NITER    NCHEK  PCHEK

   7        8      4000       1     1,000

 NVOL    ICOMB      VOL       PRES        TEMP        RM1X


   1       1     48.300       .000    530.000        .000
   2      -0**********        .000    530.000        .000
   3      -0     23.800       .000    530.000        .000
   4      -0     32.800       .000    530.000        .000
   5       3       .467       .000    530.000        .000
   6      -0100000.000     375.000    530.000        .000
   7      -0100000.000     375.000    530.000       1,000

 IADMIT  IADMIT  IRTYPE  CAREA   CCOEFF    TIMON     TIMOFF    TIMOPN    TIMCLS
   1       2       2     2,895    1.000    -.0000    -.0000    -.0000    -,0000
   5       1       0      ,085    1.000    -.0000    -,0000    -,0000    -,0000
   3       1       0      ,438    1.000    -.0000    -,0000    -,0000    -.0000
   4       1       0      ,349    1.000    -.0000    -.0000    -,0000    -,0000
   6       3       0      ,500    1.000     .0280     ,1300     .0100     ,0087
   7       4       0      ,332    1.000     .0260     ,1220     ,0087     ,0074
   6       5       0      ,007    1.000     .0100     ,1300     .0100     .0100
   7       5       0      ,012    1.000     .0000     ,1200     .0100     ,0100

SPTIME    SPKTL     SPKF
 ,0000     ,0500     .0025

 SPGAP    SPARKP    SPARKE     DC
 ,0250   50000.0  100,0000   3.6000




                                                 ** DYNAMICS RESULTS **

                                                  TIME =    ,026193


    ITERATION NO    600
      VOL NO                    PRESSURE      TEMP       ( OX
                                 (PSIA)        (R)

        1 (COMBUSTOR)            13.82      5226,4     93,806
        2                          .01      5380,7     89,020
        3                        13.55      4146,1     88,917
        4                        42.46      1557,7     99.031
        5 (PILOT)               295.96      6376.4     86,754
        6                       374.99       530.0       .000
        7                       374.99       530,0    100.000
      DELT USED     .000025

       THRUST=          ,00       IMPULSE=        .00000        ISP=          ,00
      ACCUM WT FLOW=     .00000

       VOL     1  PRES CHANGE=     4.930PSI PER MS
```

```
                                          ** DYNAMICS RESULTS **

                                             TIME =    .028671


ITERATION NC    620
   VOL NC                  PRESSURE      TEMP       ( OX
                            (PSIA)        (R)

     1 (CCMBUSTOR)          17.16       5190.5     94.218
     2                        .01       5368.6     89.456
     3                      18.43       1531.5     61.420
     4                      59.91       1351.5     99.245
     5 (PILCT)             295.96       6376.4     86.754
     6                     374.99        530.0       .000
     7                     374.99        530.0    100.000
    DELT LSED    .000025

     THRUST=          .00       IMPULSE=        .00000        ISP=            .00
    ACCUM WT FLOW=        .00000

     VOL     1  PRES CHANGE=      8.043PSI PER MS
```

```
                                          ** DYNAMICS RESULTS **

                                             TIME =    .029164


ITERATION NC    640
   VOL NC                  PRESSURE      TEMP       ( OX
                            (PSIA)        (R)

     1 (CCMBUSTOR)          22.08       5434.0     92.102
     2                        .01       5363.6     89.829
     3                      26.41        697.3     35.323
     4                      59.66       1199.5     99.396
     5 (PILCT)             295.95       6376.4     86.754
     6                     374.99        530.0       .000
     7                     374.99        530.0    100.000
    DELT LSED    .000025

     THRUST=          .00       IMPULSE=        .00000        ISP=            .00
    ACCUM WT FLOW=        .00000

     VOL     1  PRES CHANGE=     11.550PSI PER MS
```

```
                                          ** DYNAMICS RESULTS **

                                             TIME =    .029656


ITERATION NO    660
   VOL NC                  PRESSURE      TEMP       ( OX
                            (PSIA)        (R)

     1 (CCMBUSTOR)          29.64       5579.8     88.330
     2                        .01       5380.3     89.878
     3                      36.97        701.9     20.038
     4                      66.48       1086.6     99.505
     5 (PILCT)             295.95       6376.4     86.754
     6                     374.98        530.0       .000
     7                     374.99        530.0    100.000
    DELT LSED    .000025

     THRUST=          .00       IMPULSE=        .00000        ISP=            .00
    ACCUM WT FLOW=        .00000

     VOL     1  PRES CHANGE=     26.789PSI PER MS
```

Sample Case III

Sample Case III is a steady state heat transfer computation for the baseline Aerojet engine using a combination of film and regenerative cooling. The first data card is the run control card with NHEAT = 1. Next comes the heat transfer title card followed by the two control cards (NNODE thru INJFL) and (TSTOP thru PRINT INTERVAL). The next two cards are (TWALLF thru OF) and (RC thru TINJO2). The next NNODE cards define the combustor geometry. Then come the film cooling cards; the first specifies the number of injection points and the following NINJ ones define injection locations, slot heights, and coolant mass flows. The last data cards required to run Sample Case III define the regenerative cooling. One card (HRWD thru XREGEN) specifies the regenerative coolant mass flow, the coolant introduction point, and the number of coolant passages. The remaining cards specify coolant passage geometry at each nodal point.

On the next page is a copy of the Sample Case III input report followed by several samples of the heat transfer output.

```
--ROCKET HEAT TRANSFER MODEL--   AEROJET REGENERATIVELY COOLED-O2/H2 ENGINE

   NNODE   NTYPFL  NFLMFL  NRGNFL  NLFL    INJFL   NCCUNT  NPRINT

    19       0       1       1       0       0      100      10

    T WALL   T SINK   CAPPA     EMISS     PC       O/F       RC       RHO      CP      TINJH2   TINJO2
    DEG F    DEG F                        PSIA               INCHES                     DEG F    DEG F
      70.      70.    0.0052    0.4000    300.0    4.000     0.70    0.3230   0.0920    70.0     70.0

                          WALL              FILM COOLING          REGEN   60 PASSAGES          LINER
                                                                  WDOT =    0.512            EMISS = 0.0

   NODE   STATION     ID      OD          HEIGHT   WDOT           HEIGHT  WIDTH   ETA F          ID       OD

     1      0.0      3.58    4.66         0.011    0.068          0.400  0.050   0.900          0.0      0.0
     2      0.01     3.58    4.66         0.0      0.0            0.400  0.050   0.900          0.0      0.0
     3      0.65     3.58    4.66         0.0      0.0            0.400  0.050   0.900          0.0      0.0
     4      1.30     3.58    4.66         0.0      0.0            0.400  0.050   0.900          0.0      0.0
     5      1.95     3.58    4.66         0.0      0.0            0.400  0.050   0.900          0.0      0.0

     6      2.60     3.58    4.66         0.0      0.0            0.400  0.050   0.900          0.0      0.0
     7      3.25     3.46    4.49         0.0      0.0            0.330  0.050   0.870          0.0      0.0
     8      3.90     3.11    3.97         0.0      0.0            0.280  0.050   0.850          0.0      0.0
     9      4.55     2.55    3.37         0.0      0.0            0.240  0.050   0.830          0.0      0.0
    10      5.20     2.12    2.81         0.0      0.0            0.200  0.050   0.800          0.0      0.0

    11      5.80     1.86    2.59         0.0      0.0            0.200  0.050   0.800          0.0      0.0
    12      6.88     3.02    3.89         0.0      0.0            0.250  0.050   0.820          0.0      0.0
    13      7.96     4.40    5.53         0.0      0.0            0.350  0.050   0.880          0.0      0.0
    14      9.04     5.53    6.65         0.001    0.048          0.400  0.050   0.900          0.0      0.0
    15      9.05     5.54    6.65         0.0      0.0            0.0    0.0     0.0            0.0      0.0

    16     10.12     6.57    6.67         0.0      0.0            0.0    0.0     0.0            0.0      0.0
    17     11.20     7.43    7.53         0.0      0.0            0.0    0.0     0.0            0.0      0.0
    18     12.27     8.28    8.38         0.0      0.0            0.0    0.0     0.0            0.0      0.0
    19     12.28     8.29    8.39         0.0      0.0            0.0    0.0     0.0            0.0      0.0
```

```
--ROCKET HEAT TRANSFER MODEL--   AEROJET REGENERATIVELY COOLED O2/H2 ENGINE

      ITERATION NO.   CORE O/F   CORE TEMP    WT. FLOW
                                  DEG F        LB/SEC
                 70     4.000     5185.3       3.057

      NODE      STATION    T WALL    T FILM    T BULK  T LINER    T H2
                                               REGEN             LINER


         1       0.0       141.5     141.4     141.4      0.0       0.0
         2       0.01      141.5     144.0     141.4      0.0       0.0
         3       0.65      144.3     308.2     141.4      0.0       0.0
         4       1.30      147.1     472.9     140.8      0.0       0.0
         5       1.95      149.5     633.9     139.6      0.0       0.0


         6       2.60      151.6     791.0     137.7      0.0       0.0
         7       3.25      155.4     946.5     135.5      0.0       0.0
         8       3.90      162.3    1110.5     132.8      0.0       0.0
         9       4.55      174.9    1297.5     129.5      0.0       0.0
        10       5.20      193.2    1500.0     125.2      0.0       0.0


        11       5.80      198.0    1699.7     119.6      0.0       0.0
        12       6.88      182.2    2016.9     105.9      0.0       0.0
        13       7.96      149.0    2274.2      87.7      0.0       0.0
        14       9.04      168.0    2461.9      70.0      0.0       0.0
        15       9.05      168.2     143.3       0.0      0.0       0.0


        16      10.12      216.1     327.7       0.0      0.0       0.0
        17      11.20      337.5     500.3       0.0      0.0       0.0
        18      12.27      284.5     661.7       0.0      0.0       0.0
        19      12.28      284.5     696.2       0.0      0.0       0.0
```

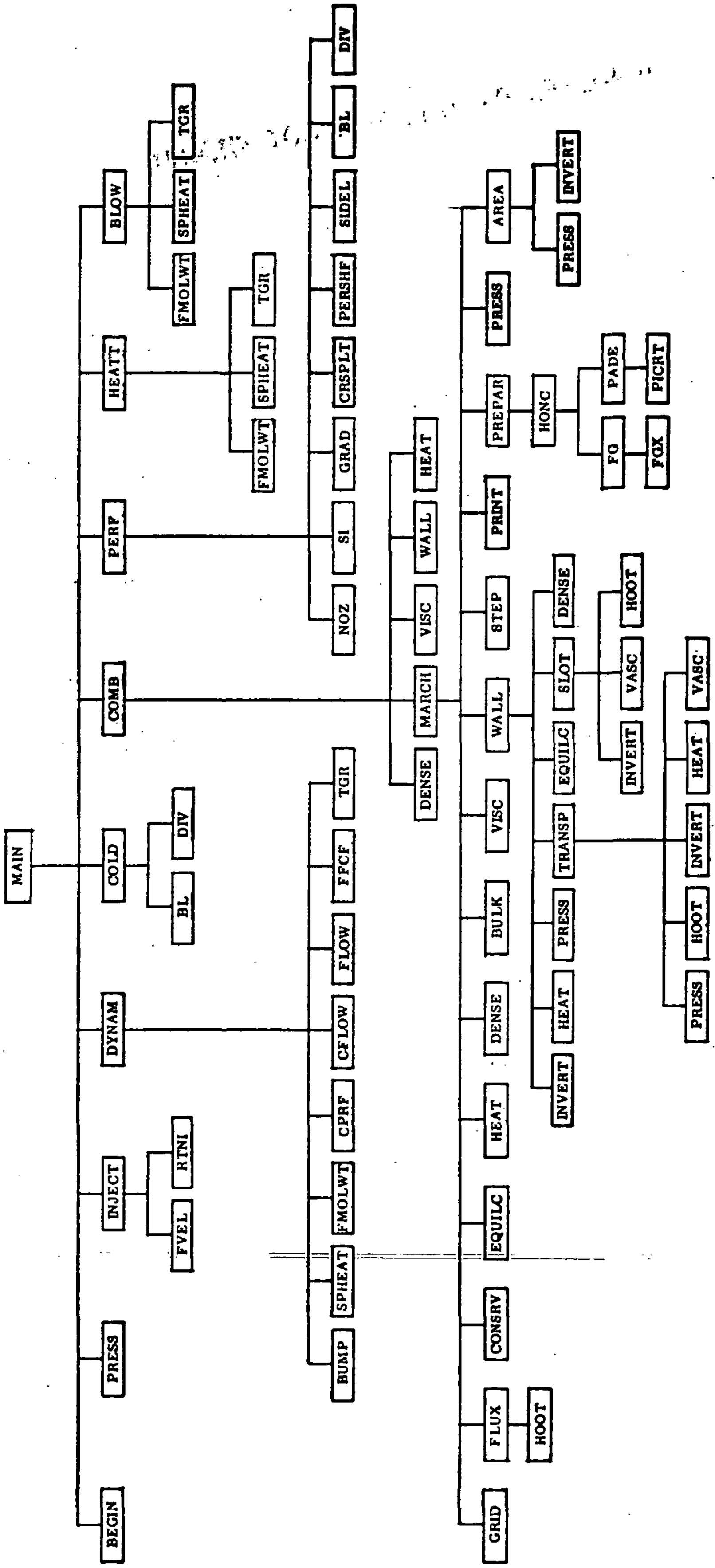
```
--ROCKET HEAT TRANSFER MODEL--   AEROJET REGENERATIVELY COOLED O2/H2 ENGINE

      ITERATION NO.   CORE O/F   CORE TEMP    WT. FLOW
                                  DEG F        LB/SEC
                 50     4.000     5185.3       3.057

      NODE      STATION    T WALL    T FILM    T BULK  T LINER    T H2
                                               REGEN             LINER


         1       0.0       140.5     140.8     140.9      0.0       0.0
         2       0.01      140.5     143.5     140.9      0.0       0.0
         3       0.65      143.8     307.8     140.9      0.0       0.0
         4       1.30      146.7     472.6     140.4      0.0       0.0
         5       1.95      149.1     633.6     139.2      0.0       0.0


         6       2.60      151.2     790.8     137.3      0.0       0.0
         7       3.25      155.0     946.5     135.1      0.0       0.0
         8       3.90      161.9    1110.4     132.4      0.0       0.0
         9       4.55      174.5    1297.5     129.0      0.0       0.0
        10       5.20      192.8    1500.1     124.8      0.0       0.0


        11       5.80      197.6    1699.9     119.2      0.0       0.0
        12       6.88      181.8    2017.2     105.5      0.0       0.0
        13       7.96      147.1    2274.6      87.3      0.0       0.0
        14       9.04      146.5    2462.3      70.0      0.0       0.0
        15       9.05      146.7     142.7       0.0      0.0       0.0


        16      10.12      198.3     327.3       0.0      0.0       0.0
        17      11.20      316.4     500.0       0.0      0.0       0.0
        18      12.27      237.8     661.5       0.0      0.0       0.0
        19      12.28      237.8     696.0       0.0      0.0       0.0
```

```
--ROCKET HEAT TRANSFER MODEL--   AEROJET REGENERATIVELY COOLED O2/H2 ENGINE

     ITERATION NO.   CORE O/F   CORE TEMP    WT. FLOW
                                 DEG F        LB/SEC
                60     4.000     5185.3       3.057

      NODE     STATION    T WALL     T FILM    T BULK  T LINER     T H2
                                               REGEN              LINER


         1       0.0      141.1      141.1     141.2      0.0       0.0
         2       0.01     141.1      143.8     141.2      0.0       0.0
         3       0.65     144.1      308.0     141.2      0.0       0.0
         4       1.30     146.9      472.8     140.6      0.0       0.0
         5       1.95     149.3      633.8     139.4      0.0       0.0


         6       2.60     151.4      790.9     137.5      0.0       0.0
         7       3.25     155.2      946.6     135.3      0.0       0.0
         8       3.90     162.1     1110.5     132.6      0.0       0.0
         9       4.55     174.7     1297.5     129.3      0.0       0.0
        10       5.20     193.0     1500.0     125.0      0.0       0.0


        11       5.80     197.8     1699.8     119.4      0.0       0.0
        12       6.88     182.0     2017.1     105.7      0.0       0.0
        13       7.96     148.1     2274.4      87.5      0.0       0.0
        14       9.04     157.8     2462.1      70.0      0.0       0.0
        15       9.05     158.0      143.0       0.0      0.0       0.0


        16      10.12     207.7      327.6       0.0      0.0       0.0
        17      11.20     327.4      500.2       0.0      0.0       0.0
        18      12.27     262.3      661.6       0.0      0.0       0.0
        19      12.28     262.3      696.1       0.0      0.0       0.0
```

APPENDIX C

PROGRAM SUBROUTINES

CHARACTERIZATION PROGRAM SUBROUTINES



SUBROUTINE DESCRIPTIONS

SUBROUTINE DESCRIPTIONS

MAIN

The MAIN routine accepts input for all models except heat transfer. Unit conversions are made in MAIN to prepare input data for the combustion subprogram which makes calculations in metric units. The MAIN routine also contains the logic for calling the principal models. It controls the dynamics model shutdown, time advance, etc.

SUBROUTINE AREA

When the thrust chamber wall radius, $r_w(x)$, is specified (MB = 2), dP/dx is evaluated in this subroutine by means of an iterative process involving the strict convergence of $r'_w = r'_w$ (dP/dx, physical chamber conditions at x) to $r_w/(x)$.

SUBROUTINE BEGIN

Molecular weights, heats of formation, and polynomial curve fits for the enthalpy and entropy of the chemical species are stored here. In addition, the units of the thermodynamic property information are converted from the English and c.g.s. systems to the S. I. system.

SUBROUTINE BLOW

This subroutine is used to calculate chamber pressure after the valves are closed. Its purpose is to reduce the computer time required to calculate blowdown conditions.

SUBROUTINE BUMP

BUMP is used to reduce the dynamics calculating increment during the first few iterations when the stable time is relatively large. The purpose is to increase the resolution of the model during the first iterations.

SUBROUTINE BULK

This subroutine integrates the dependent variable (u, H, and α^j or $\tilde{\alpha}^k$) radially across the chamber, at each axial station, to determine their bulk values, and then calls the appropriate subroutines to determine bulk values for the dependent variables (h, T, C_p, ρ, w, etc.). These bulk quantities are used for printout purposes, and where needed, in applying the chamber wall boundary conditions.

SUBROUTINE CFLOW

CFLOW is used to calculate mass flow for a choked duct given the 4fL/D parameter.

SUBROUTINE COLD

Subroutine COLD calculates performance of engine which propellants are not ignited.

SUBROUTINE COMB

COMB is the calling program for the combustion module. Once called by MAIN, COMB controls combustion calculations much as MAIN controls the overall characterization run.

SUBROUTINE CONSRV

An explicit solution of the conservation equations for velocity, total enthalpy, and either the chemical specie mass fraction (frozen (ICHEM = 1) or finite-rate chemistry (ICHEM = 3), or the chemical element mass fraction (equilibrium (ICHEM = 2) chemistry) diffusion equations is performed in this subroutine.

SUBROUTINE CPRF

This routine calculates the choking pressure ratio for a duct for a given 4fL/D parameter.

SUBROUTINE CRSPLT

Crossplot data from GRAD against composite kinetic curve of NASA CR 72601 to find Freeze Area Ratio.

SUBROUTINE DENSE

The density and molecular weight at each grid point in the flow field are computed here as a function of p(x), T (x, ψ) and the $\alpha^j (x, \psi)$.

SUBROUTINE DIV , BL, SIDEL

DIV calculates percent I_{sp} loss due to divergence.

SUBROUTINE BL

Subroutine BL calculates boundary layer loss.

SUBROUTINE SIDEL

SIDEL converts percent recombination to specific impulse.

SUBROUTINE DYNAM

DYNAM contains the logic for calculating mass flow rates and volume mixtures, temperatures, and pressures. It contains the logic for sparking, ignition, and quenching tests, and also calculates the stable time and the instantaneous orifice cross-sectional areas. All dynamics output information is printed from DYNAM.

SUBROUTINE EQUILC

When the option of equilibrium chemistry is chosen (ICHEM = 2), the chemical species composition at each flow field grid point is computed here using a "complete combustion" model. The specie mass fractions ($\alpha^J (x, \psi)$) for the species H_2O, H_2 and O_2 are computed from the element mass fractions ($\tilde{\alpha}(x, \psi)$) for H and O.

SUBROUTINE FFCF

Subroutine FFCF is used to compute duct friction coefficient given the flow Reynold's number assuming a smooth duct surface.

SUBROUTINE FG

The coefficients and forcing vector of the linearized chemical kinetic reactions are evaluated in this subroutine.

SUBROUTINE FGX

The coefficients and forcing vector of the linearized chemical kinetic reactions are evaluated in this subroutine.

SUBROUTINE FLOW

FLOW is used to calculate mass flow rate in ducts where choking does not occur. Two parameters are required, the pressure ratio across the duct and the friction parameter, 4fL/D.

SUBROUTINE FLUX

When either the transpiration-cooling model, or film-cooling model, is being used at the wall, this subroutine monitors the amount of mass being added to the flow field, and adds additional grid points to the computation as required.

SUBROUTINE FMOLWT

This routine calculates the molecular weight of the products of GO_2/GH_2 combustion given the mixture ratio.

SUBROUTINE GRAD, NOZ

GRAD determines the area ratio gradient of the nozzle at specified area ratios based on NASA CR 72601. NOZ is used to calculate dA/dX from NASA CR 72601.

SUBROUTINE GRID

This subroutine controls the addition, or subtraction, of streamline grid points from the finite difference flow field computation. Grid points are added above (until the wall is reached) and below (for $\psi(1) > 0$) the present grid according to tests involving the principal flow field variables (H, u, and α^j or $\tilde{\alpha}^k$). In addition, when one less than double the initial number of grid points is in use, alternate grid points are discarded, the interval between grid points doubled, and the computation returns to using the initial number of grid points.

SUBROUTINE HEAT

The thermodynamic properties at each flow field grid point are computed here. There are two options. For KOPT = 1, $C_p (x, \psi)$, $h (x, \psi)$, and $H (x, \psi)$ are determined from $T (x, \psi)$, $u (x, \psi)$ and $\alpha^j (x, \psi)$. For KOPT = 2, $T (x, \psi)$, $C_p (x, \psi)$, and $h (x, \psi)$ are determined for $H (x, \psi)$, $u (x, \psi)$, and $\alpha^j (x, \psi)$.

SUBROUTINE HEATT

HEATT is the main heat transfer routine. It accepts heat transfer input and creates the thermal model. It contains the logic to calculate temperatures using any of the available cooling models. HEATT also prints results and input parameters in a concise format.

SUBROUTINE HONC

The finite rate chemistry calculation is controlled in this subroutine. It regulates the chemistry time steps, calls the subroutines which (a) compute the reaction rates, (b) evaluates the linearized equations using a Pade' rational approximation for the exponentials representing the solution of the coupled first order linear ordinary differential equations, and (c) solve the matrix which represents the integration of the chemical kinetics equations over the particular time step.

SUBROUTINE INJECT

Subroutine INJECT is used to compute starting profiles for the combustion model. It calculates velocity, temperature, and species mass fractions at each combustion grid point. It zeros all species mass fractions other than those of O_2 and H_2.

SUBROUTINE INVERT

This subroutine serves the same purpose as HEAT, but for a single grid point at a time rather than the entire grid. "Entry HOOT" is part of this subroutine.

SUBROUTINE MARCH

This subroutine performs the calling function for the combustion calculations. It calls the subprograms to make sequential calculations in the axial direction.

SUBROUTINE PADE

This subroutine controls the matrix solution of the integrated linearized chemical kinetics equations and prepares the inputs for the actual matrix calculation.

SUBROUTINE PERF

This subprogram is the main driver for computing the performance of the rocket engine using data transferred from the combustion and dynamic programs. I_{sp}, Thrust, and C_F are calculated.

SUBROUTINE PERSHF

This routine determines percent recombination as a function of Freeze Area Ratio from CRSPLT.

SUBROUTINE PREPAR

When the finite rate chemistry option (ICHEM = 3) is employed, this subroutine serves as a connecting link between the flow field computation and the chemical kinetics model. Here inputs are prepared for the chemistry subroutines, the chemistry time steps related to the flow field step size, and the results of the chemistry computation prepared for insertion into the flow field computation.

SUBROUTINE PRESS

This subroutine is used to evaluate an arbitrary function and its first derivative, f(x) and df (x)/dx, from input polynomials of as high as fifth order.

SUBROUTINE PICRT

The solution of the matrix representing the linearized chemical kinetic equations is performed here.

SUBROUTINE PRINT

This subroutine is used to printout the radial profiles of the principal flow variables, such as H, u, t, α^j, h, etc. In addition, dependent variables which are not required for the flow field computation, but are of interest, such as C_p, γ, φ, H_s, etc., are computed and printed. Also, average properties across the flow field are computed and printed.

SUBROUTINE RTNI

RTNI is a routine which used Newton's method to solve nonlinear equations. It is used by INJECT to solve for flow Mach number.

SUBROUTINE SI

Subroutine SI calculates equilibrium and frozen specific impulse for each streamline.

SUBROUTINE SLOT

When slot cooling is used at the chamber wall (ISOBAT = 4), this subroutine is used to apply the wall boundary conditions to the conservation equations for energy and diffusion.

SUBROUTINE SPHEAT

SPHEAT calculates the ratio of specific heats of the products of GO_2/GH_2 combustion given the mixture ratio.

SUBROUTINE STEP

The step-size, Δx, for the next flow step is computed here. The step-size is defined by applying the von Neuman stability criterion to the conservation equations. This results in the criterion that

$$\Delta x = \text{minimum} \left[\left(\frac{(\Delta\psi)^2}{4\mu Sc} \right) \qquad \left(\frac{\psi_{m,n} (\Delta\psi)^2}{\left(\frac{b}{Sc}\right)_{n,m+\frac{1}{2}} + \left(\frac{b}{Sc}\right)_{n,m-\frac{1}{2}}} \right) (r_{n,m+1} - r_{n,m}) \right]$$

SUBROUTINE TGR

TGR is used to calculate equilibrium combustion gas temperature for given mixture ratio and pressure. It is utilized in the dynamics model and the heat transfer model.

SUBROUTINE TRANSP

When transpiration cooling is used at the chamber wall (ISOBAT = 3), this subroutine is employed to apply the wall boundary conditions to the conservation of energy and diffusion equations.

SUBROUTINE VASC

This subroutine calculates a laminar viscosity, using the Sutherland model for air, for use in applying the wall boundary conditions.

SUBROUTINE VISC

The turbulent viscosity coefficient, $\mu(x)$, is computed here using a model employing Hersch's mixing parameter:

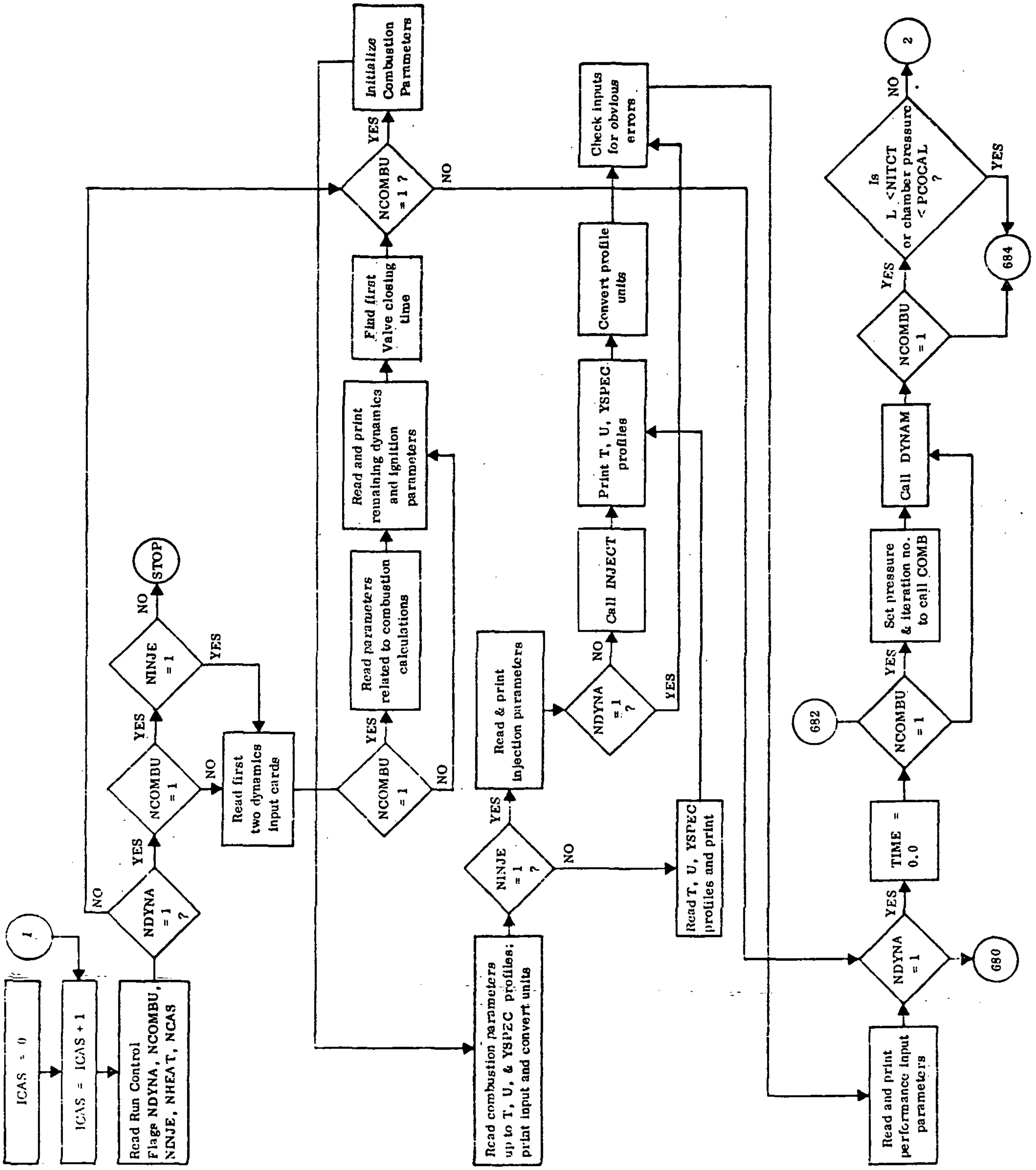
$$\mu = \frac{\alpha k D \rho u}{1+x/S}$$

SUBROUTINE WALL

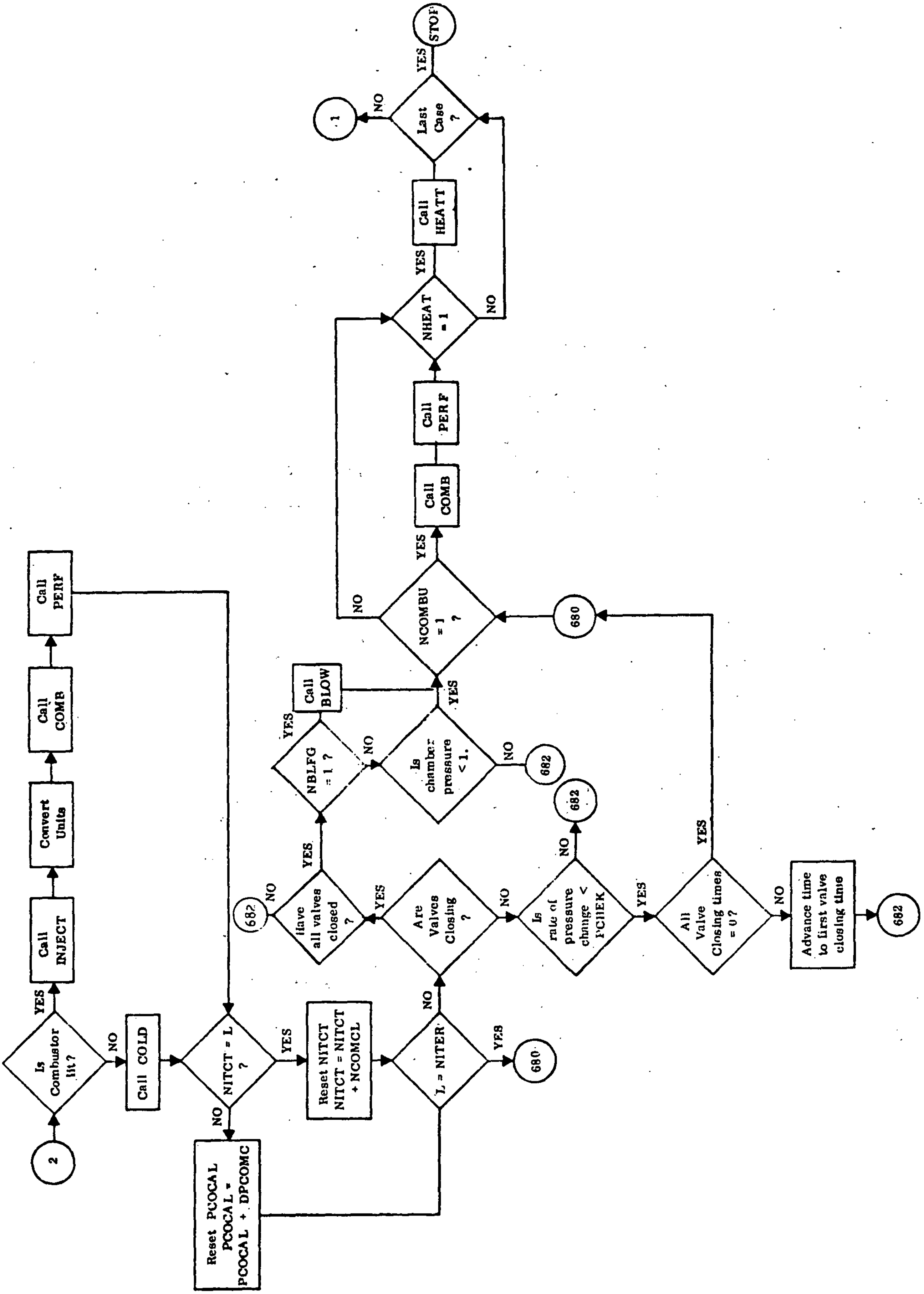
This subroutine controls the application of the wall boundary conditions to the flow field computation. After first testing as to whether the wall lies above, or is in the flow field at a given x station, it proceeds to apply the boundary conditions to the momentum conservation equation, and then calls the appropriate subroutine to apply the B. C. to the energy and diffusion conservation equations.

SUBROUTINE FLOW CHARTS

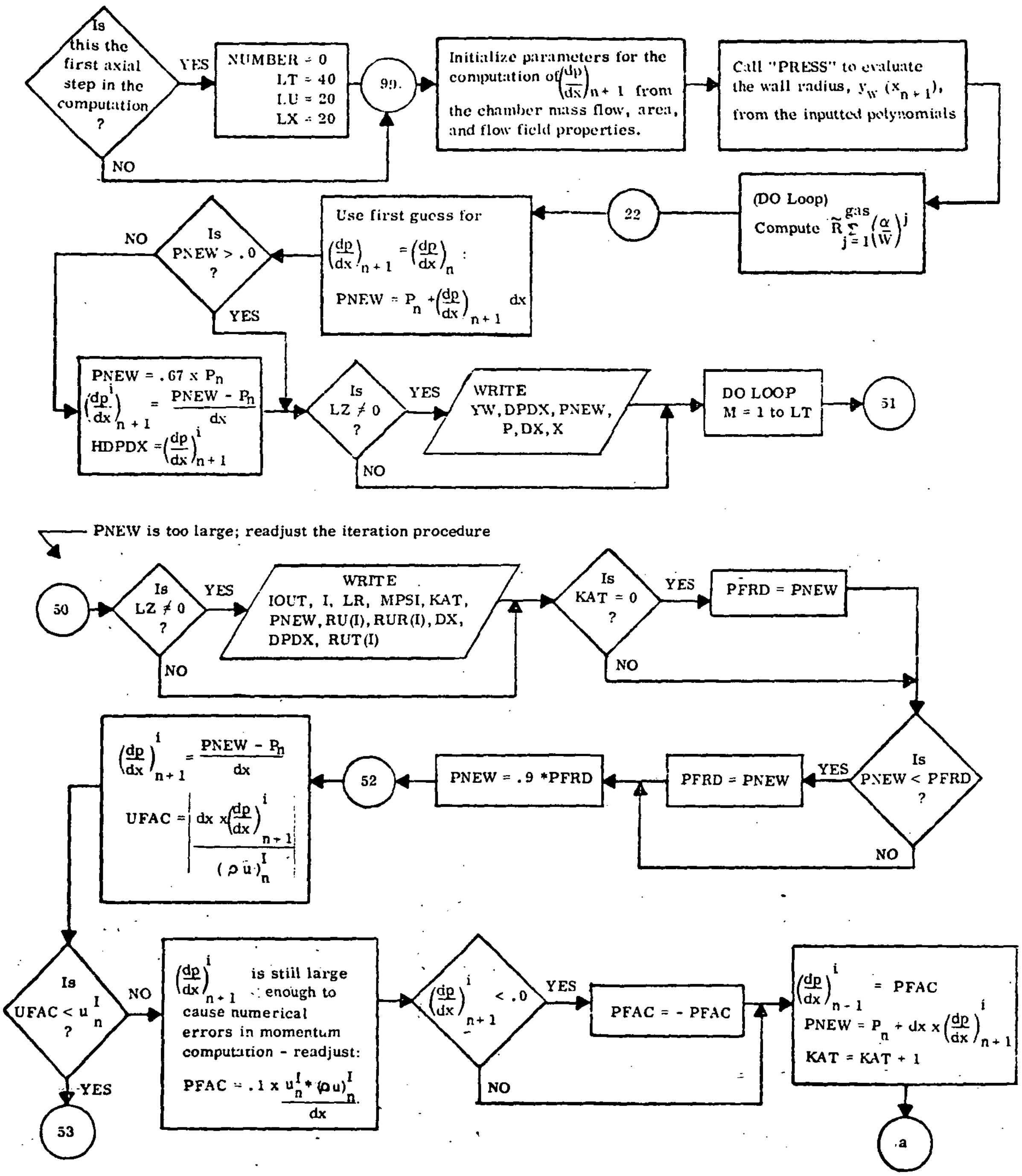
MAIN


ICAS = 0
1
ICAS = ICAS + 1
Read Run Control Flags NDYNA, NCOMBU, NINJE, NHEAT, NCAS
NDYNA = 1 ?
NO
YES
NCOMBU = 1
YES
NO
NINJE = 1
NO
STOP
YES
Read first two dynamics input cards
NCOMBU = 1
YES
NO
Read parameters related to combustion calculations
Read and print remaining dynamics and ignition parameters
Find first Valve closing time
NCOMBU = 1 ?
YES
NO
Initialize Combustion Parameters
Read combustion parameters up to T, U, & YSPEC profiles; print input and convert units
NINJE = 1 ?
YES
NO
Read & print injection parameters
NDYNA = 1 ?
NO
YES
Call INJECT
Print T, U, YSPEC profiles
Convert profile units
Check inputs for obvious errors
Read T, U, YSPEC profiles and print
Read and print performance input parameters
NDYNA = 1
680
YES
TIME = 0.0
682
NCOMBU = 1
YES
Set pressure & iteration no. to call COMB
Call DYNAM
NCOMBU = 1
YES
Is L <NITCT or chamber pressure < PCOCAL ?
NO
2
YES
684

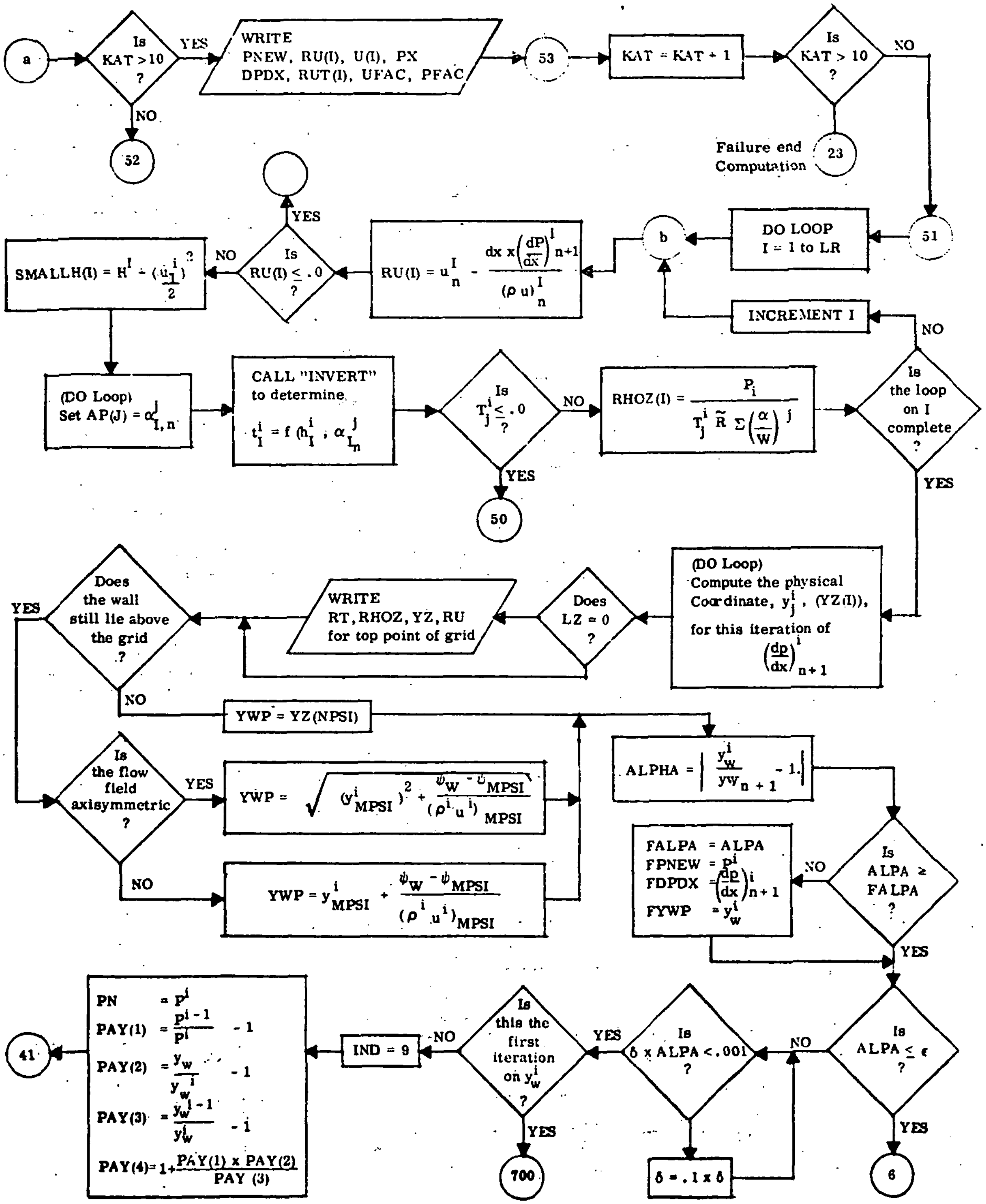
MAIN (Cont.)


2
Is Combustor lit?
YES
NO
Call INJECT
Convert Units
Call COMB
Call PERF
Call COLD
NITCT = L ?
YES
NO
Reset PCOCAL PCOCAL = PCOCAL + DPCOMC
Reset NITCT NITCT = NITCT + NCOMCL
L = NITER
YES
NO
680
Are Valves Closing ?
YES
NO
Have all valves closed ?
YES
NO
682
NBLFG = 1 ?
YES
NO
Call BLOW
Is chamber pressure < 1.
YES
NO
682
Is rate of pressure change < PCHEK
YES
NO
682
All Valve Closing times = 0?
YES
NO
Advance time to first valve closing time
682
680
NCOMBU = 1 ?
YES
NO
Call COMB
Call PERF
NHEAT = 1
YES
NO
Call HEATT
Last Case ?
YES
NO
1
STOP

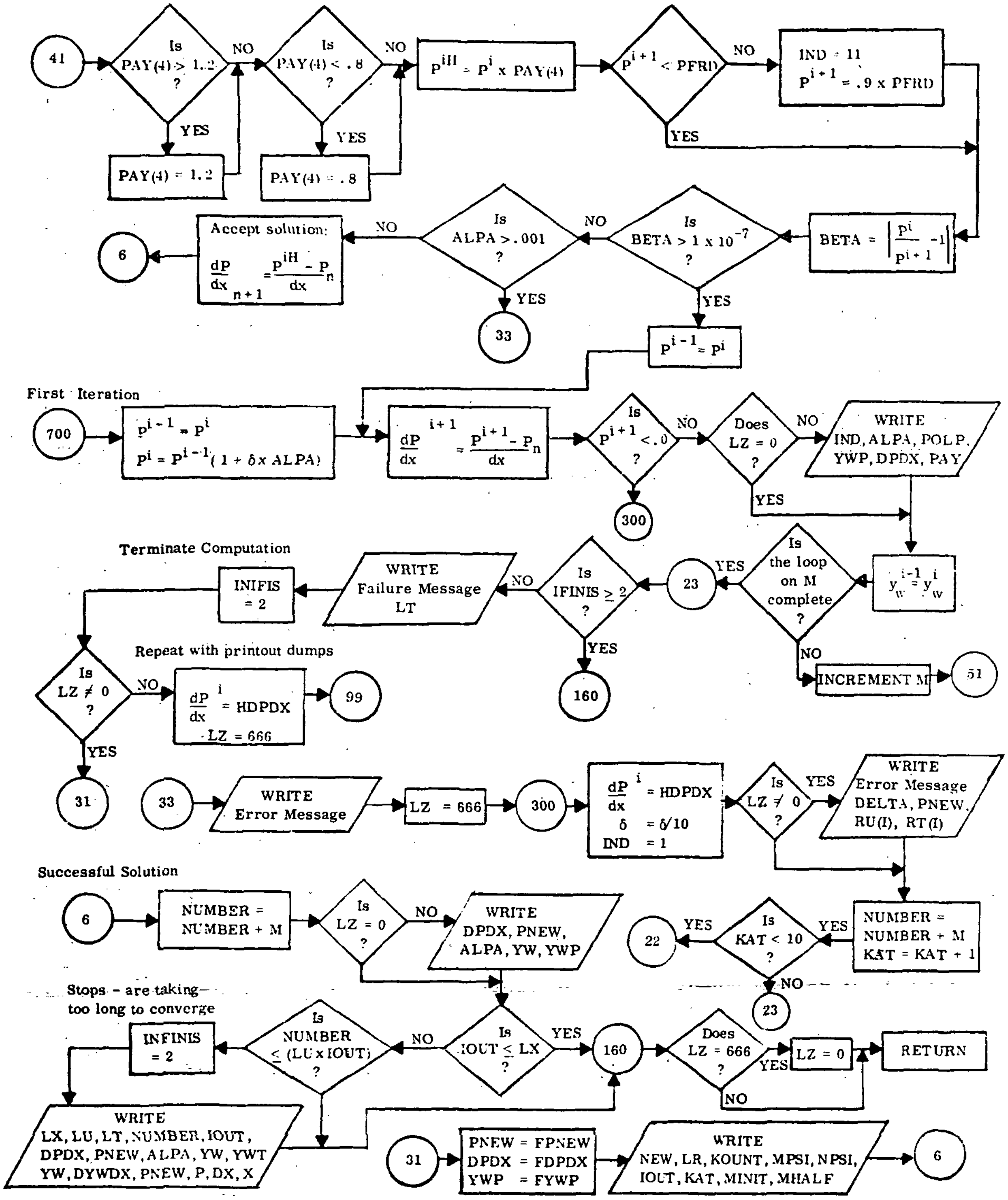
SUBROUTINE AREA


Is this the first axial step in the computation ?
YES
NO
NUMBER = 0
LT = 40
LU = 20
LX = 20
99
Initialize parameters for the computation of $\left(\frac{dp}{dx}\right)_{n+1}$ from the chamber mass flow, area, and flow field properties.
Call "PRESS" to evaluate the wall radius, $y_w(x_{n+1})$, from the inputted polynomials
(DO Loop) Compute $\tilde{R}\sum_{j=1}^{gas}\left(\frac{\alpha}{W}\right)^j$
22
Use first guess for $\left(\frac{dp}{dx}\right)_{n+1} = \left(\frac{dp}{dx}\right)_n$:
$PNEW = P_n + \left(\frac{dp}{dx}\right)_{n+1} dx$
Is PNEW > .0 ?
NO
YES
$PNEW = .67 \times P_n$
$\left(\frac{dp}{dx}\right)^i_{n+1} = \frac{PNEW - P_n}{dx}$
$HDPDX = \left(\frac{dp}{dx}\right)^i_{n+1}$
Is LZ ≠ 0 ?
YES
NO
WRITE YW, DPDX, PNEW, P, DX, X
DO LOOP M = 1 to LT
51
PNEW is too large; readjust the iteration procedure
50
Is LZ ≠ 0 ?
YES
NO
WRITE IOUT, I, LR, MPSI, KAT, PNEW, RU(I), RUR(I), DX, DPDX, RUT(I)
Is KAT = 0 ?
YES
NO
PFRD = PNEW
Is PNEW < PFRD ?
YES
NO
PFRD = PNEW
PNEW = .9 *PFRD
52
$\left(\frac{dp}{dx}\right)^i_{n+1} = \frac{PNEW - P_n}{dx}$
$UFAC = \left| \frac{dx \times \left(\frac{dp}{dx}\right)^i_{n+1}}{(\rho u)^I_n} \right|$
Is $UFAC < u^I_n$?
NO
YES
53
$\left(\frac{dp}{dx}\right)^i_{n+1}$ is still large enough to cause numerical errors in momentum computation - readjust:
$PFAC = .1 \times u^I_n * \frac{(\rho u)^I_n}{dx}$
$\left(\frac{dp}{dx}\right)^i_{n+1} < .0$
YES
NO
PFAC = - PFAC
$\left(\frac{dp}{dx}\right)^i_{n-1} = PFAC$
$PNEW = P_n + dx \times \left(\frac{dp}{dx}\right)^i_{n+1}$
KAT = KAT + 1
a

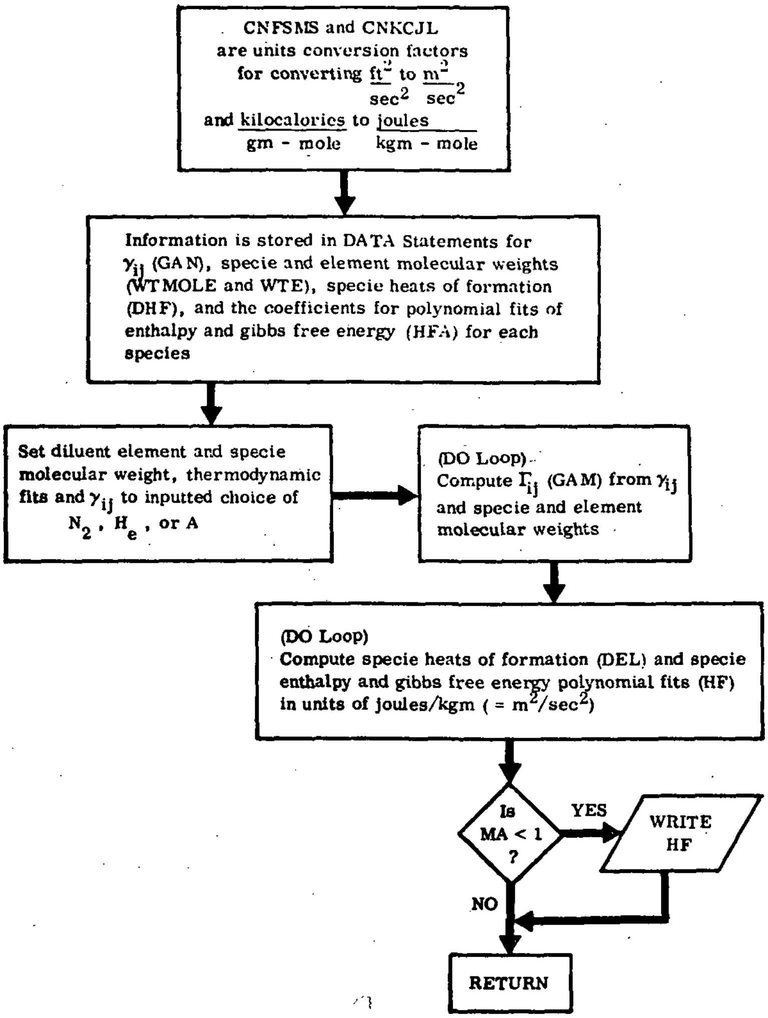
SUBROUTINE AREA (Cont.)


a
Is KAT > 10 ?
YES
NO
52
WRITE PNEW, RU(I), U(I), PX DPDX, RUT(I), UFAC, PFAC
53
KAT = KAT + 1
Is KAT > 10 ?
NO
Failure end Computation
23
51
DO LOOP I = 1 to LR
b
INCREMENT I
NO
$RU(I) = u_n^I - \frac{dx \times \left(\frac{dP}{dx}\right)_{n+1}^I}{(\rho u)_n^I}$
Is RU(I) ≤ . 0 ?
YES
NO
$SMALLH(I) = H^I - \frac{(u_I^i)^2}{2}$
(DO Loop) Set $AP(J) = \alpha_{I,n}^j$
CALL "INVERT" to determine
$t_I^i = f(h_I^i, \alpha_{I_n}^j)$
Is $T_j^i \leq .0$?
NO
YES
50
$RHOZ(I) = \frac{P_i}{T_j^i \tilde{R} \Sigma\left(\frac{\alpha}{W}\right)^j}$
Is the loop on I complete ?
YES
(DO Loop) Compute the physical Coordinate, y_j^i, (YZ(I)), for this iteration of $\left(\frac{dp}{dx}\right)_{n+1}^i$
Does LZ = 0 ?
WRITE RT, RHOZ, YZ, RU for top point of grid
Does the wall still lie above the grid ?
YES
NO
YWP = YZ(NPSI)
Is the flow field axisymmetric ?
YES
NO
$YWP = \sqrt{(y_{MPSI}^i)^2 + \frac{\psi_W - \psi_{MPSI}}{(\rho^i u^i)_{MPSI}}}$
$YWP = y_{MPSI}^i + \frac{\psi_W - \psi_{MPSI}}{(\rho^i u^i)_{MPSI}}$
$ALPHA = \left| \frac{y_W^i}{yw_{n+1}} - 1. \right|$
Is ALPA ≥ FALPA ?
NO
YES
FALPA = ALPA
FPNEW = P^i
FDPDX = $\left(\frac{dp}{dx}\right)_{n+1}^i$
FYWP = y_w^i
Is ALPA ≤ ε ?
YES
NO
6
Is δ x ALPA < .001 ?
YES
δ = .1 x δ
Is this the first iteration on y_w^i ?
YES
NO
700
IND = 9
PN = P^i
PAY(1) = $\frac{P^{i-1}}{P^i} - 1$
PAY(2) = $\frac{y_w}{y_w^i} - 1$
PAY(3) = $\frac{y_w^{i-1}}{y_w^i} - 1$
PAY(4) = $1 + \frac{PAY(1) \times PAY(2)}{PAY(3)}$
41

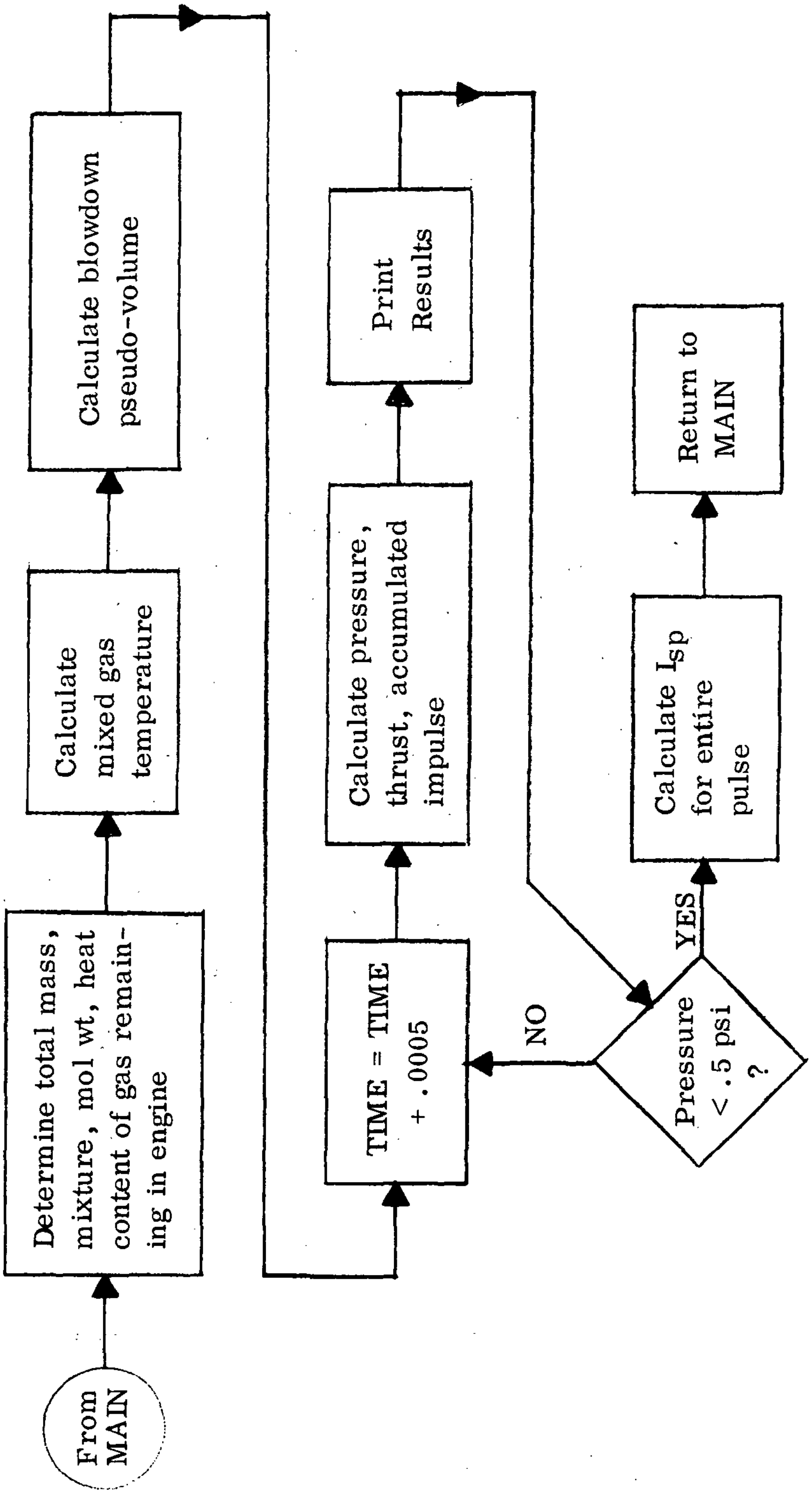
SUBROUTINE AREA (Cont.)



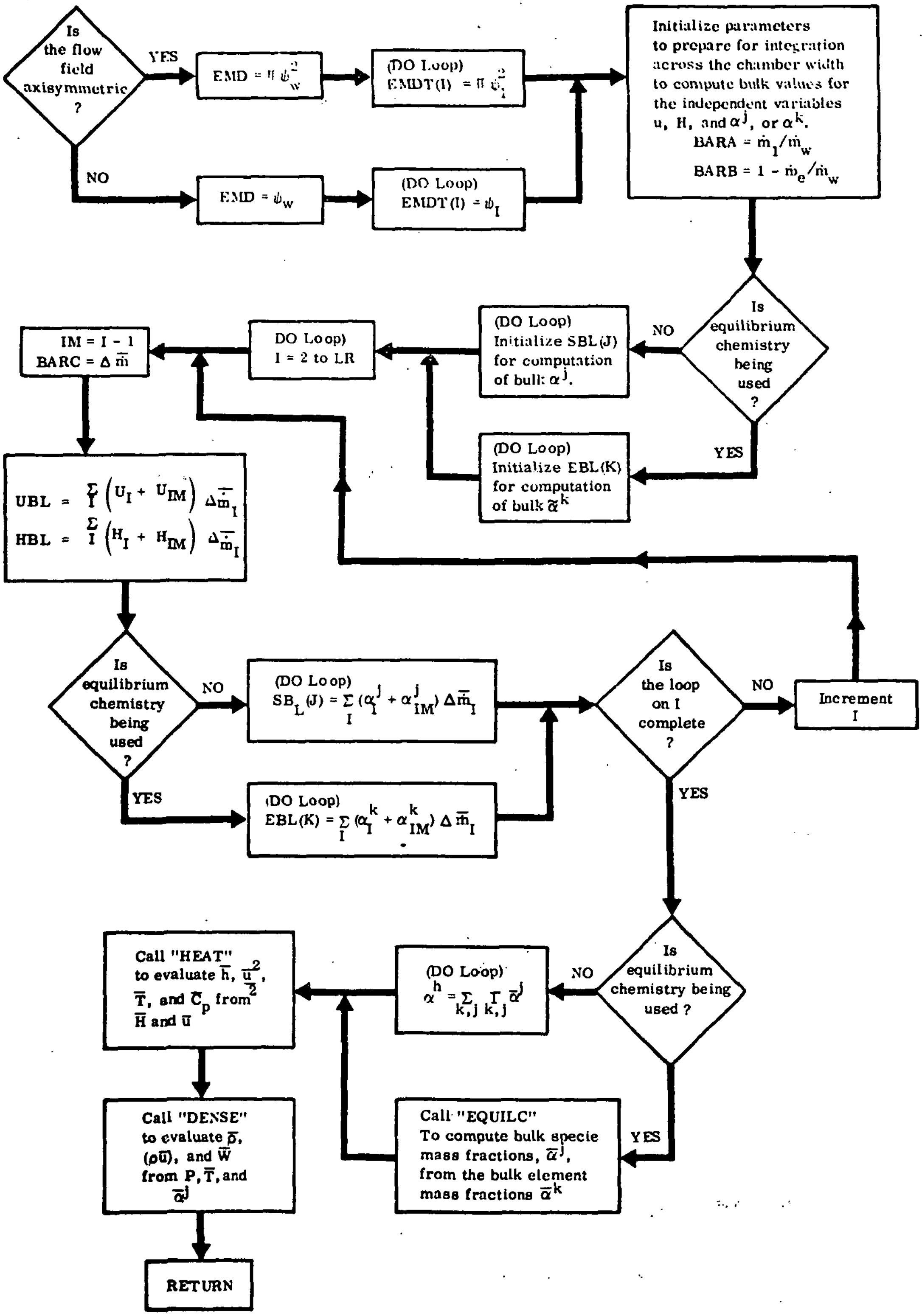
SUBROUTINE BEGIN


CNFSMS and CNKCJL are units conversion factors for converting $\frac{ft^2}{sec^2}$ to $\frac{m^2}{sec^2}$ and $\frac{kilocalories}{gm - mole}$ to $\frac{joules}{kgm - mole}$
Information is stored in DATA Statements for γ_{ij} (GAN), specie and element molecular weights (WTMOLE and WTE), specie heats of formation (DHF), and the coefficients for polynomial fits of enthalpy and gibbs free energy (HFA) for each species
Set diluent element and specie molecular weight, thermodynamic fits and γ_{ij} to inputted choice of N_2, H_e, or A
(DO Loop) Compute Γ_{ij} (GAM) from γ_{ij} and specie and element molecular weights
(DO Loop) Compute specie heats of formation (DEL) and specie enthalpy and gibbs free energy polynomial fits (HF) in units of joules/kgm (= m^2/sec^2)
Is MA < 1 ?
YES
WRITE HF
NO
RETURN

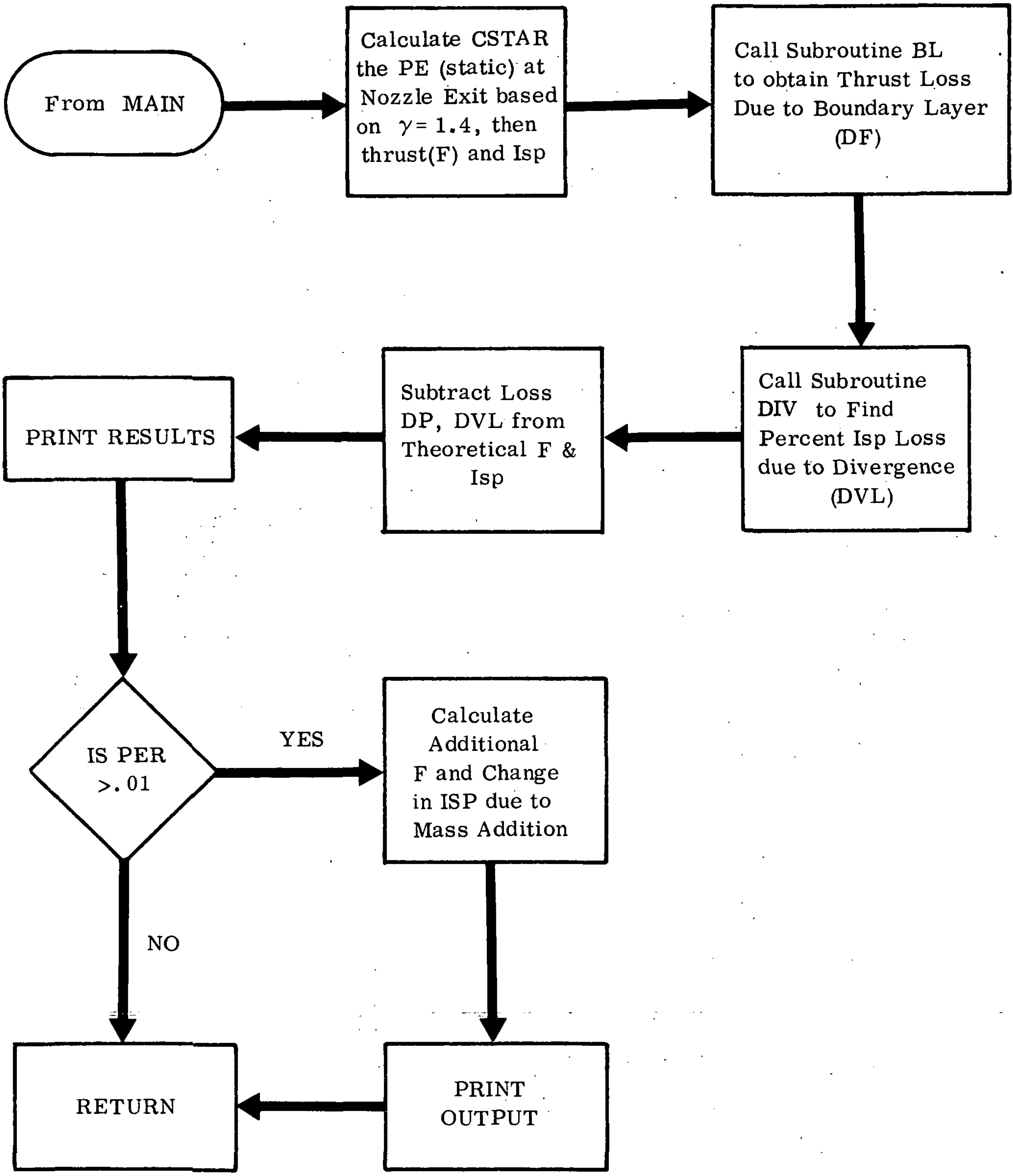
SUBROUTINE BLOW


From MAIN
Determine total mass, mixture, mol wt, heat content of gas remaining in engine
Calculate mixed gas temperature
Calculate blowdown pseudo-volume
TIME = TIME + .0005
Calculate pressure, thrust, accumulated impulse
Print Results
Pressure < .5 psi ?
NO
YES
Calculate I_{sp} for entire pulse
Return to MAIN

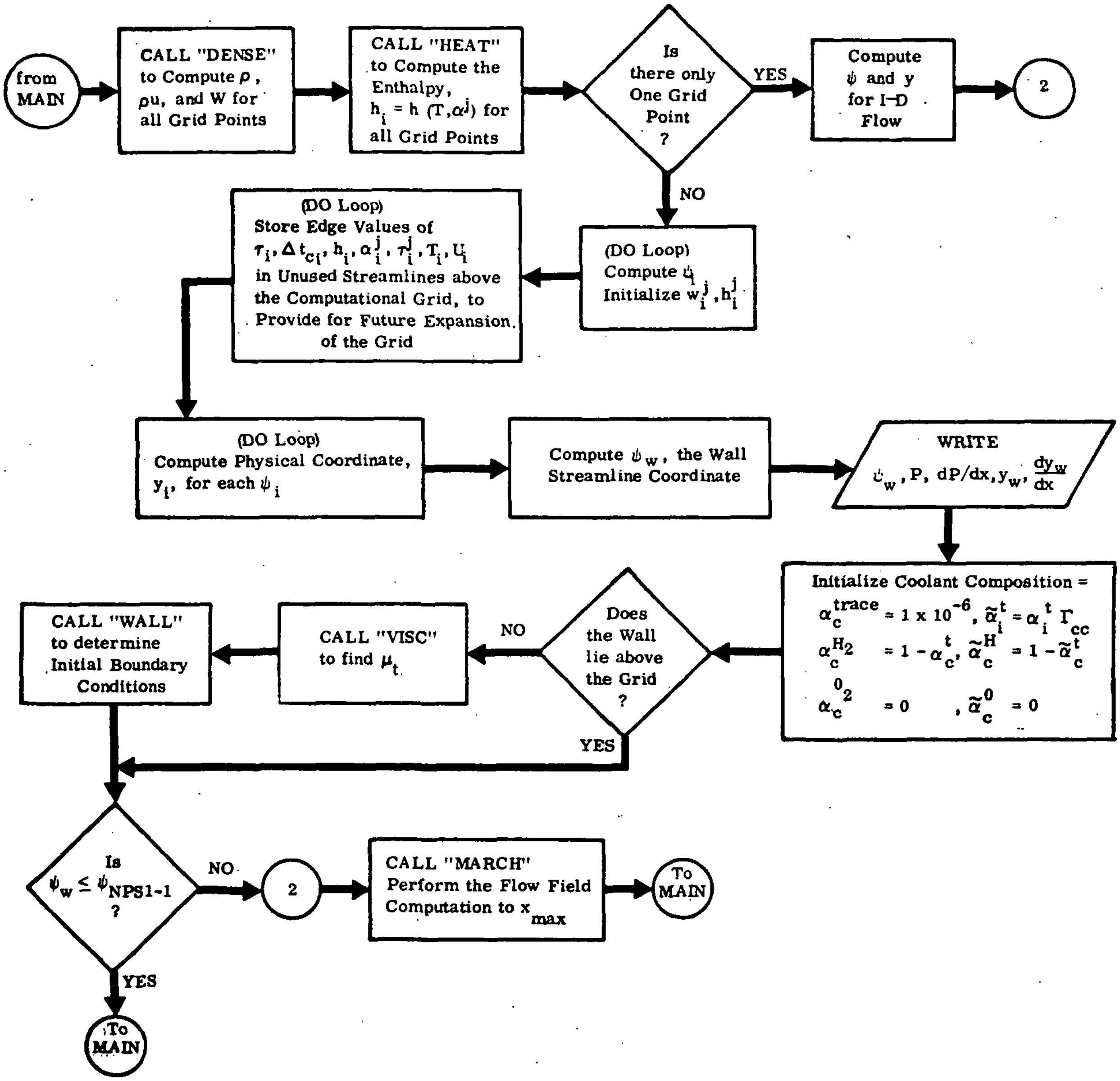
SUBROUTINE BULK


Is the flow field axisymmetric ?
YES
NO
EMD = $\pi \psi_w^2$
(DO Loop) EMDT(I) = $\pi \psi_I^2$
EMD = ψ_w
(DO Loop) EMDT(I) = ψ_I
Initialize parameters to prepare for integration across the chamber width to compute bulk values for the independent variables u, H, and α^j, or α^k.
BARA = $\dot{m}_I/\dot{m}_w$
BARB = 1 - $\dot{m}_e/\dot{m}_w$
Is equilibrium chemistry being used ?
NO
YES
(DO Loop) Initialize SBL(J) for computation of bulk α^j.
(DO Loop) Initialize EBL(K) for computation of bulk $\tilde{\alpha}^k$
DO Loop) I = 2 to LR
IM = I - 1
BARC = $\Delta \bar{m}$
UBL = $\sum_I \left(U_I + U_{IM}\right) \Delta\bar{m}_I$
HBL = $\sum_I \left(H_I + H_{IM}\right) \Delta\bar{m}_I$
Is equilibrium chemistry being used ?
NO
YES
(DO Loop) $SB_L(J) = \sum_I (\alpha_I^j + \alpha_{IM}^j) \Delta\bar{m}_I$
(DO Loop) $EBL(K) = \sum_I (\alpha_I^k + \alpha_{IM}^k) \Delta\bar{m}_I$
Is the loop on I complete ?
NO
Increment I
YES
Is equilibrium chemistry being used ?
NO
YES
(DO Loop) $\alpha^h = \sum_{k,j} \Gamma_{k,j} \tilde{\alpha}^j$
Call "EQUILC" To compute bulk specie mass fractions, $\bar{\alpha}^j$, from the bulk element mass fractions $\bar{\alpha}^k$
Call "HEAT" to evaluate $\bar{h}$, $\bar{u}^2$, $\bar{T}$, and $\bar{C}_p$ from $\bar{H}$ and $\bar{u}^2$
Call "DENSE" to evaluate $\bar{\rho}$, $(\rho\bar{u})$, and $\bar{W}$ from P, $\bar{T}$, and $\bar{\alpha}^j$
RETURN

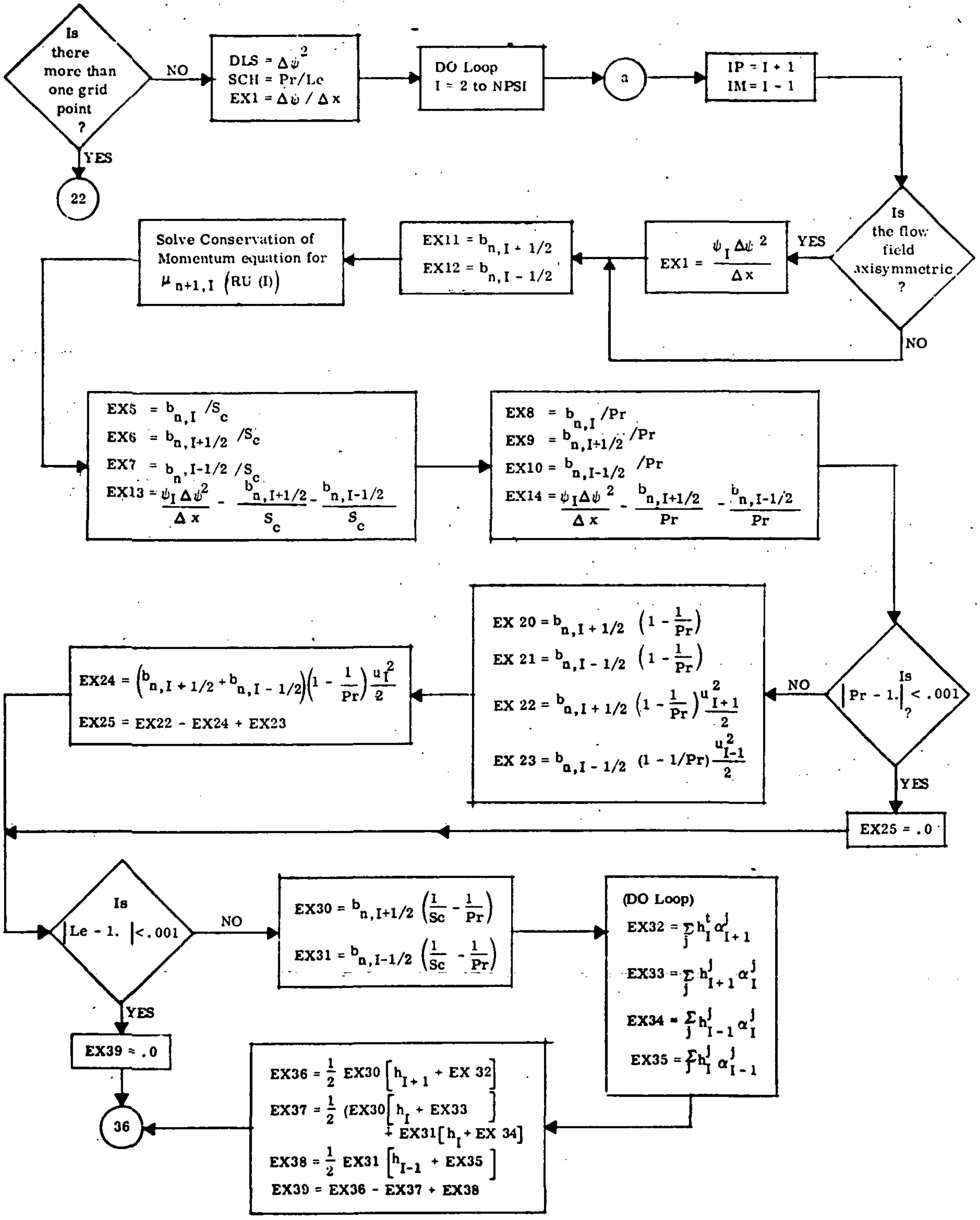
SUBROUTINE COLD


From MAIN
Calculate CSTAR the PE (static) at Nozzle Exit based on $\gamma = 1.4$, then thrust(F) and Isp
Call Subroutine BL to obtain Thrust Loss Due to Boundary Layer (DF)
Call Subroutine DIV to Find Percent Isp Loss due to Divergence (DVL)
Subtract Loss DP, DVL from Theoretical F & Isp
PRINT RESULTS
IS PER >.01
YES
Calculate Additional F and Change in ISP due to Mass Addition
NO
RETURN
PRINT OUTPUT

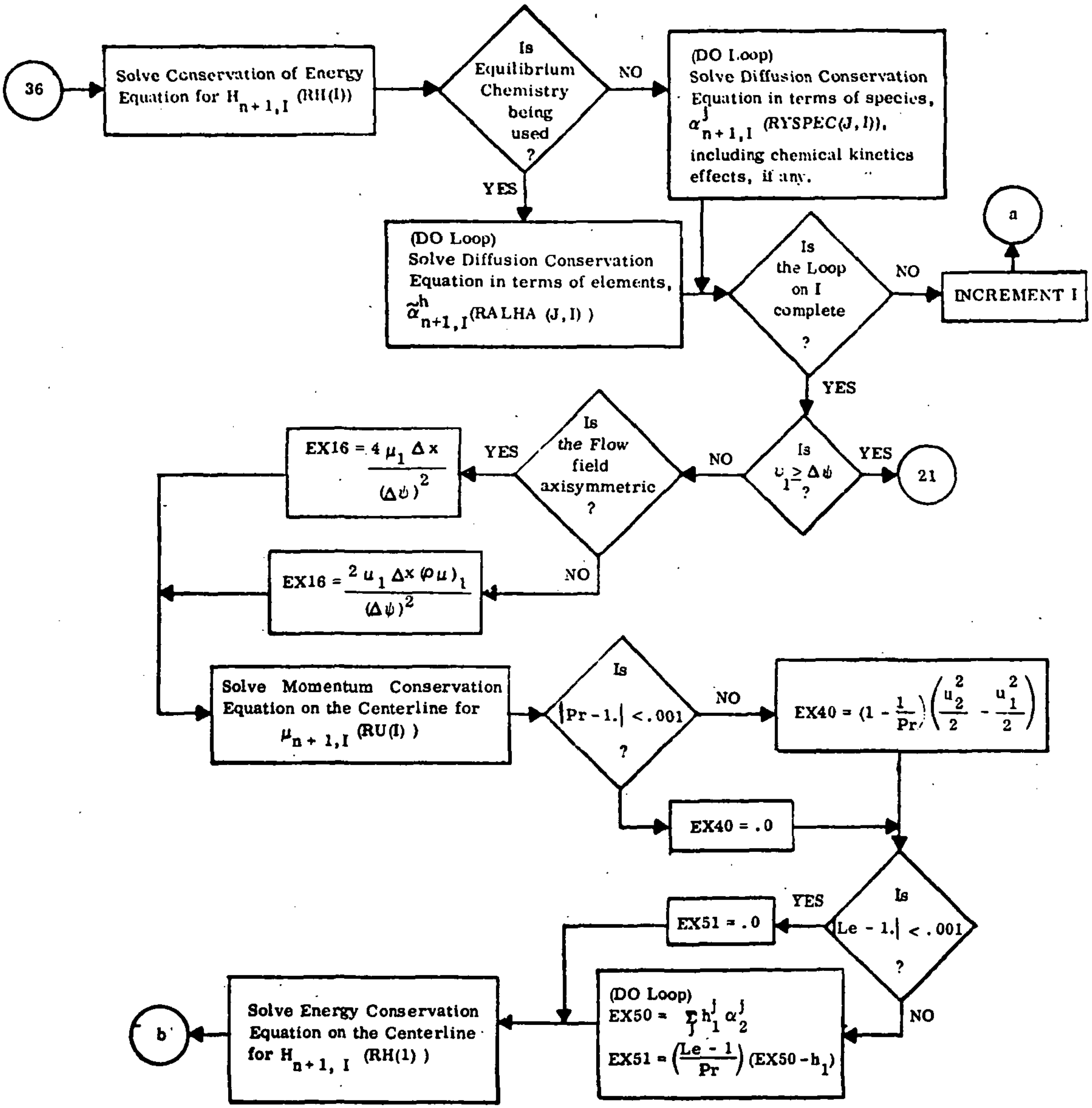
SUBROUTINE COMB


from MAIN
CALL "DENSE" to Compute ρ, ρu, and W for all Grid Points
CALL "HEAT" to Compute the Enthalpy, $h_i = h(T,\alpha^j)$ for all Grid Points
Is there only One Grid Point ?
YES
Compute ψ and y for I-D Flow
2
NO
(DO Loop) Compute ψ_i Initialize w_i^j, h_i^j
(DO Loop) Store Edge Values of $\tau_i, \Delta t_{ci}, h_i, \alpha_i^j, \tau_i^j, T_i, U_i$ in Unused Streamlines above the Computational Grid, to Provide for Future Expansion of the Grid
(DO Loop) Compute Physical Coordinate, y_i, for each ψ_i
Compute ψ_w, the Wall Streamline Coordinate
WRITE ψ_w, P, dP/dx, y_w, $\frac{dy_w}{dx}$
Initialize Coolant Composition = $\alpha_c^{trace} = 1 \times 10^{-6}$, $\tilde{\alpha}_i^t = \alpha_i^t \Gamma_{cc}$ $\alpha_c^{H_2} = 1 - \alpha_c^t$, $\tilde{\alpha}_c^H = 1 - \tilde{\alpha}_c^t$ $\alpha_c^{O_2} = 0$, $\tilde{\alpha}_c^0 = 0$
Does the Wall lie above the Grid ?
NO
YES
CALL "VISC" to find μ_t
CALL "WALL" to determine Initial Boundary Conditions
Is $\psi_w \leq \psi_{NPS1-1}$?
NO
YES
2
CALL "MARCH" Perform the Flow Field Computation to x_{max}
To MAIN
To MAIN

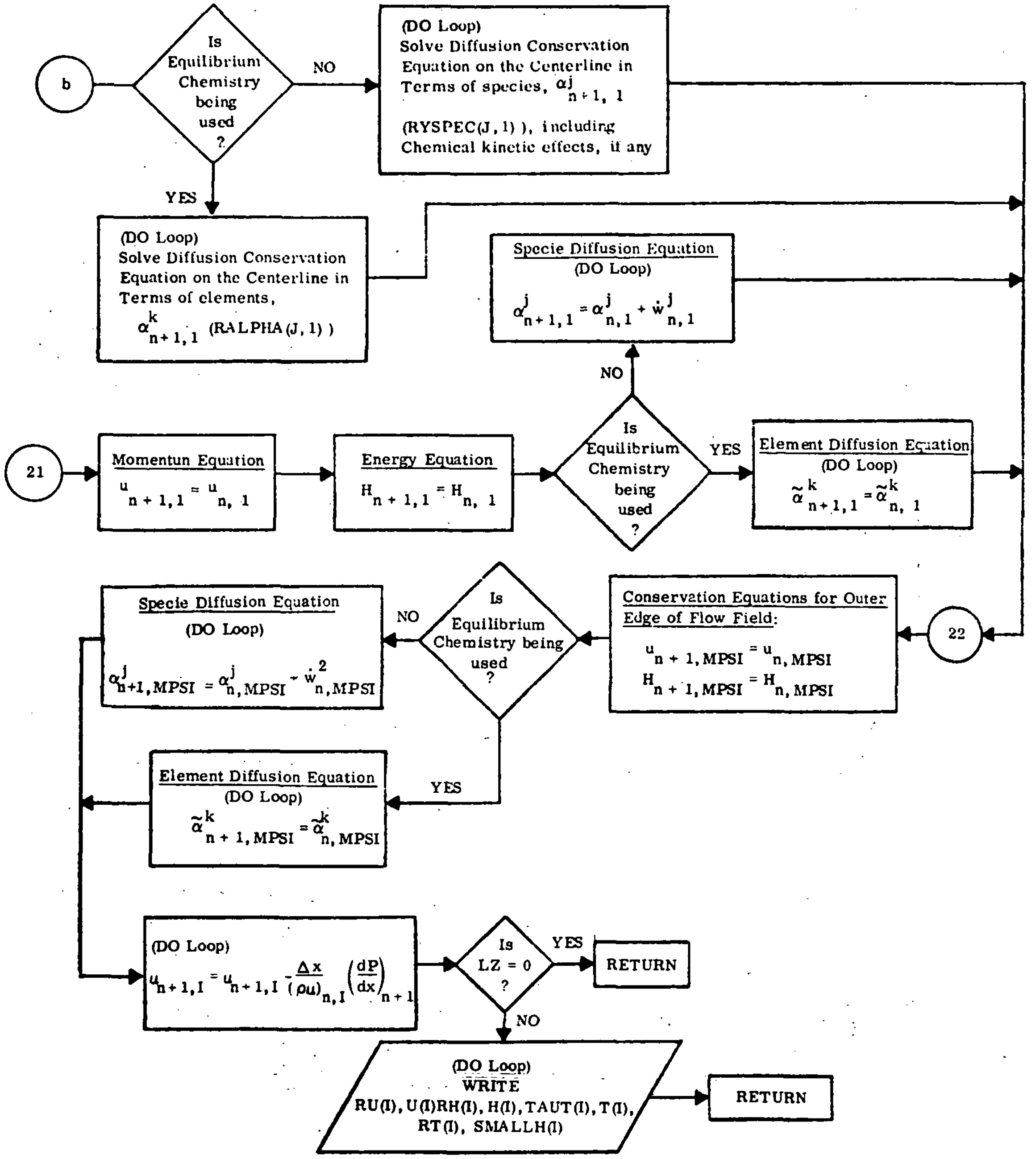
SUBROUTINE CONSRV


Is there more than one grid point ?
NO
YES
22
DLS = $\Delta\psi^2$
SCH = Pr/Le
EX1 = $\Delta\psi$ / Δx
DO Loop
I = 2 to NPSI
a
IP = I + 1
IM = I - 1
Is the flow field axisymmetric ?
YES
NO
EX1 = $\frac{\psi_I \Delta\psi^2}{\Delta x}$
EX11 = $b_{n,I+1/2}$
EX12 = $b_{n,I-1/2}$
Solve Conservation of Momentum equation for $\mu_{n+1,I}$ (RU (I))
EX5 = $b_{n,I}/S_c$
EX6 = $b_{n,I+1/2}/S_c$
EX7 = $b_{n,I-1/2}/S_c$
EX13 = $\frac{\psi_I\Delta\psi^2}{\Delta x} - \frac{b_{n,I+1/2}}{S_c} - \frac{b_{n,I-1/2}}{S_c}$
EX8 = $b_{n,I}$/Pr
EX9 = $b_{n,I+1/2}$/Pr
EX10 = $b_{n,I-1/2}$/Pr
EX14 = $\frac{\psi_I\Delta\psi^2}{\Delta x} - \frac{b_{n,I+1/2}}{Pr} - \frac{b_{n,I-1/2}}{Pr}$
Is $|Pr - 1.| < .001$?
NO
YES
EX 20 = $b_{n,I+1/2}\left(1 - \frac{1}{Pr}\right)$
EX 21 = $b_{n,I-1/2}\left(1 - \frac{1}{Pr}\right)$
EX 22 = $b_{n,I+1/2}\left(1 - \frac{1}{Pr}\right)\frac{u^2_{I+1}}{2}$
EX 23 = $b_{n,I-1/2}$ (1 - 1/Pr)$\frac{u^2_{I-1}}{2}$
EX24 = $\left(b_{n,I+1/2} + b_{n,I-1/2}\right)\left(1 - \frac{1}{Pr}\right)\frac{u_I^2}{2}$
EX25 = EX22 - EX24 + EX23
EX25 = .0
Is $|Le - 1.| < .001$
NO
YES
EX39 = .0
36
EX30 = $b_{n,I+1/2}\left(\frac{1}{Sc} - \frac{1}{Pr}\right)$
EX31 = $b_{n,I-1/2}\left(\frac{1}{Sc} - \frac{1}{Pr}\right)$
(DO Loop)
EX32 = $\sum_j h_I^t \alpha^j_{I+1}$
EX33 = $\sum_j h^j_{I+1} \alpha^j_I$
EX34 = $\sum_j h^j_{I-1} \alpha^j_I$
EX35 = $\sum_j h^j_I \alpha^j_{I-1}$
EX36 = $\frac{1}{2}$ EX30 $\left[h_{I+1} + EX\ 32\right]$
EX37 = $\frac{1}{2}$ (EX30$\left[h_I + EX33\right]$ + EX31$\left[h_I + EX\ 34\right]$
EX38 = $\frac{1}{2}$ EX31 $\left[h_{I-1} + EX35\right]$
EX39 = EX36 - EX37 + EX38

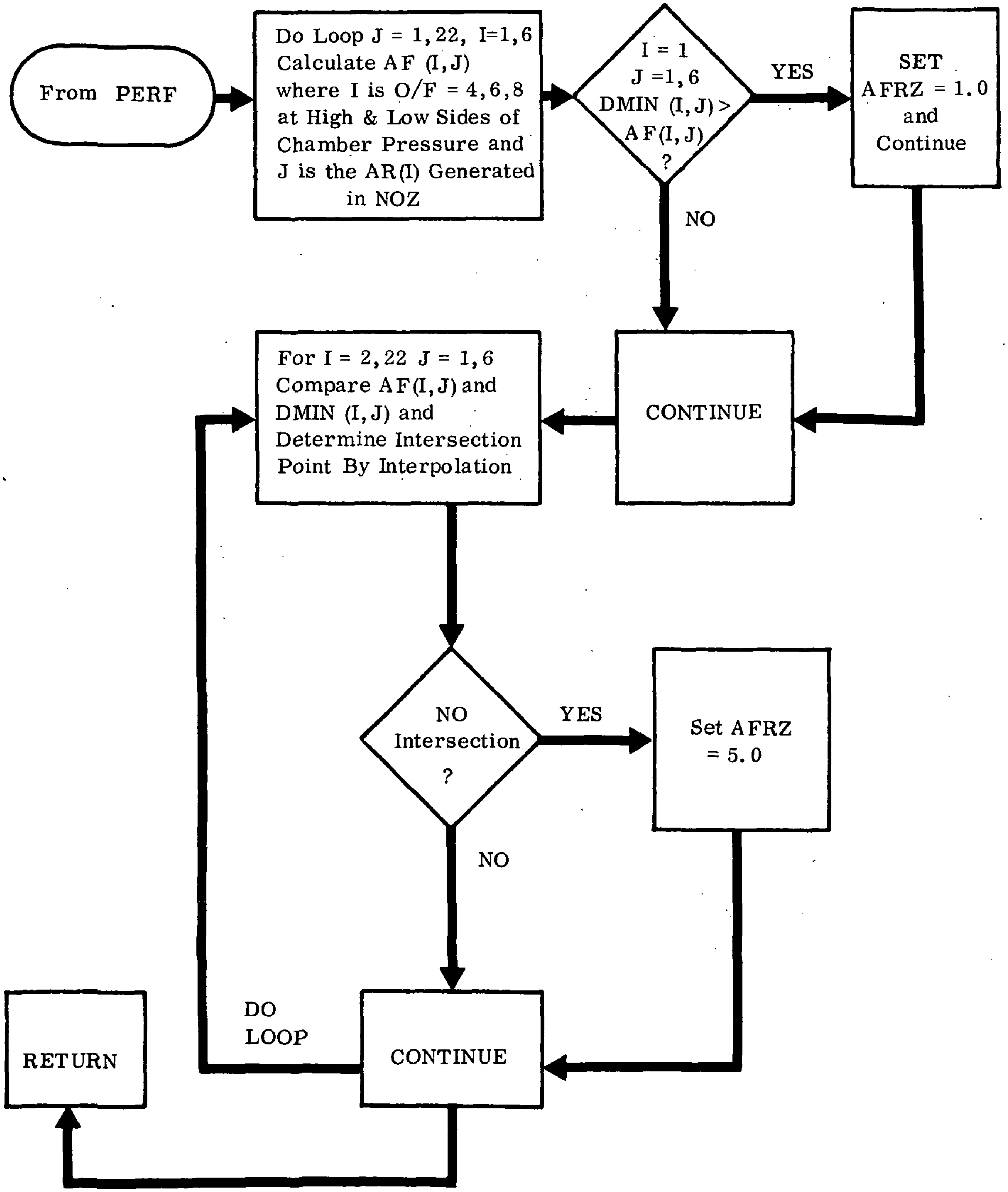
SUBROUTINE CONSRV (Cont.)


36
Solve Conservation of Energy Equation for $H_{n+1,I}$ (RH(I))
Is Equilibrium Chemistry being used ?
NO
(DO Loop) Solve Diffusion Conservation Equation in terms of species, $\alpha^j_{n+1,I}$ (RYSPEC(J, I)), including chemical kinetics effects, if any.
YES
(DO Loop) Solve Diffusion Conservation Equation in terms of elements, $\tilde{\alpha}^h_{n+1,I}$(RALHA (J,I))
Is the Loop on I complete ?
NO
INCREMENT I
a
YES
Is $\upsilon_1 \geq \Delta\psi$?
YES
21
NO
Is the Flow field axisymmetric ?
YES
$EX16 = \frac{4\,\mu_1\,\Delta x}{(\Delta\psi)^2}$
NO
$EX16 = \frac{2\,u_1\,\Delta x\,(\rho u)_1}{(\Delta\psi)^2}$
Solve Momentum Conservation Equation on the Centerline for $\mu_{n+1,I}$ (RU(I))
Is $|Pr - 1.| < .001$?
NO
$EX40 = \left(1 - \frac{1}{Pr}\right)\left(\frac{u_2^2}{2} - \frac{u_1^2}{2}\right)$
EX40 = .0
Is $|Le - 1.| < .001$?
YES
EX51 = .0
NO
(DO Loop)
$EX50 = \sum_j h_1^j\,\alpha_2^j$
$EX51 = \left(\frac{Le - 1}{Pr}\right)(EX50 - h_1)$
Solve Energy Conservation Equation on the Centerline for $H_{n+1,I}$ (RH(1))
b

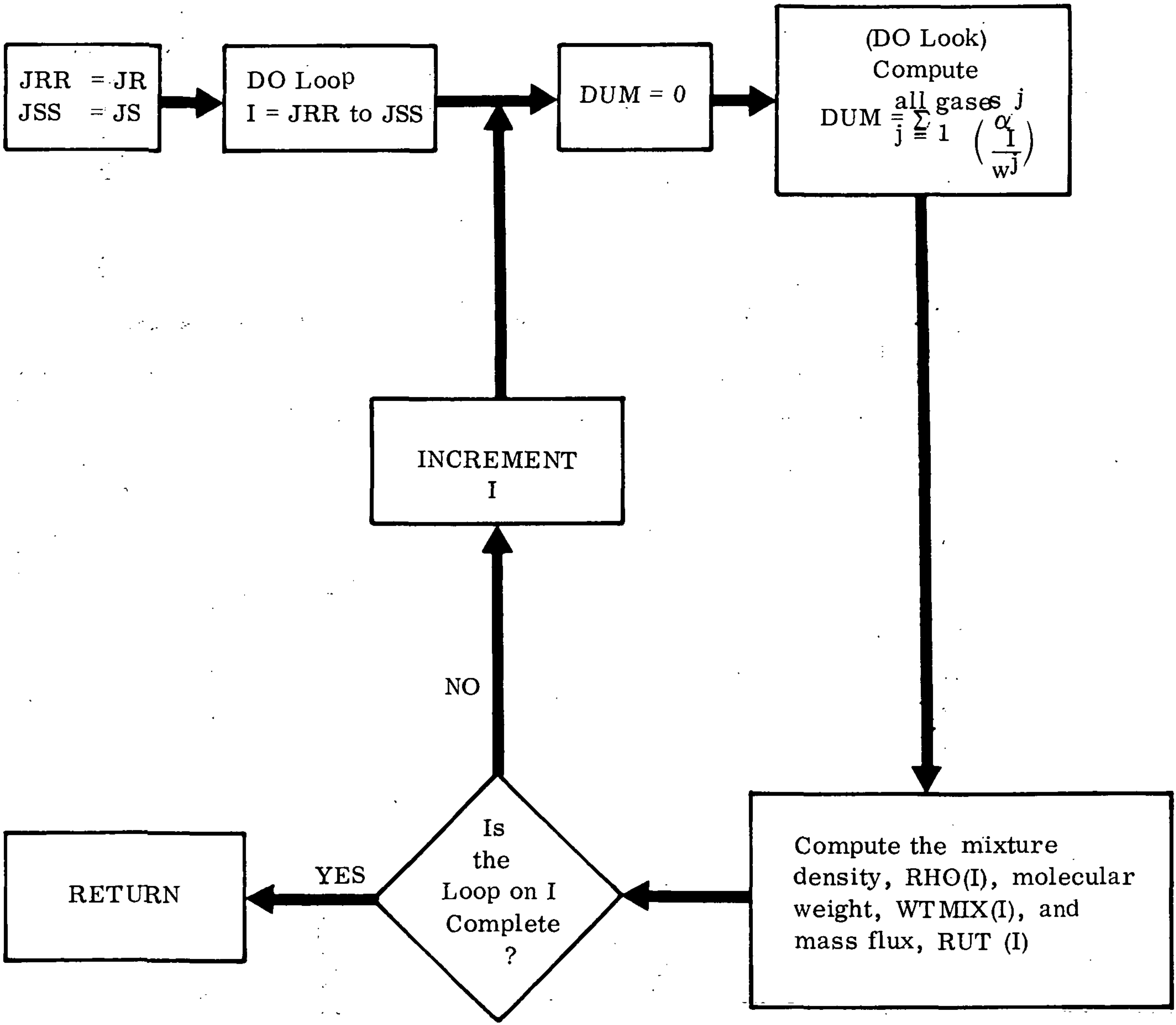
SUBROUTINE CONSRV (Cont.)


b
Is Equilibrium Chemistry being used ?
NO
(DO Loop) Solve Diffusion Conservation Equation on the Centerline in Terms of species, $\alpha^j_{n+1,1}$ (RYSPEC(J,1)), including Chemical kinetic effects, if any
YES
(DO Loop) Solve Diffusion Conservation Equation on the Centerline in Terms of elements, $\alpha^k_{n+1,1}$ (RALPHA (J, 1))
Specie Diffusion Equation (DO Loop)
$\alpha^j_{n+1,1} = \alpha^j_{n,1} + \dot{w}^j_{n,1}$
NO
21
Momentun Equation
$u_{n+1,1} = u_{n,1}$
Energy Equation
$H_{n+1,1} = H_{n,1}$
Is Equilibrium Chemistry being used ?
YES
Element Diffusion Equation (DO Loop)
$\tilde{\alpha}^k_{n+1,1} = \tilde{\alpha}^k_{n,1}$
Specie Diffusion Equation (DO Loop)
$\alpha^j_{n+1,MPSI} = \alpha^j_{n,MPSI} - \dot{w}^2_{n,MPSI}$
NO
Is Equilibrium Chemistry being used ?
Conservation Equations for Outer Edge of Flow Field:
$u_{n+1,MPSI} = u_{n,MPSI}$
$H_{n+1,MPSI} = H_{n,MPSI}$
22
Element Diffusion Equation (DO Loop)
$\tilde{\alpha}^k_{n+1,MPSI} = \tilde{\alpha}^k_{n,MPSI}$
YES
(DO Loop)
$u_{n+1,I} = u_{n+1,I} - \frac{\Delta x}{(\rho u)_{n,I}} \left(\frac{dP}{dx}\right)_{n+1}$
Is LZ = 0 ?
YES
RETURN
NO
(DO Loop) WRITE RU(I), U(I)RH(I), H(I), TAUT(I), T(I), RT(I), SMALLH(I)
RETURN

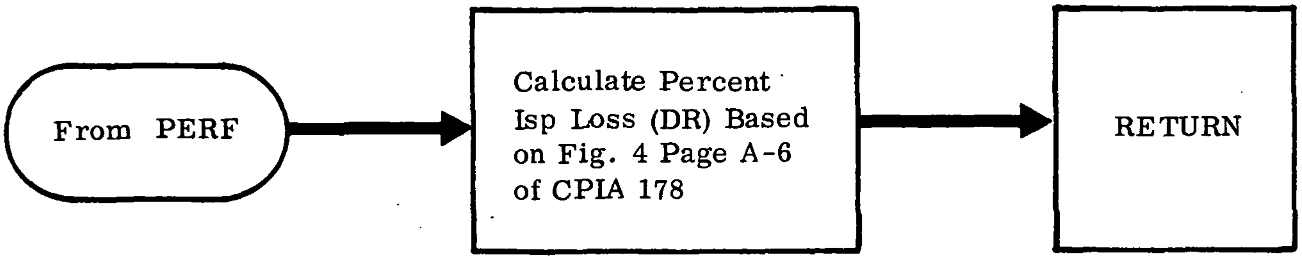
SUBROUTINE CRSPLT

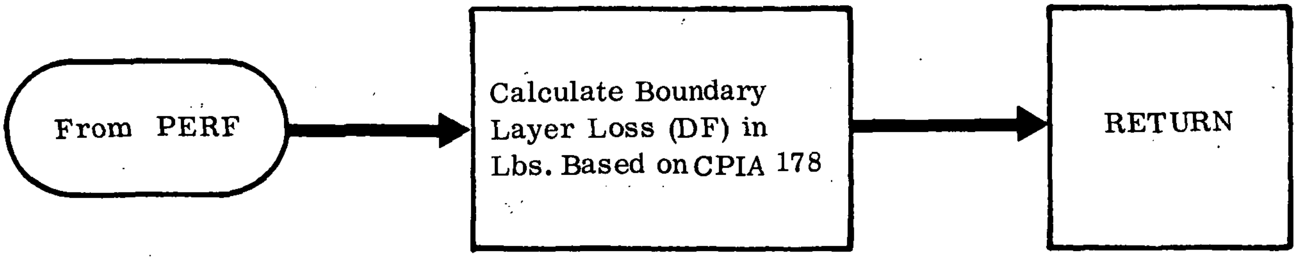

From PERF
Do Loop J = 1, 22, I=1,6 Calculate AF (I, J) where I is O/F = 4,6,8 at High & Low Sides of Chamber Pressure and J is the AR(I) Generated in NOZ
I = 1 J =1, 6 DMIN (I, J) > AF(I, J) ?
YES
SET AFRZ = 1.0 and Continue
NO
CONTINUE
For I = 2, 22 J = 1, 6 Compare AF(I, J) and DMIN (I, J) and Determine Intersection Point By Interpolation
NO Intersection ?
YES
Set AFRZ = 5.0
NO
DO LOOP
CONTINUE
RETURN

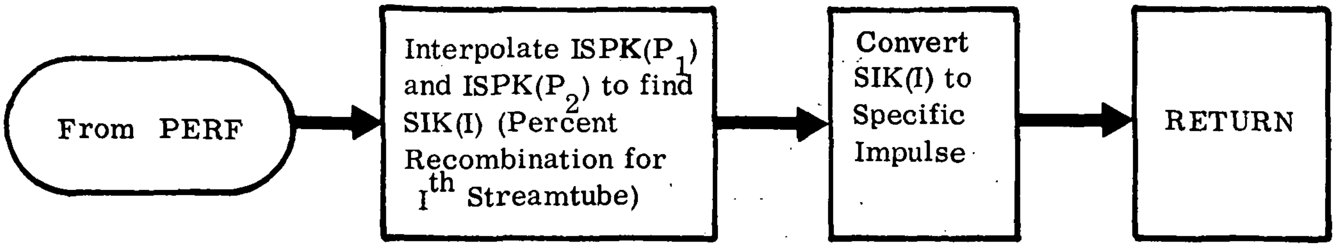
SUBROUTINE DENSE (JR, JS)


JRR = JR
JSS = JS
DO Loop
I = JRR to JSS
DUM = 0
(DO Look)
Compute
DUM = $\sum_{j=1}^{\text{all gases}} \left(\frac{\alpha_I^j}{w^j}\right)$
INCREMENT
I
NO
Is the Loop on I Complete ?
YES
RETURN
Compute the mixture density, RHO(I), molecular weight, WTMIX(I), and mass flux, RUT (I)

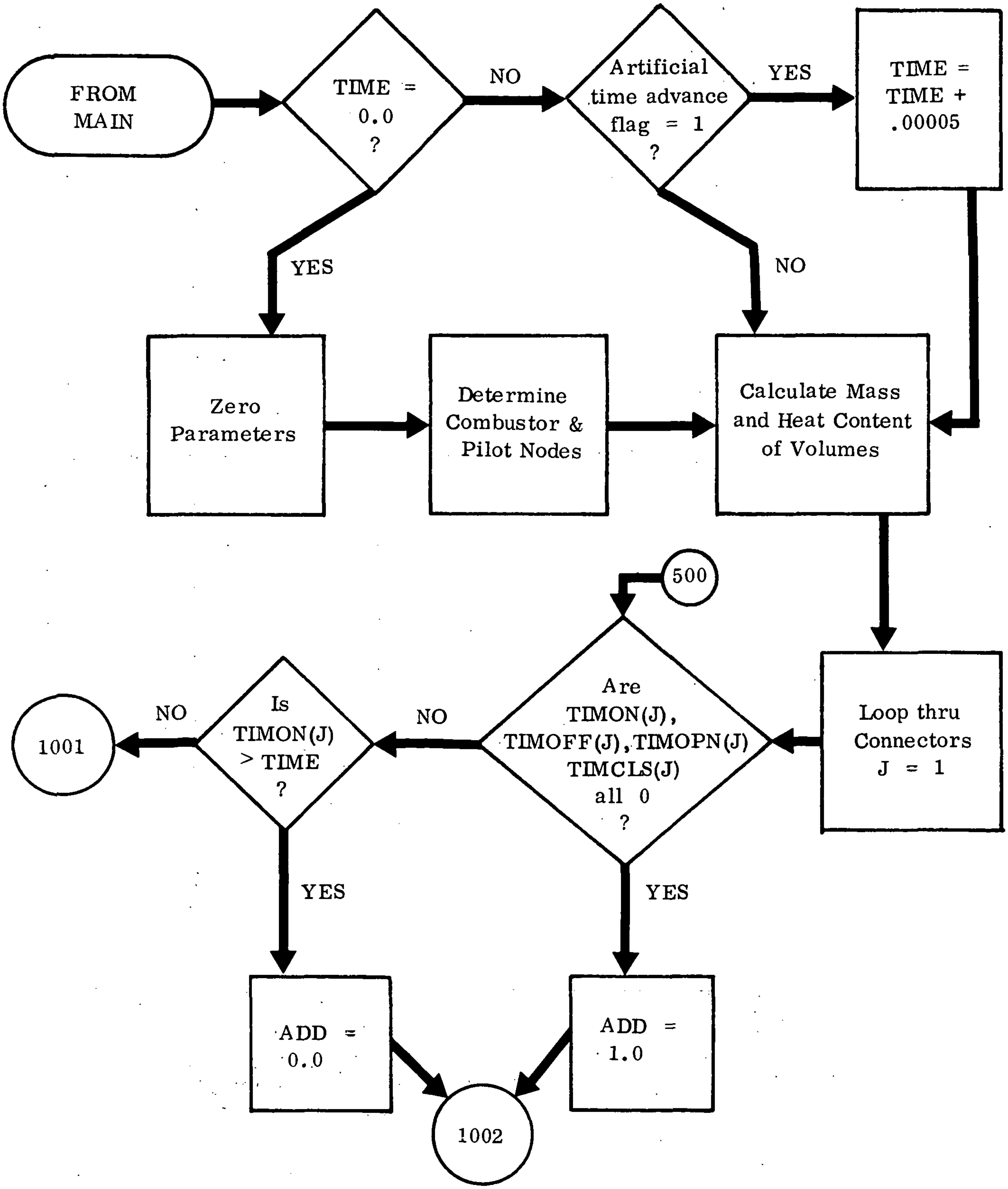
SUBROUTINE DIV, BL, SIDEL


From PERF
Calculate Percent Isp Loss (DR) Based on Fig. 4 Page A-6 of CPIA 178
RETURN

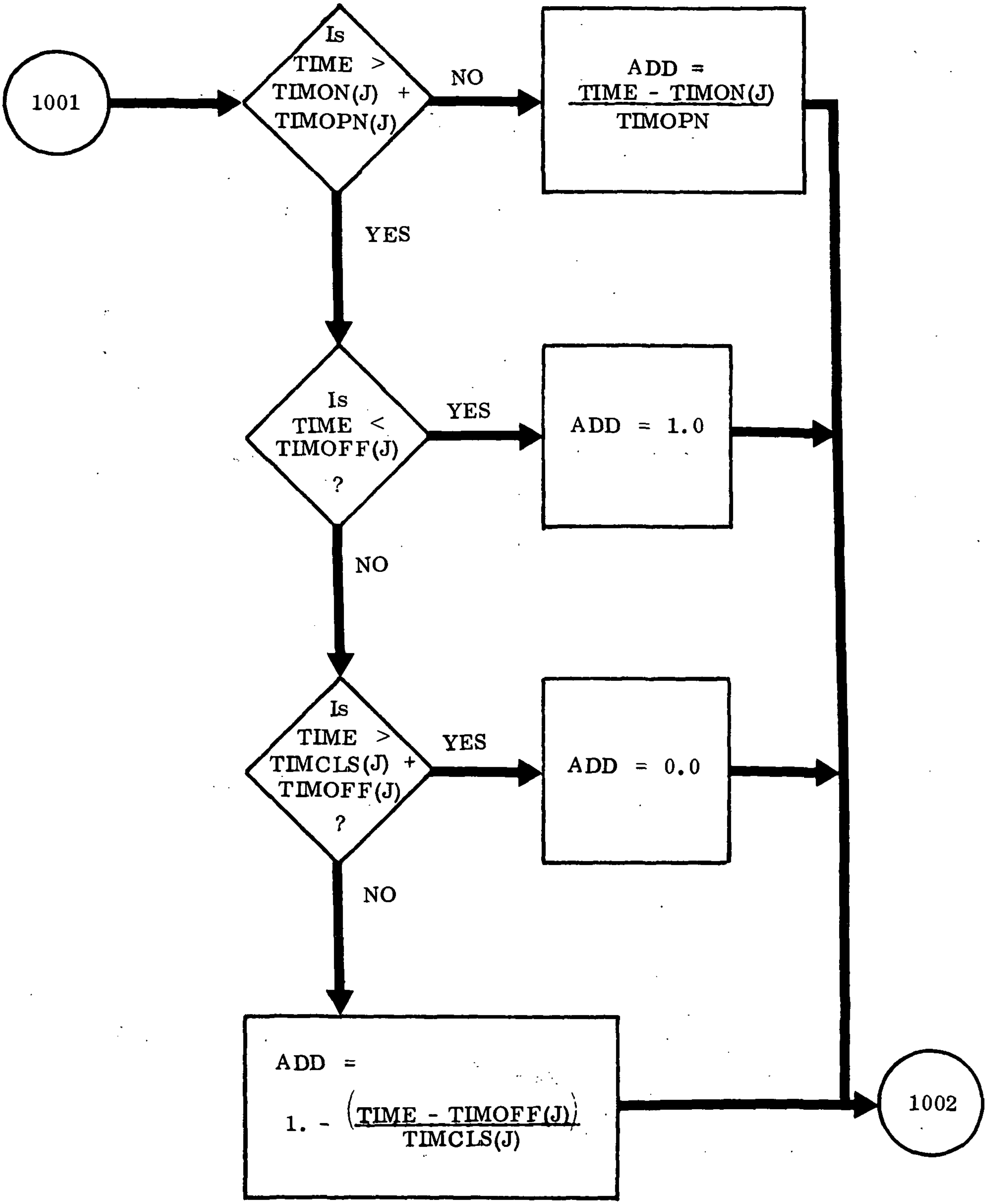

From PERF
Calculate Boundary Layer Loss (DF) in Lbs. Based on CPIA 178
RETURN


From PERF
Interpolate ISPK(P_1) and ISPK(P_2) to find SIK(I) (Percent Recombination for I^{th} Streamtube)
Convert SIK(I) to Specific Impulse
RETURN

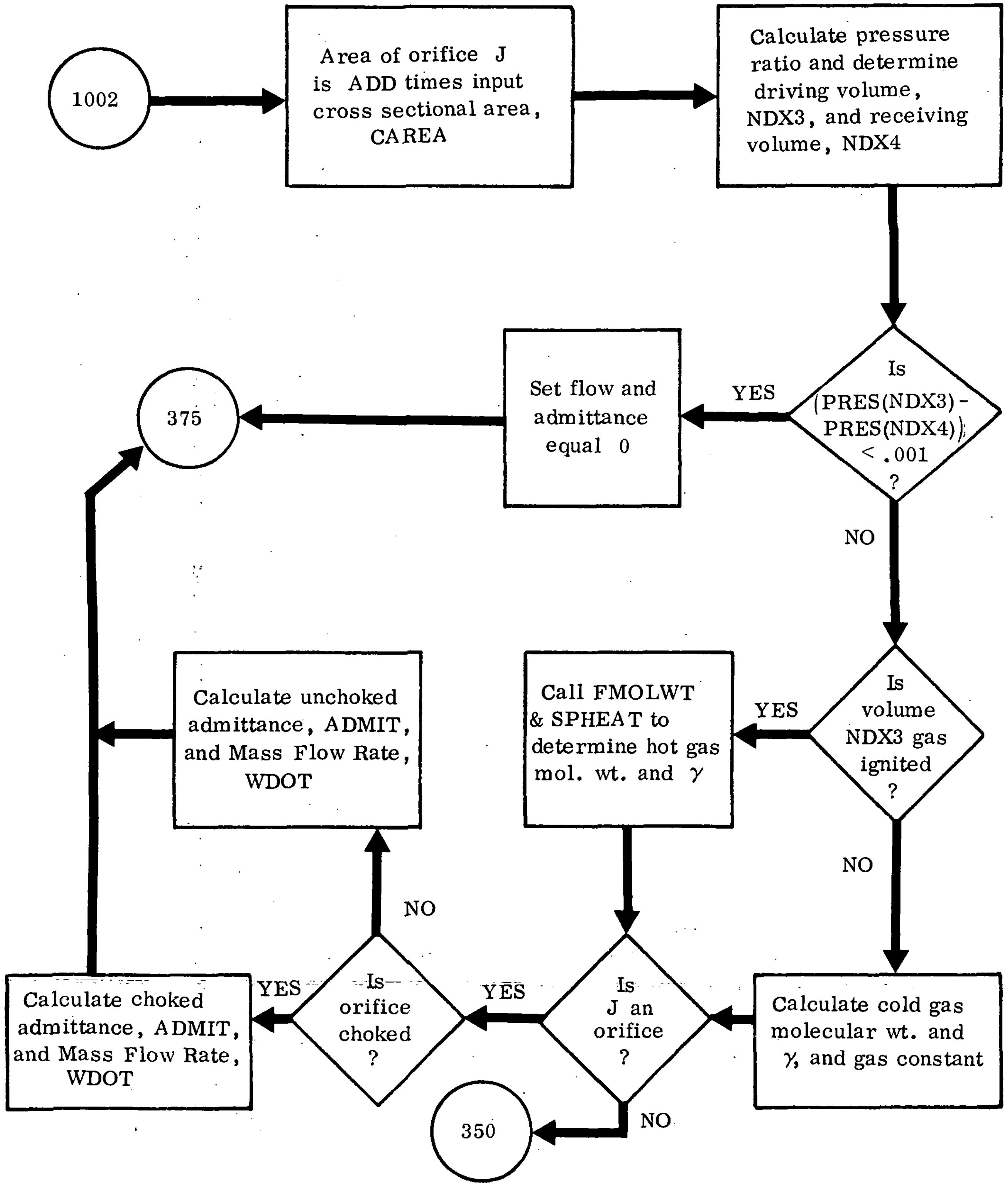
SUBROUTINE DYNAM


FROM MAIN
TIME = 0.0 ?
NO
Artificial time advance flag = 1 ?
YES
TIME = TIME + .00005
YES
NO
Zero Parameters
Determine Combustor & Pilot Nodes
Calculate Mass and Heat Content of Volumes
500
Loop thru Connectors J = 1
Are TIMON(J), TIMOFF(J), TIMOPN(J) TIMCLS(J) all 0 ?
NO
Is TIMON(J) > TIME ?
NO
1001
YES
YES
ADD = 0.0
ADD = 1.0
1002

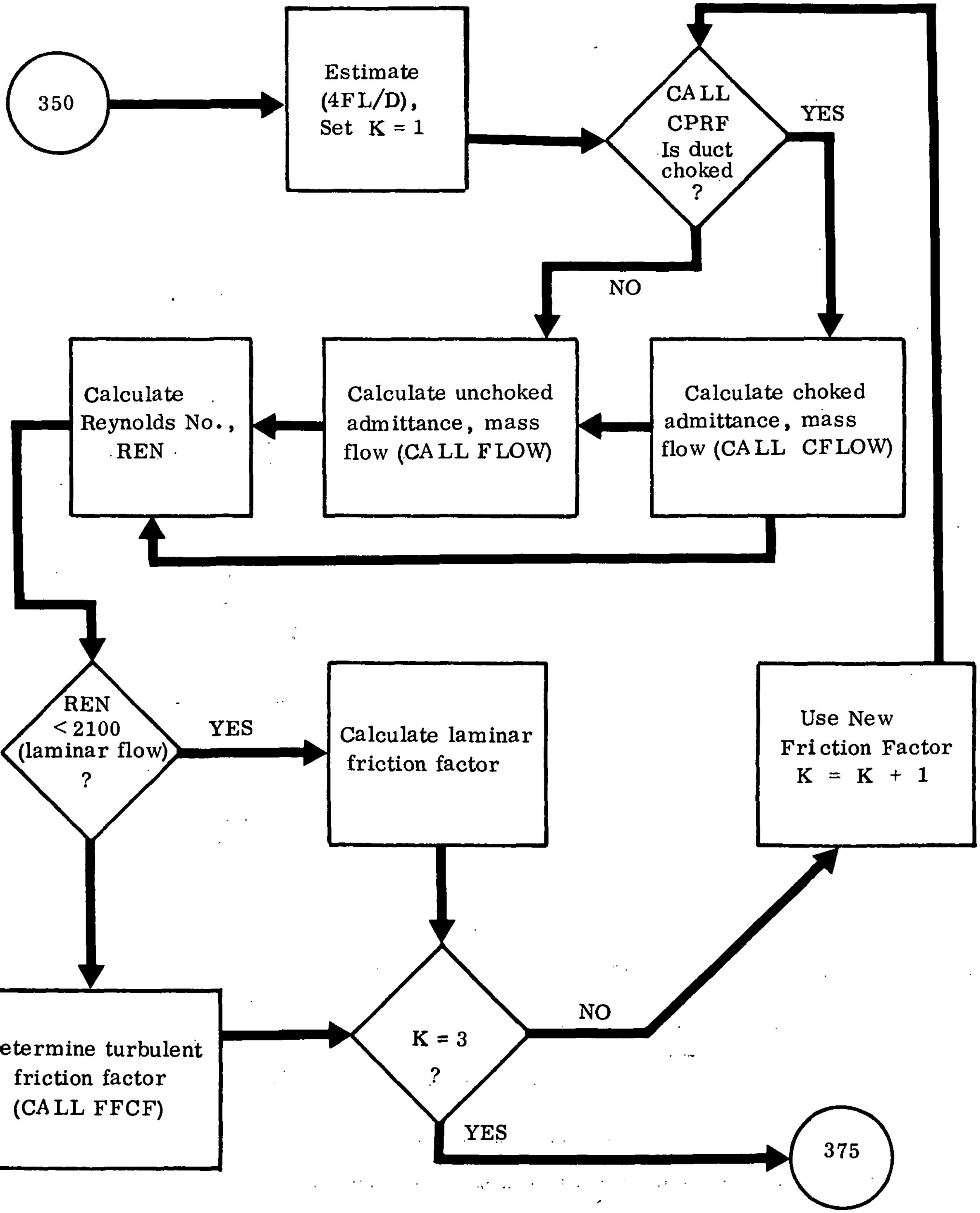
SUBROUTINE DYNAM (Cont.)


1001
Is TIME > TIMON(J) + TIMOPN(J)
NO
ADD = $\frac{\text{TIME - TIMON(J)}}{\text{TIMOPN}}$
YES
Is TIME < TIMOFF(J) ?
YES
ADD = 1.0
NO
Is TIME > TIMCLS(J) + TIMOFF(J) ?
YES
ADD = 0.0
NO
ADD = 1. - $\left(\frac{\text{TIME - TIMOFF(J)}}{\text{TIMCLS(J)}}\right)$
1002

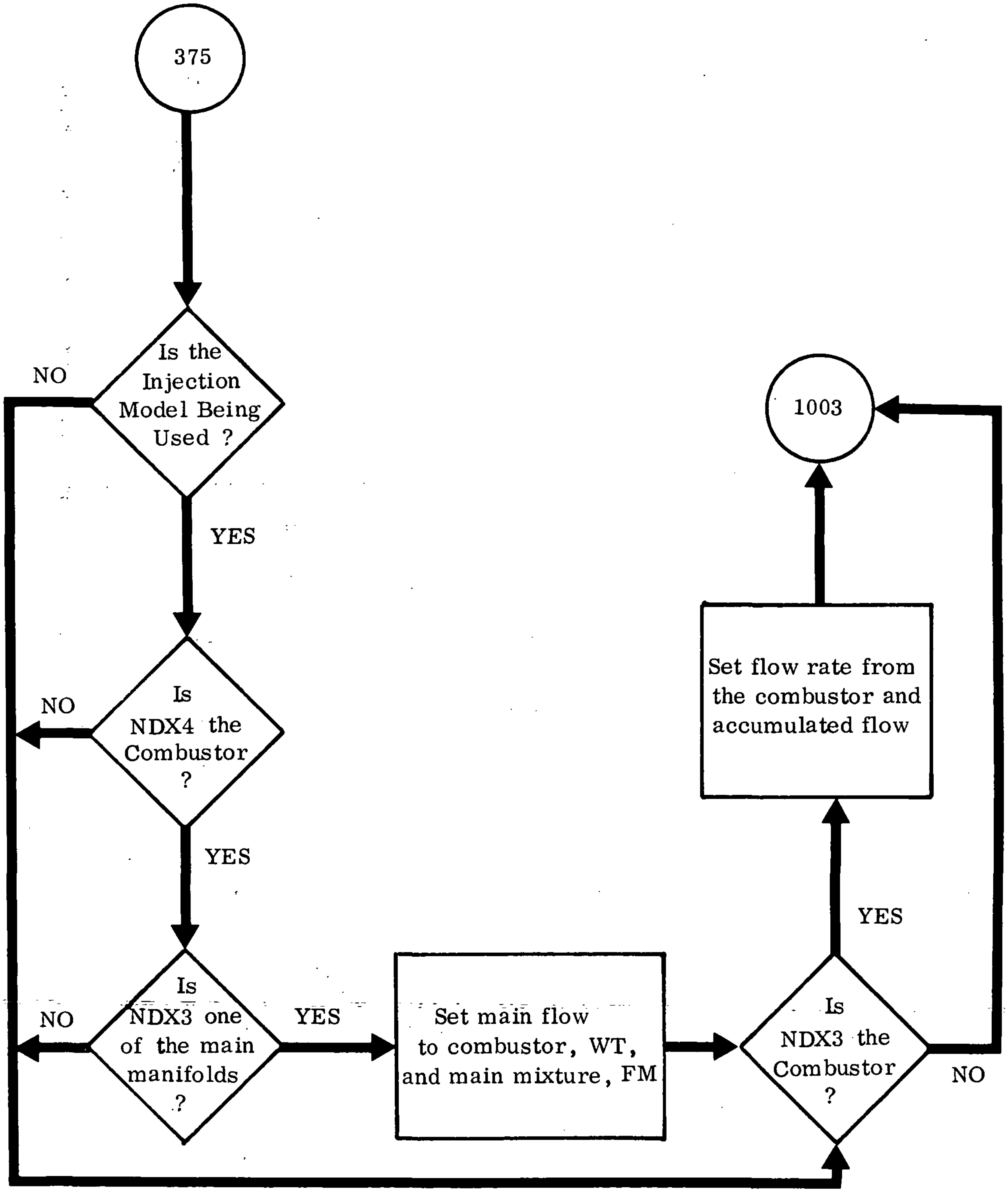
SUBROUTINE DYNAM (Cont.)


1002
Area of orifice J is ADD times input cross sectional area, CAREA
Calculate pressure ratio and determine driving volume, NDX3, and receiving volume, NDX4
Is (PRES(NDX3) - PRES(NDX4)) < .001 ?
YES
Set flow and admittance equal 0
375
NO
Is volume NDX3 gas ignited ?
YES
Call FMOLWT & SPHEAT to determine hot gas mol. wt. and γ
NO
Calculate cold gas molecular wt. and γ, and gas constant
Is J an orifice ?
YES
NO
350
Is orifice choked ?
YES
Calculate choked admittance, ADMIT, and Mass Flow Rate, WDOT
NO
Calculate unchoked admittance, ADMIT, and Mass Flow Rate, WDOT

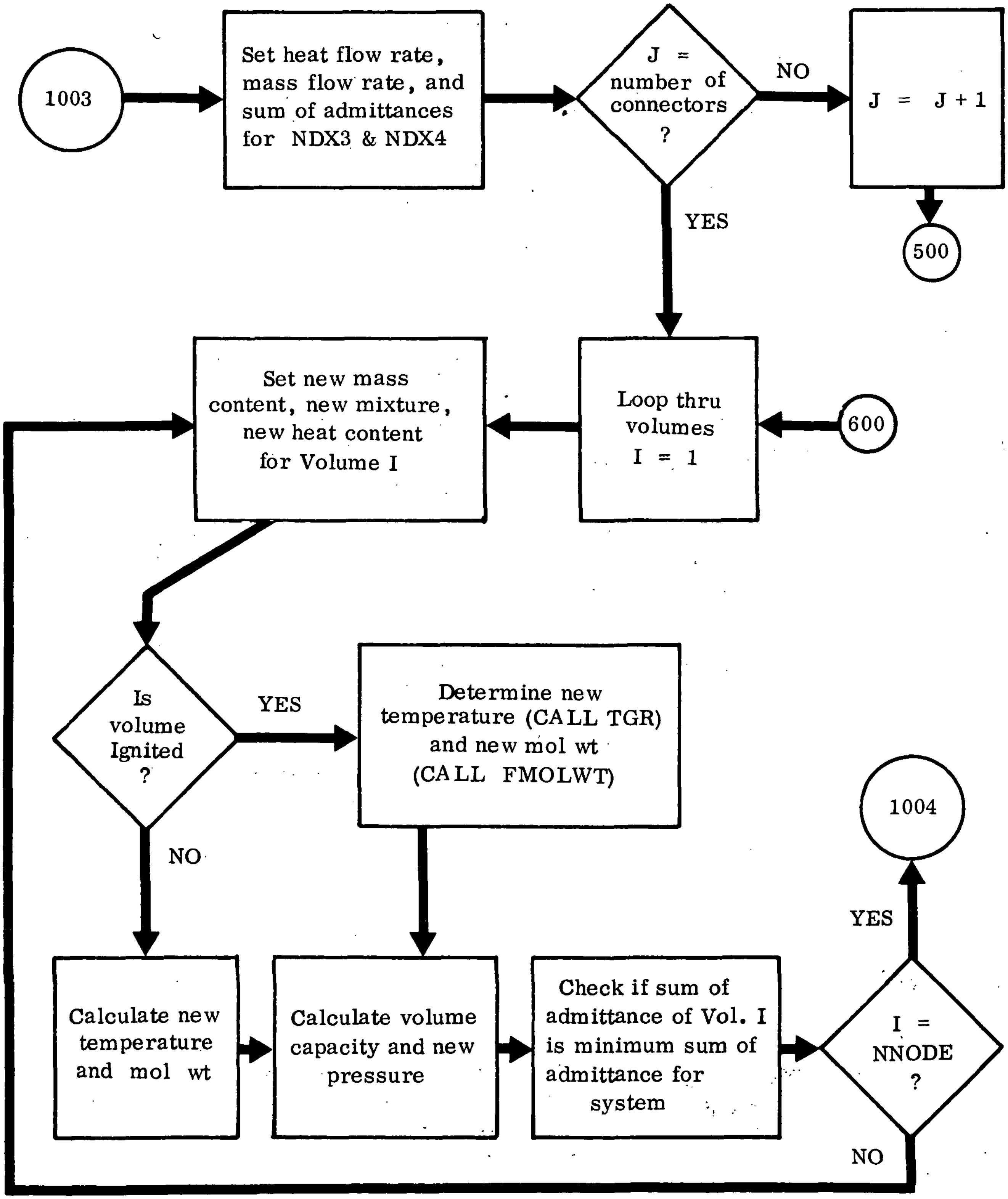
SUBROUTINE DYNAM (Cont.)


350
Estimate (4FL/D), Set K = 1
CALL CPRF Is duct choked ?
YES
NO
Calculate choked admittance, mass flow (CALL CFLOW)
Calculate unchoked admittance, mass flow (CALL FLOW)
Calculate Reynolds No., REN
REN < 2100 (laminar flow) ?
YES
Calculate laminar friction factor
Use New Friction Factor K = K + 1
Determine turbulent friction factor (CALL FFCF)
K = 3 ?
NO
YES
375

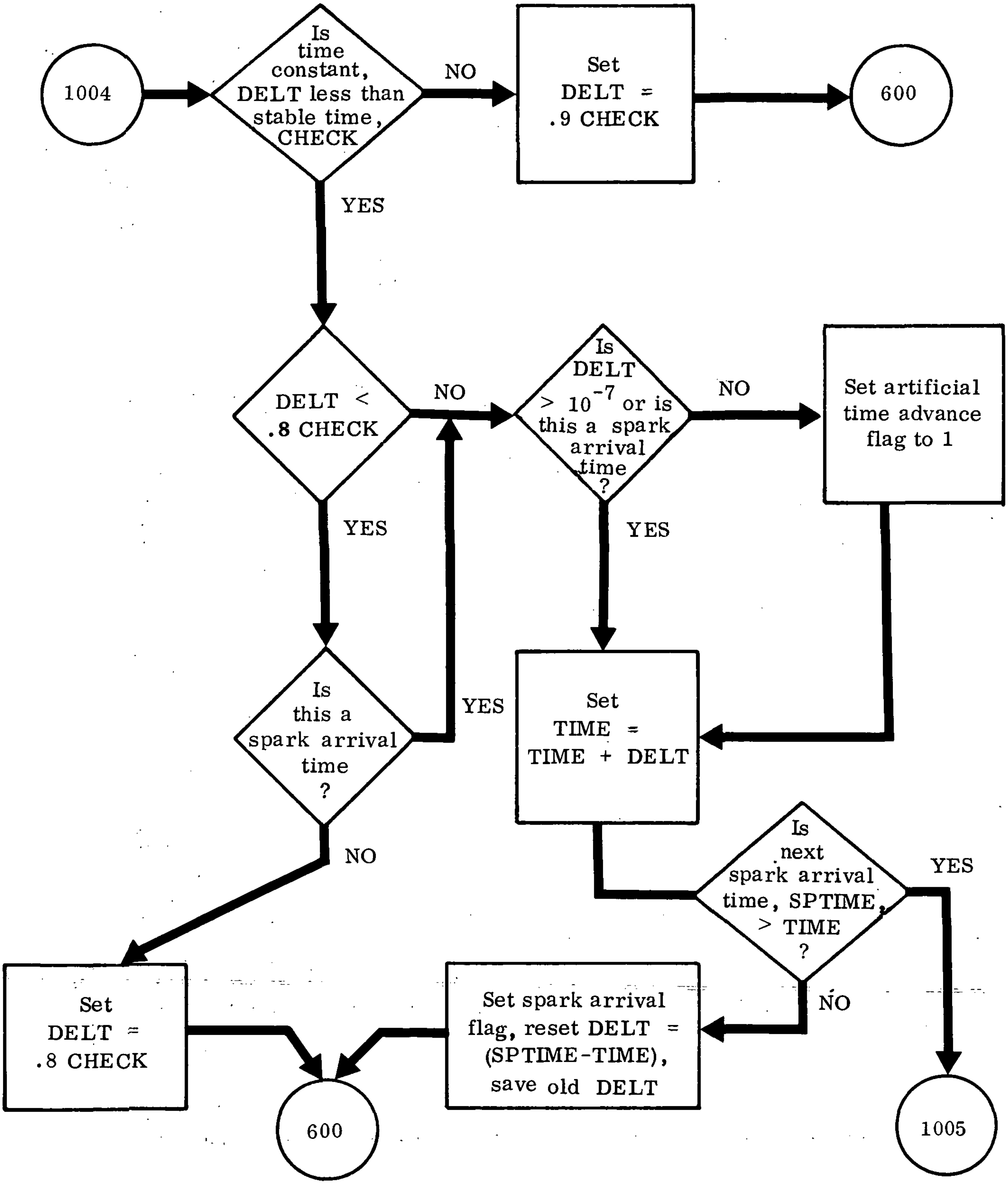
SUBROUTINE DYNAM (Cont.)


375
Is the Injection Model Being Used ?
NO
YES
Is NDX4 the Combustor ?
NO
YES
Is NDX3 one of the main manifolds ?
NO
YES
Set main flow to combustor, WT, and main mixture, FM
Is NDX3 the Combustor ?
NO
YES
Set flow rate from the combustor and accumulated flow
1003

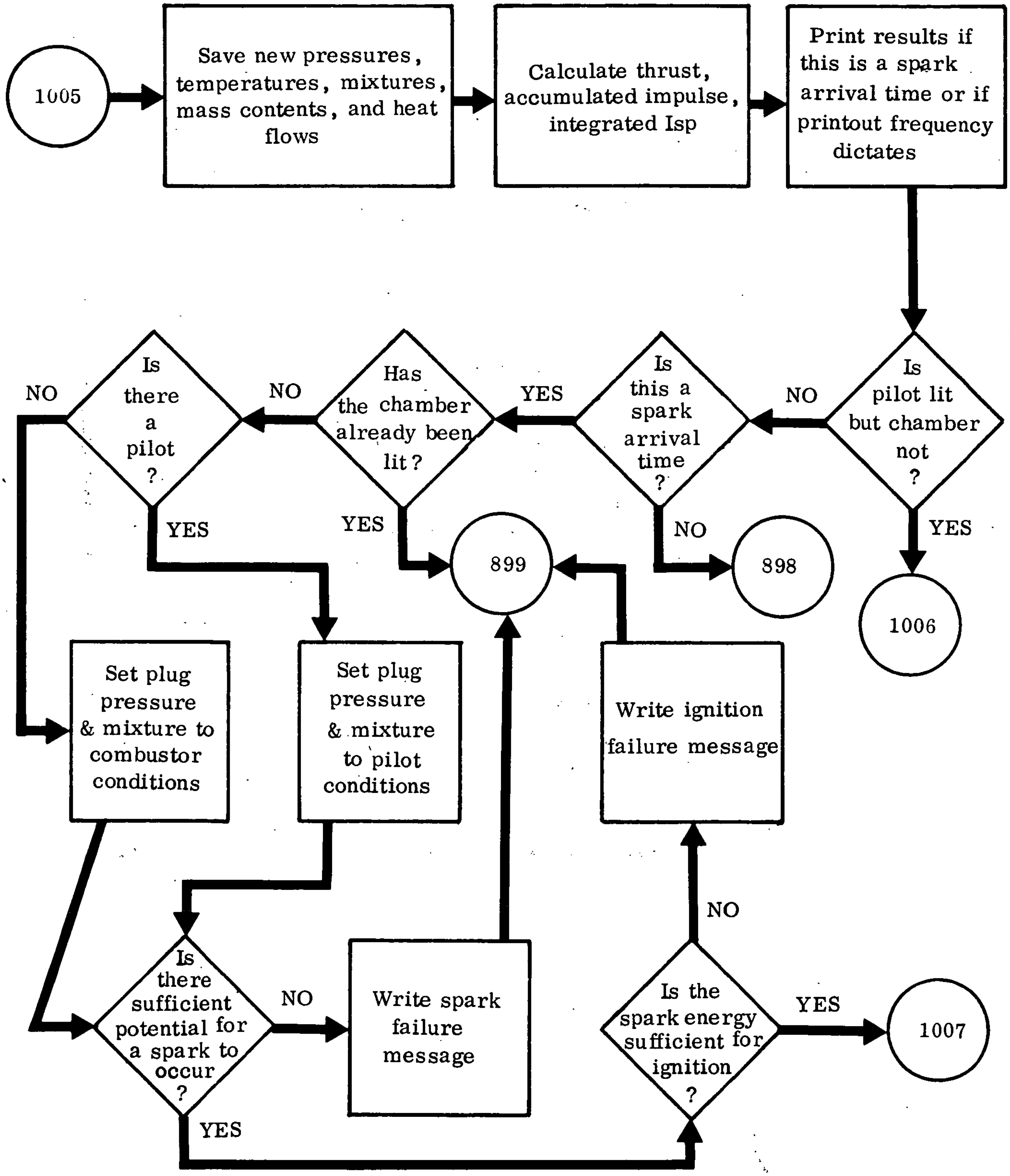
SUBROUTINE DYNAM (Cont.)


1003
Set heat flow rate, mass flow rate, and sum of admittances for NDX3 & NDX4
J = number of connectors ?
NO
J = J + 1
500
YES
Set new mass content, new mixture, new heat content for Volume I
Loop thru volumes I = 1
600
Is volume Ignited ?
YES
Determine new temperature (CALL TGR) and new mol wt (CALL FMOLWT)
1004
NO
YES
Calculate new temperature and mol wt
Calculate volume capacity and new pressure
Check if sum of admittance of Vol. I is minimum sum of admittance for system
I = NNODE ?
NO

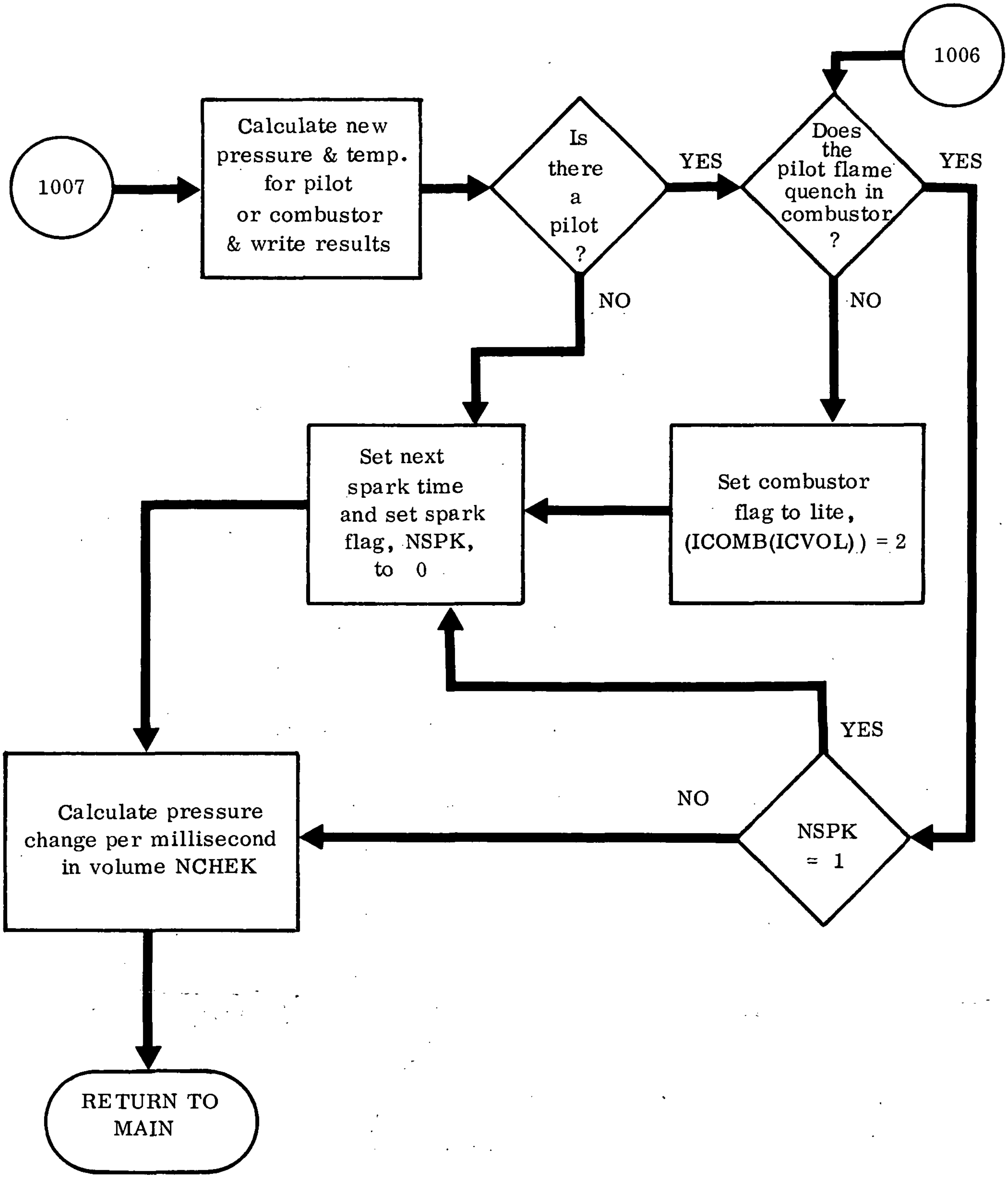
SUBROUTINE DYNAM (Cont.)


1004
Is time constant, DELT less than stable time, CHECK
NO
Set DELT = .9 CHECK
600
YES
DELT < .8 CHECK
NO
Is DELT > 10^{-7} or is this a spark arrival time ?
NO
Set artificial time advance flag to 1
YES
YES
Is this a spark arrival time ?
YES
Set TIME = TIME + DELT
NO
Is next spark arrival time, SPTIME, > TIME ?
YES
Set DELT = .8 CHECK
NO
Set spark arrival flag, reset DELT = (SPTIME-TIME), save old DELT
600
1005

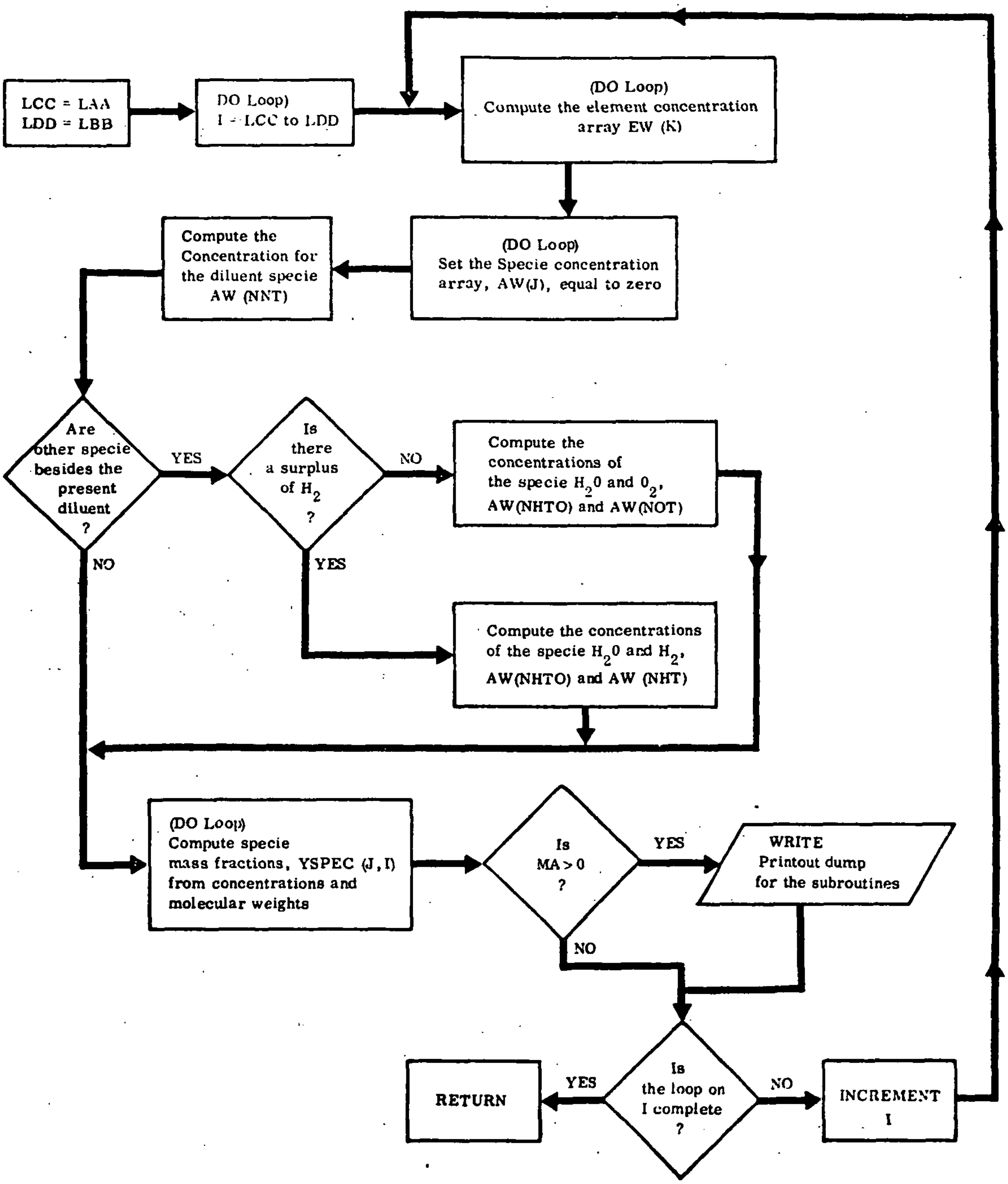
SUBROUTINE DYNAM (Cont.)


1005
Save new pressures, temperatures, mixtures, mass contents, and heat flows
Calculate thrust, accumulated impulse, integrated Isp
Print results if this is a spark arrival time or if printout frequency dictates
Is pilot lit but chamber not ?
YES
1006
NO
Is this a spark arrival time ?
NO
898
YES
Has the chamber already been lit ?
YES
899
NO
Is there a pilot ?
YES
NO
Set plug pressure & mixture to combustor conditions
Set plug pressure & mixture to pilot conditions
Is there sufficient potential for a spark to occur ?
NO
Write spark failure message
YES
Is the spark energy sufficient for ignition ?
YES
1007
NO
Write ignition failure message

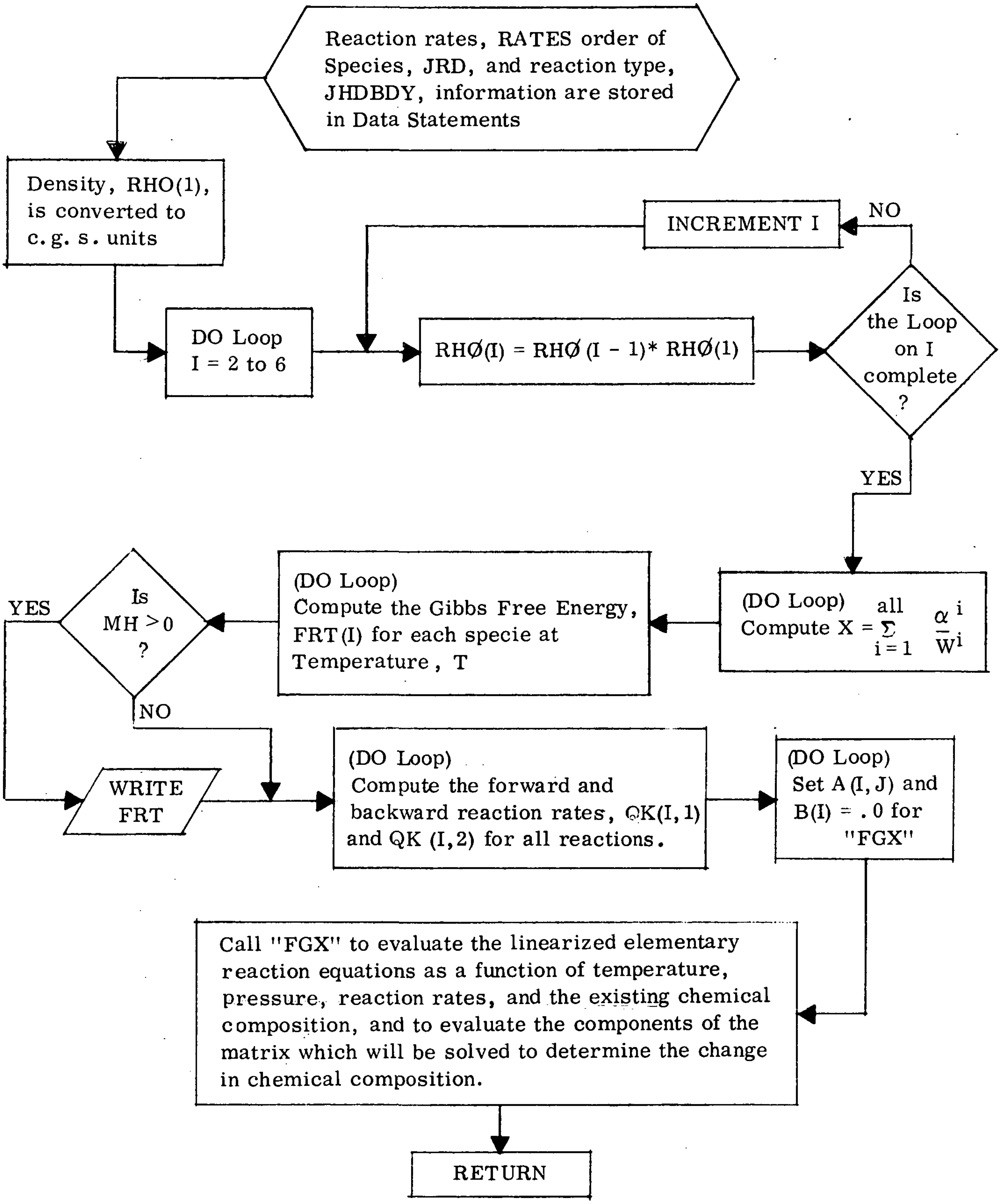
SUBROUTINE DYNAM (Cont.)


1006
1007
Calculate new pressure & temp. for pilot or combustor & write results
Is there a pilot ?
YES
Does the pilot flame quench in combustor ?
YES
NO
NO
Set next spark time and set spark flag, NSPK, to 0
Set combustor flag to lite, (ICOMB(ICVOL)) = 2
YES
NO
NSPK = 1
Calculate pressure change per millisecond in volume NCHEK
RETURN TO MAIN

SUBROUTINE EQUILC (LAA, LBB)


LCC = LAA
LDD = LBB
DO Loop)
I = LCC to LDD
(DO Loop)
Compute the element concentration array EW (K)
(DO Loop)
Set the Specie concentration array, AW(J), equal to zero
Compute the Concentration for the diluent specie AW (NNT)
Are other specie besides the present diluent ?
YES
NO
Is there a surplus of H2 ?
NO
YES
Compute the concentrations of the specie H20 and 02, AW(NHTO) and AW(NOT)
Compute the concentrations of the specie H20 and H2, AW(NHTO) and AW (NHT)
(DO Loop)
Compute specie mass fractions, YSPEC (J,I) from concentrations and molecular weights
Is MA > 0 ?
YES
NO
WRITE
Printout dump for the subroutines
Is the loop on I complete ?
YES
NO
RETURN
INCREMENT I

SUBROUTINE FG


Reaction rates, RATES order of Species, JRD, and reaction type, JHDBDY, information are stored in Data Statements
Density, RHO(1), is converted to c.g.s. units
DO Loop I = 2 to 6
INCREMENT I
NO
RHØ(I) = RHØ (I - 1)* RHØ(1)
Is the Loop on I complete ?
YES
(DO Loop) Compute $X = \sum_{i=1}^{all} \frac{\alpha_i}{W_i}$
(DO Loop) Compute the Gibbs Free Energy, FRT(I) for each specie at Temperature, T
Is MH > 0 ?
YES
NO
WRITE FRT
(DO Loop) Compute the forward and backward reaction rates, QK(I,1) and QK (I,2) for all reactions.
(DO Loop) Set A (I, J) and B(I) = .0 for "FGX"
Call "FGX" to evaluate the linearized elementary reaction equations as a function of temperature, pressure, reaction rates, and the existing chemical composition, and to evaluate the components of the matrix which will be solved to determine the change in chemical composition.
RETURN

SUBROUTINE FGX

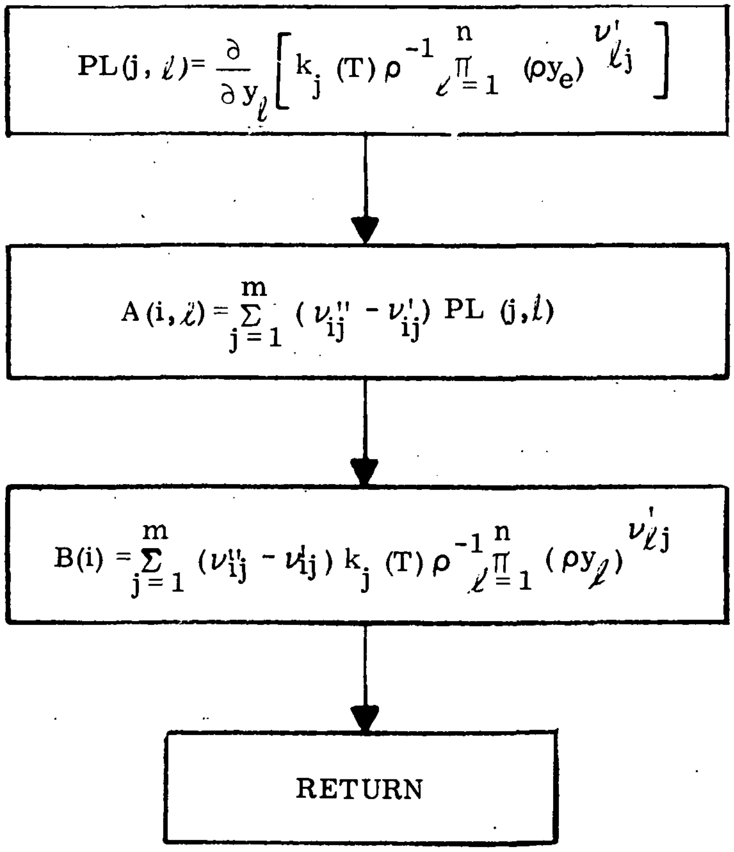
The linearized kinetics equations are of the form

$$\dot{y}_i = A_{ij}\,\Delta y_j + B_i$$

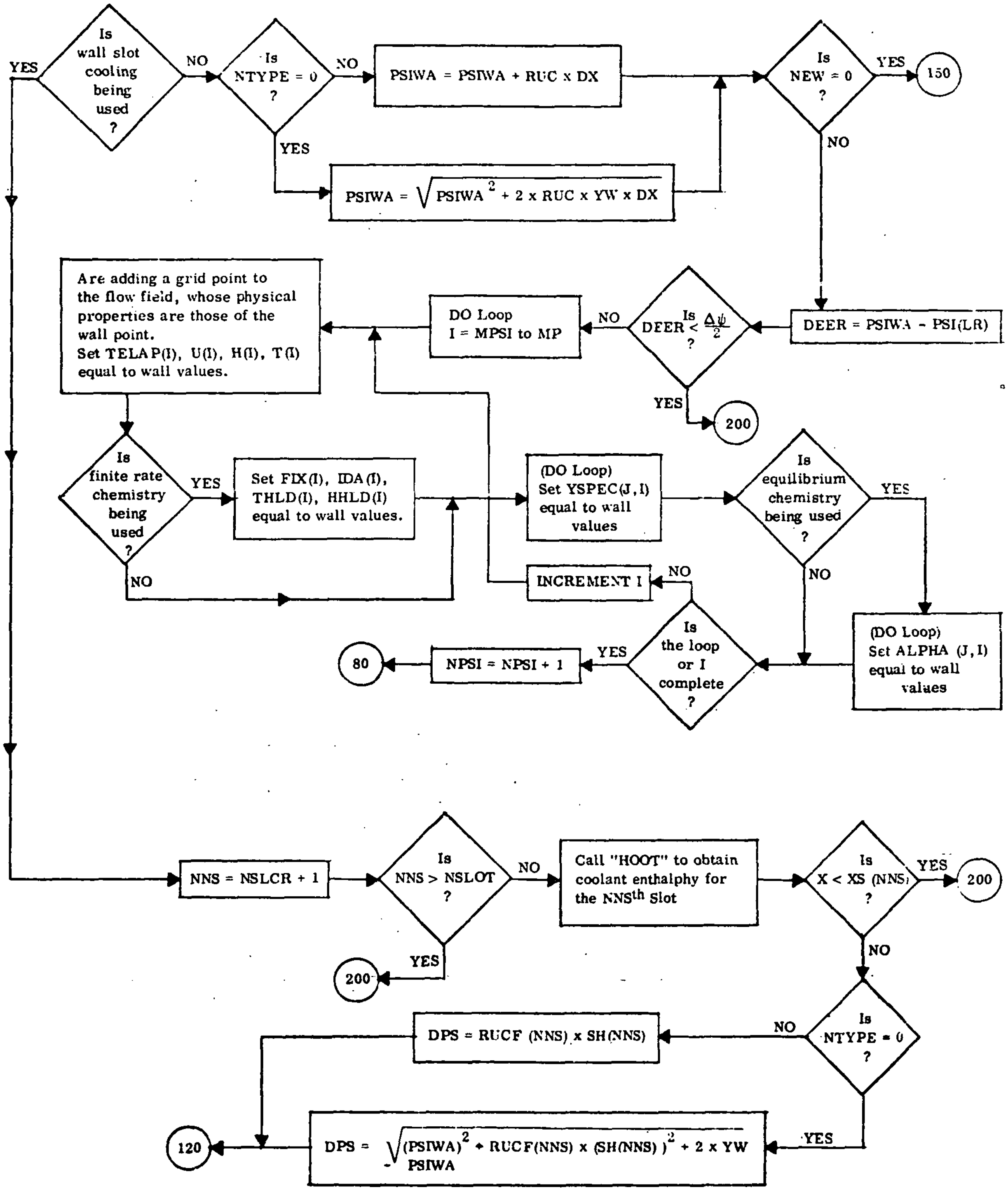
where $A_{ij} = \dfrac{\partial f_i}{\partial y_j}\,(t_o, y_o)$ and $B_i = f_i\,(t_o, y_o)$

and $f_i = \sum_{j=1}^{m} (\nu''_{ij} - \nu'_{ij})\, k_j\,(T)\, \rho^{-1} \prod_{\ell=1}^{n} (\rho\, y_e)^{\nu'_{\ell j}}$

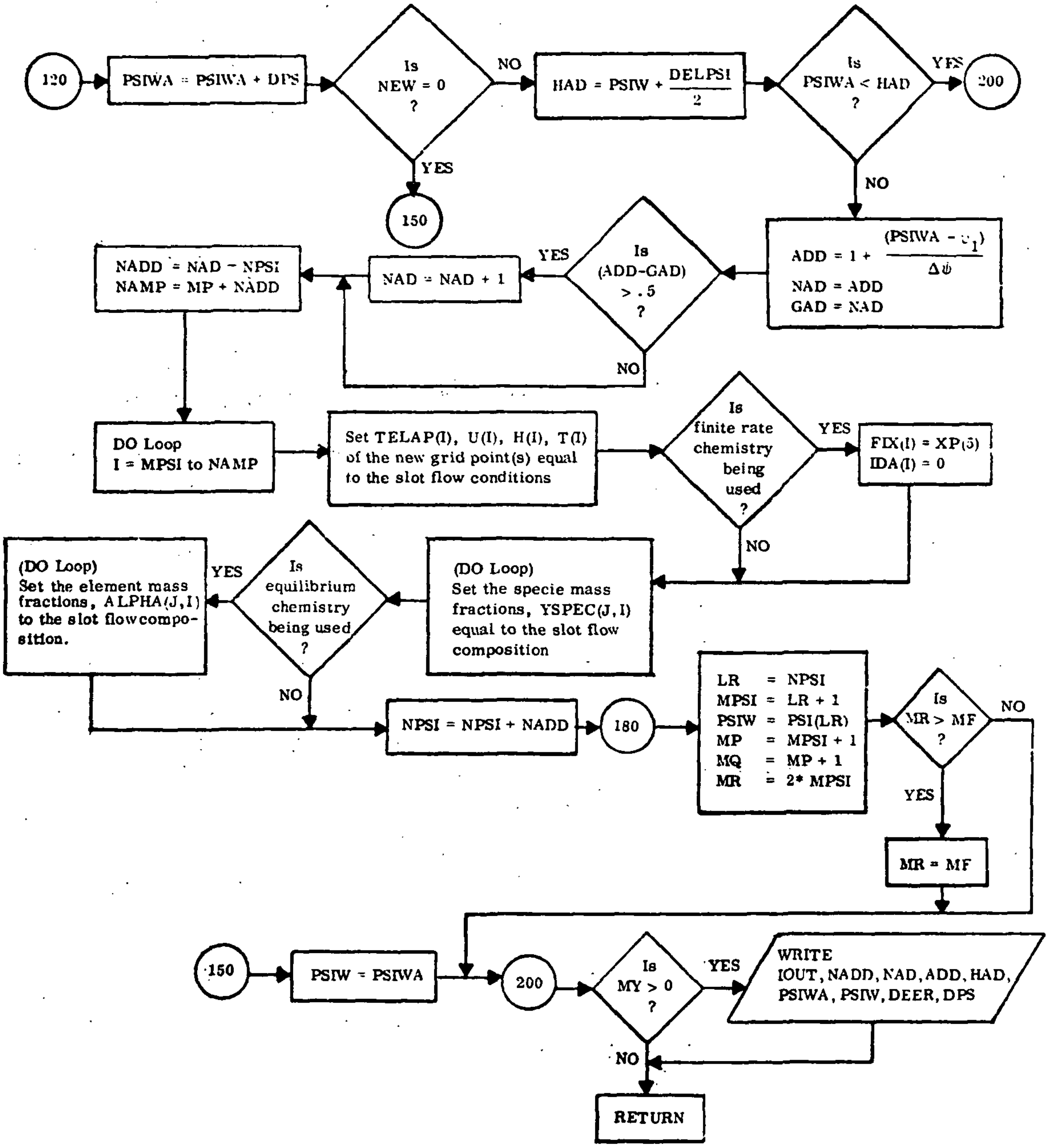
In this subroutine A_{ij} and B_i are evaluated for (T, ρ, t_o, y_o^i)


$PL(j, \ell) = \dfrac{\partial}{\partial y_\ell}\left[k_j\,(T)\,\rho^{-1} \prod_{\ell=1}^{n} (\rho y_e)^{\nu'_{\ell j}}\right]$
$A(i, \ell) = \sum_{j=1}^{m} (\nu''_{ij} - \nu'_{ij})\; PL\,(j, \ell)$
$B(i) = \sum_{j=1}^{m} (\nu''_{ij} - \nu'_{ij})\, k_j\,(T)\,\rho^{-1} \prod_{\ell=1}^{n} (\rho y_\ell)^{\nu'_{\ell j}}$
RETURN

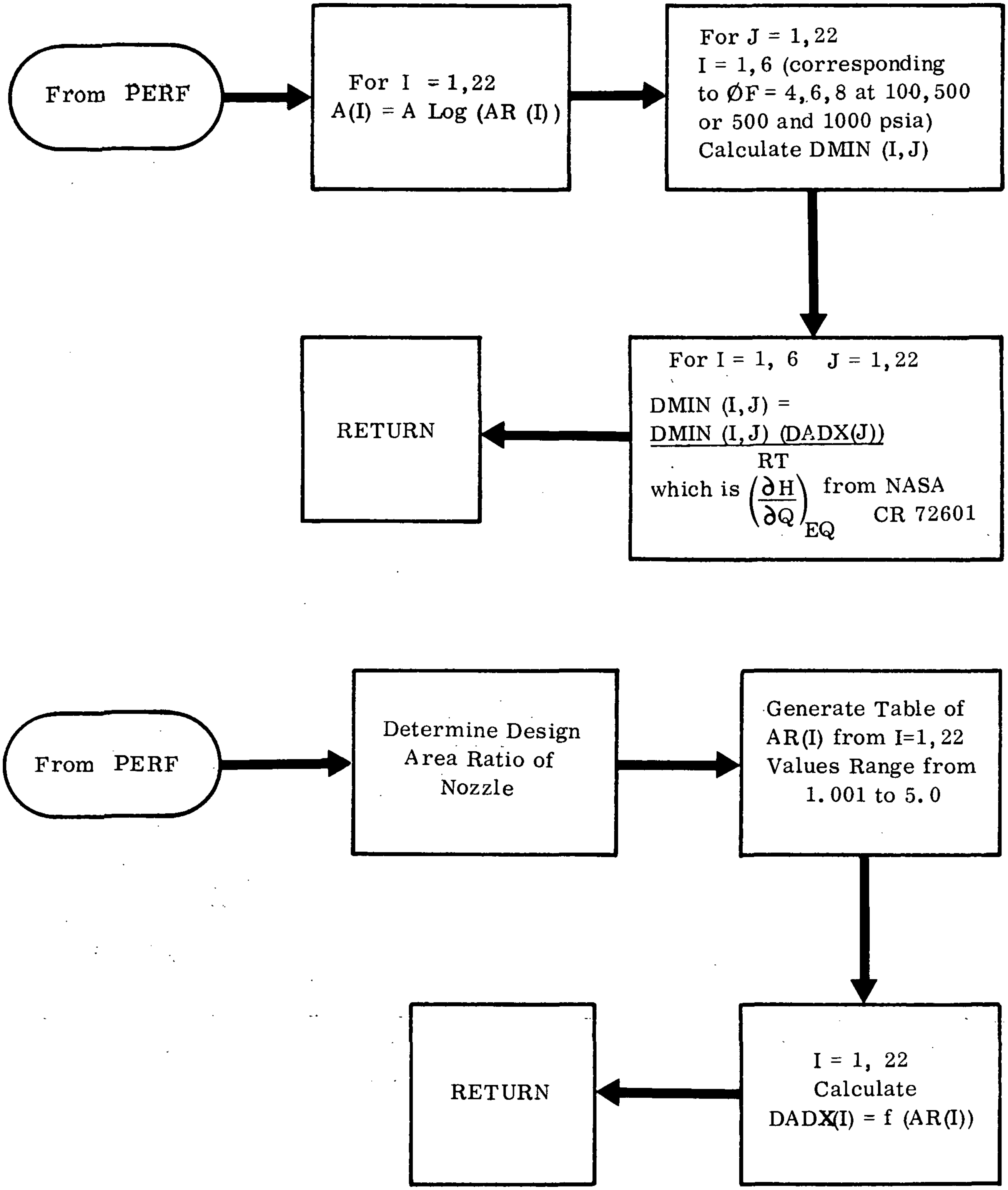
SUBROUTINE FLUX


Is wall slot cooling being used ?
YES
NO
Is NTYPE = 0 ?
NO
YES
PSIWA = PSIWA + RUC x DX
PSIWA = $\sqrt{\text{PSIWA}^2 + 2 \times \text{RUC} \times \text{YW} \times \text{DX}}$
Is NEW = 0 ?
YES
150
NO
DEER = PSIWA - PSI(LR)
Is DEER < $\frac{\Delta\psi}{2}$?
NO
YES
200
DO Loop I = MPSI to MP
Are adding a grid point to the flow field, whose physical properties are those of the wall point. Set TELAP(I), U(I), H(I), T(I) equal to wall values.
Is finite rate chemistry being used ?
YES
NO
Set FIX(I), IDA(I), THLD(I), HHLD(I) equal to wall values.
(DO Loop) Set YSPEC(J,I) equal to wall values
Is equilibrium chemistry being used ?
YES
NO
(DO Loop) Set ALPHA (J,I) equal to wall values
INCREMENT I
NO
Is the loop or I complete ?
YES
NPSI = NPSI + 1
80
NNS = NSLCR + 1
Is NNS > NSLOT ?
NO
YES
200
Call "HOOT" to obtain coolant enthalphy for the NNSth Slot
Is X < XS (NNS) ?
YES
200
NO
Is NTYPE = 0 ?
NO
YES
DPS = RUCF (NNS) x SH(NNS)
DPS = $\sqrt{\frac{(\text{PSIWA})^2 + \text{RUCF(NNS)} \times (\text{SH(NNS)})^2 + 2 \times \text{YW}}{\text{PSIWA}}}$
120

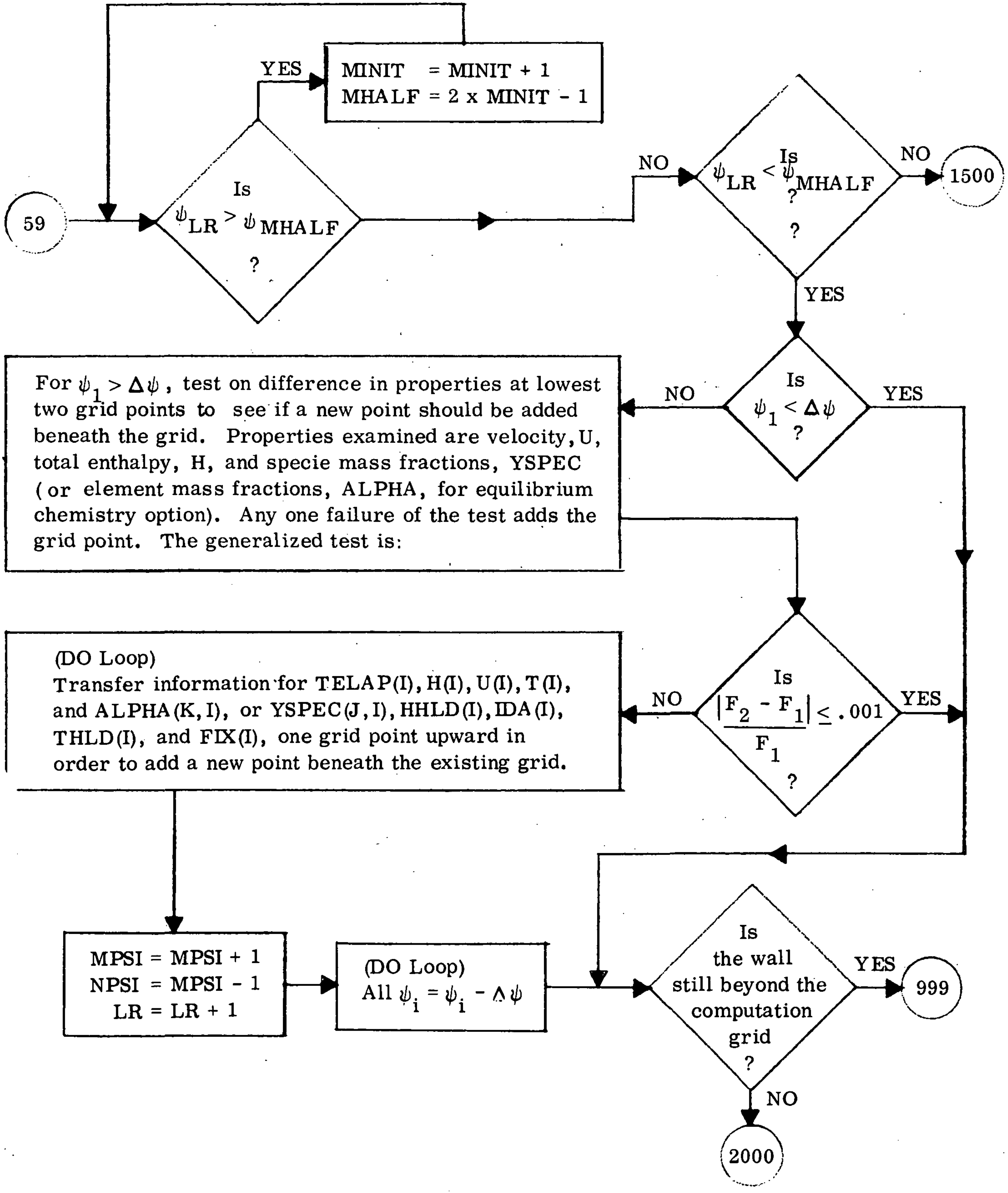
SUBROUTINE FLUX (Cont.)



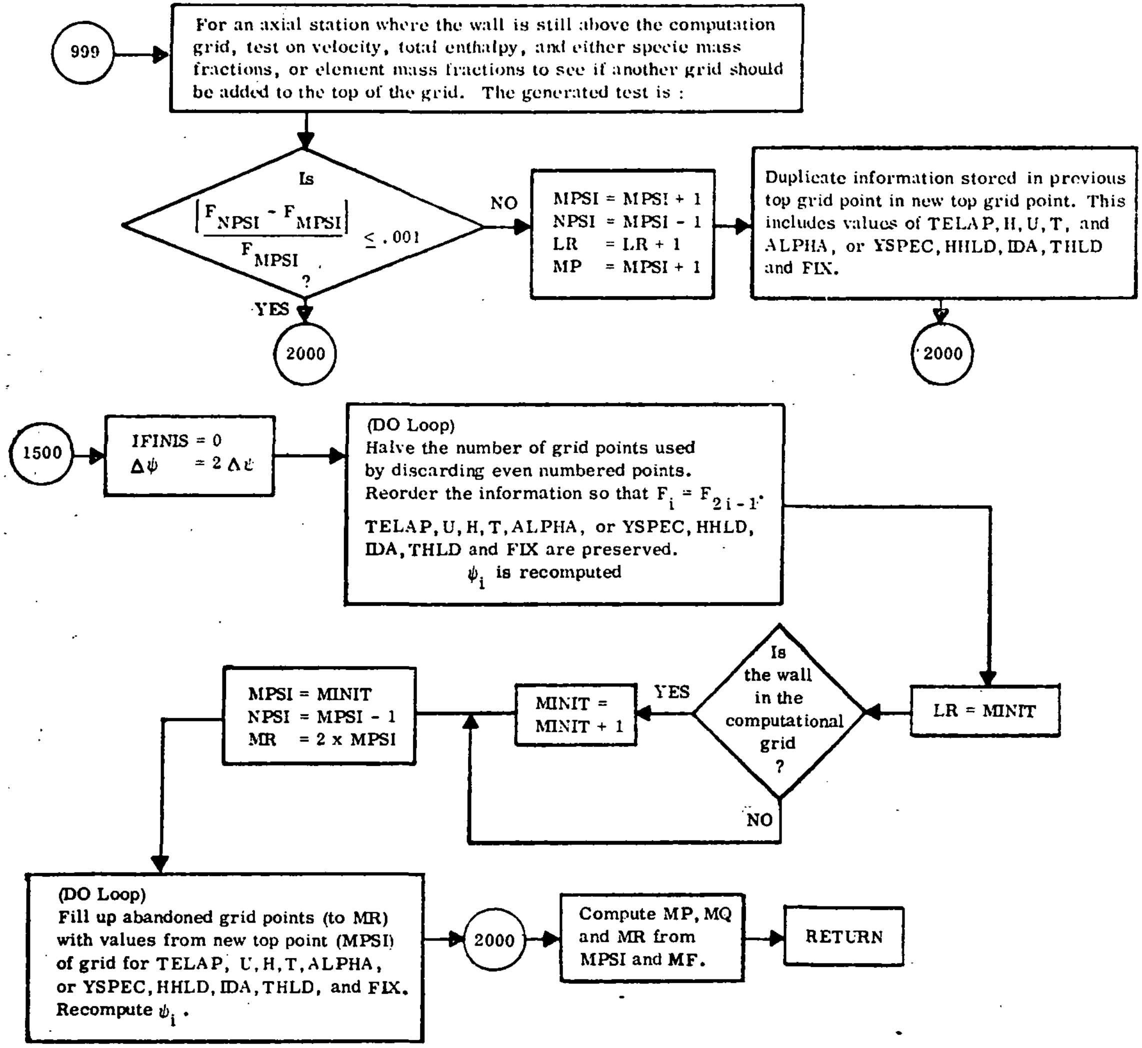
SUBROUTINE GRAD, NOZ


From PERF
For I = 1,22
A(I) = A Log (AR (I))
For J = 1,22
I = 1,6 (corresponding to ØF = 4,6,8 at 100,500 or 500 and 1000 psia)
Calculate DMIN (I,J)
For I = 1, 6 J = 1,22
DMIN (I,J) = DMIN (I,J) (DADX(J)) / RT
which is (∂H/∂Q)EQ from NASA CR 72601
RETURN
From PERF
Determine Design Area Ratio of Nozzle
Generate Table of AR(I) from I=1,22 Values Range from 1.001 to 5.0
I = 1, 22
Calculate
DADX(I) = f (AR(I))
RETURN

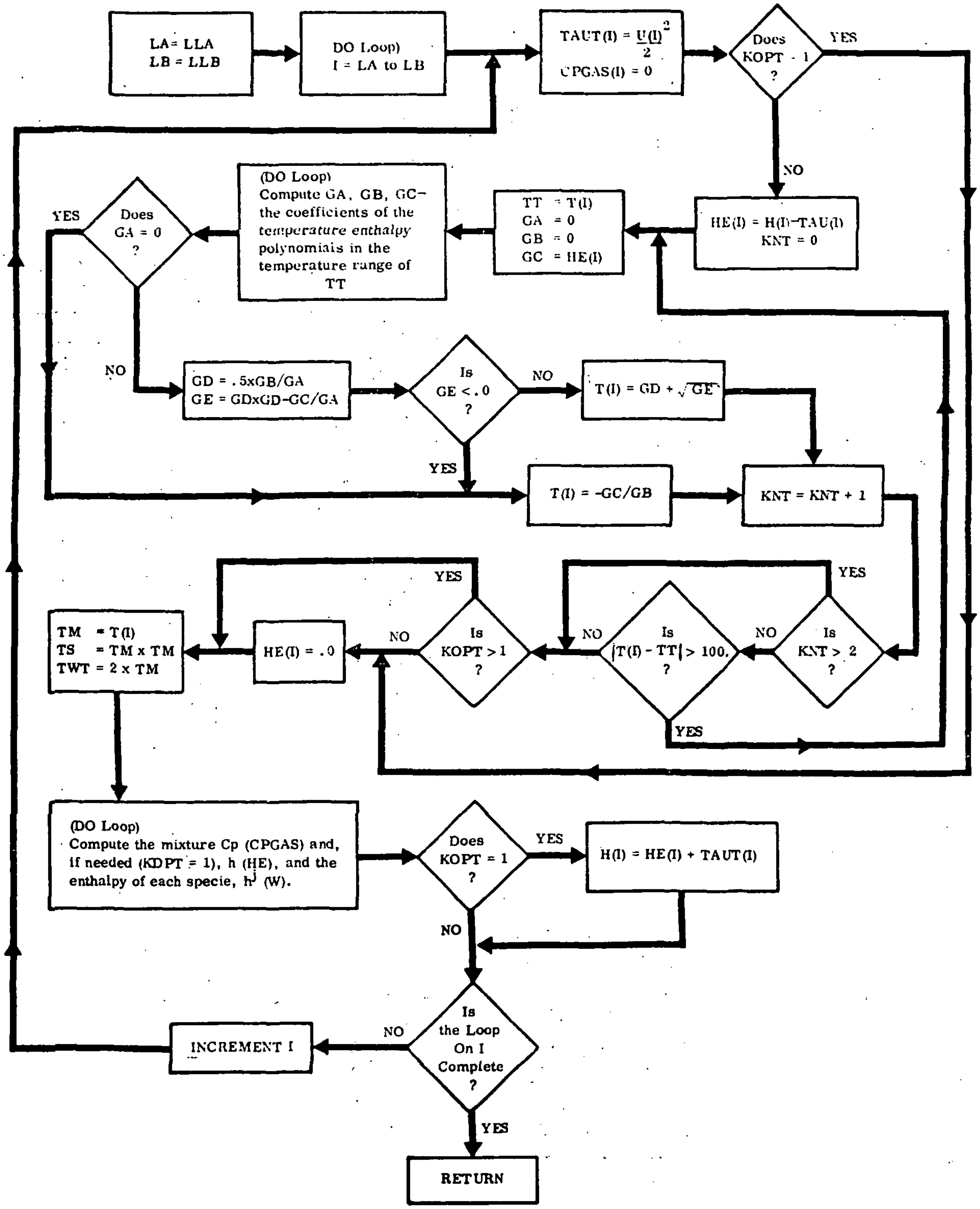
SUBROUTINE GRID


59
Is $\psi_{LR} > \psi_{MHALF}$?
YES
MINIT = MINIT + 1
MHALF = 2 x MINIT - 1
NO
Is $\psi_{LR} < \psi_{MHALF}$?
NO
1500
YES
Is $\psi_1 < \Delta\psi$?
NO
YES
For $\psi_1 > \Delta\psi$, test on difference in properties at lowest two grid points to see if a new point should be added beneath the grid. Properties examined are velocity, U, total enthalpy, H, and specie mass fractions, YSPEC (or element mass fractions, ALPHA, for equilibrium chemistry option). Any one failure of the test adds the grid point. The generalized test is:
Is $\frac{|F_2 - F_1|}{F_1} \leq .001$?
NO
YES
(DO Loop)
Transfer information for TELAP(I), H(I), U(I), T(I), and ALPHA(K, I), or YSPEC(J, I), HHLD(I), IDA(I), THLD(I), and FIX(I), one grid point upward in order to add a new point beneath the existing grid.
MPSI = MPSI + 1
NPSI = MPSI - 1
LR = LR + 1
(DO Loop)
All $\psi_i = \psi_i - \Delta\psi$
Is the wall still beyond the computation grid ?
YES
999
NO
2000

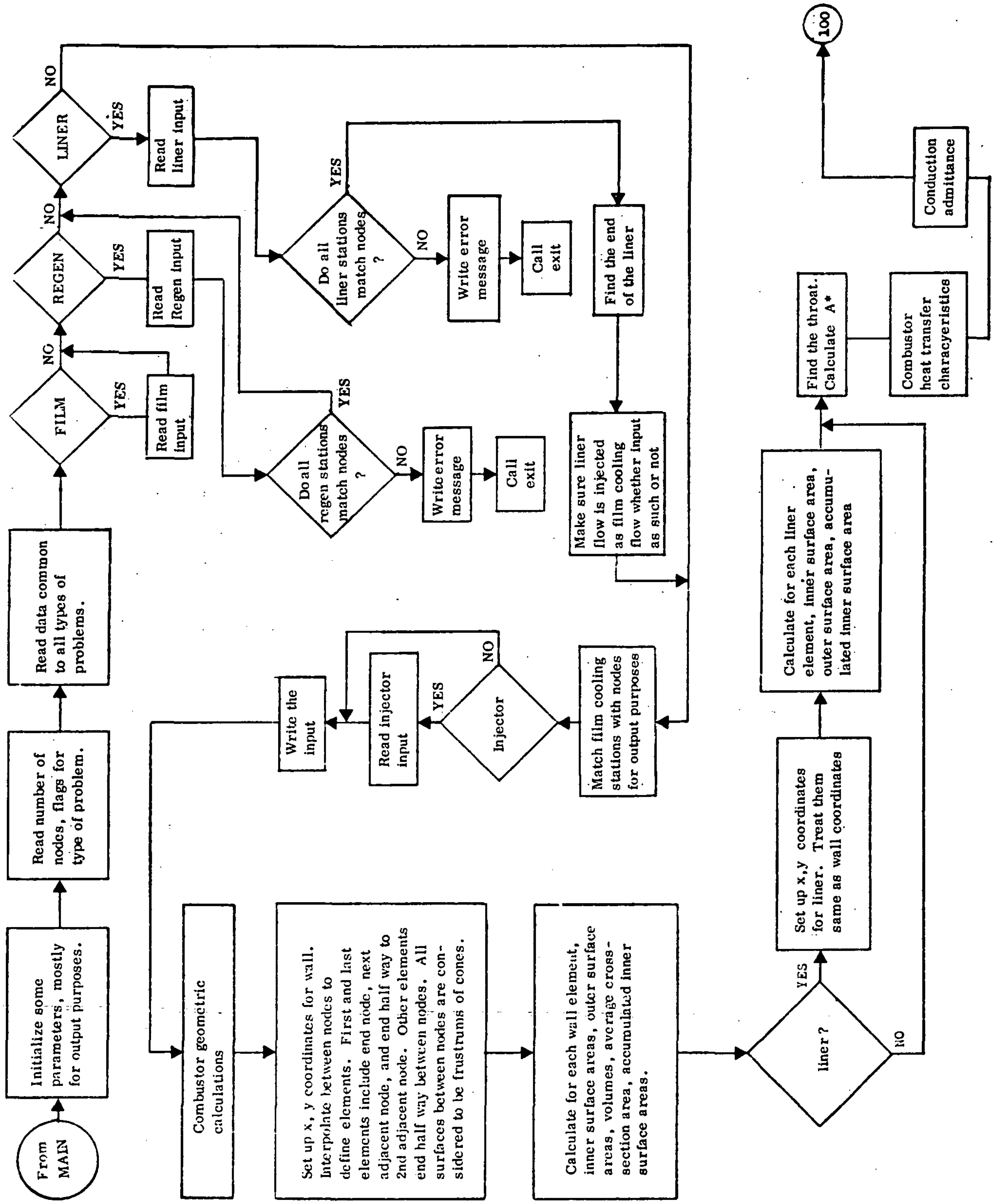
SUBROUTINE GRID (Cont.)


999
For an axial station where the wall is still above the computation grid, test on velocity, total enthalpy, and either specie mass fractions, or element mass fractions to see if another grid should be added to the top of the grid. The generated test is :
Is $\frac{|F_{NPSI} - F_{MPSI}|}{F_{MPSI}} \leq .001$?
NO
MPSI = MPSI + 1
NPSI = MPSI - 1
LR = LR + 1
MP = MPSI + 1
Duplicate information stored in previous top grid point in new top grid point. This includes values of TELAP, H, U, T, and ALPHA, or YSPEC, HHLD, IDA, THLD and FIX.
2000
YES
2000
1500
IFINIS = 0
$\Delta\psi = 2 \Delta\psi$
(DO Loop)
Halve the number of grid points used by discarding even numbered points. Reorder the information so that $F_i = F_{2i-1}$.
TELAP, U, H, T, ALPHA, or YSPEC, HHLD, IDA, THLD and FIX are preserved.
ψ_i is recomputed
LR = MINIT
Is the wall in the computational grid ?
YES
NO
MINIT = MINIT + 1
MPSI = MINIT
NPSI = MPSI - 1
MR = 2 x MPSI
(DO Loop)
Fill up abandoned grid points (to MR) with values from new top point (MPSI) of grid for TELAP, U, H, T, ALPHA, or YSPEC, HHLD, IDA, THLD, and FIX.
Recompute ψ_i .
2000
Compute MP, MQ and MR from MPSI and MF.
RETURN

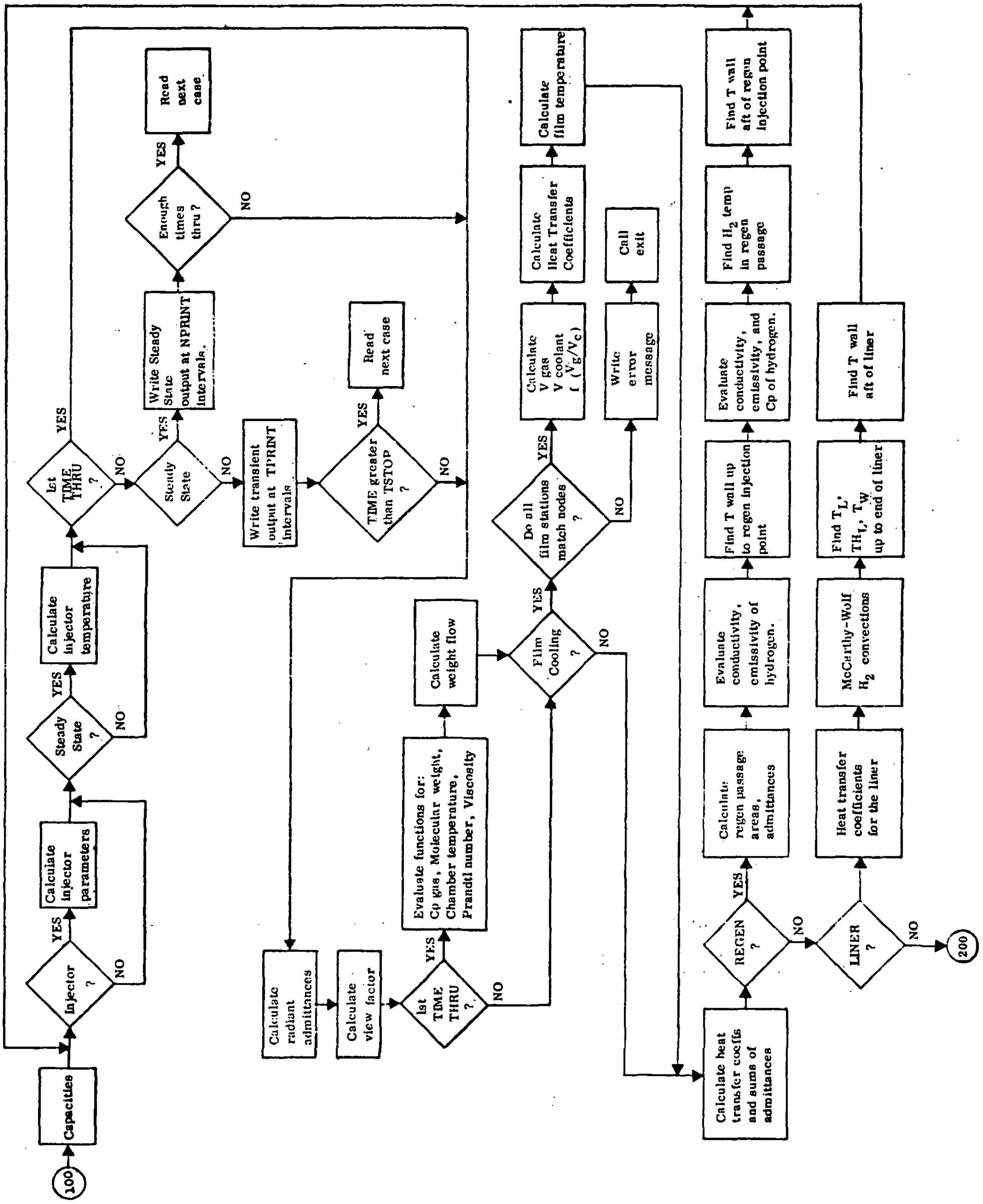
SUBROUTINE HEAT (LLA, LLB)



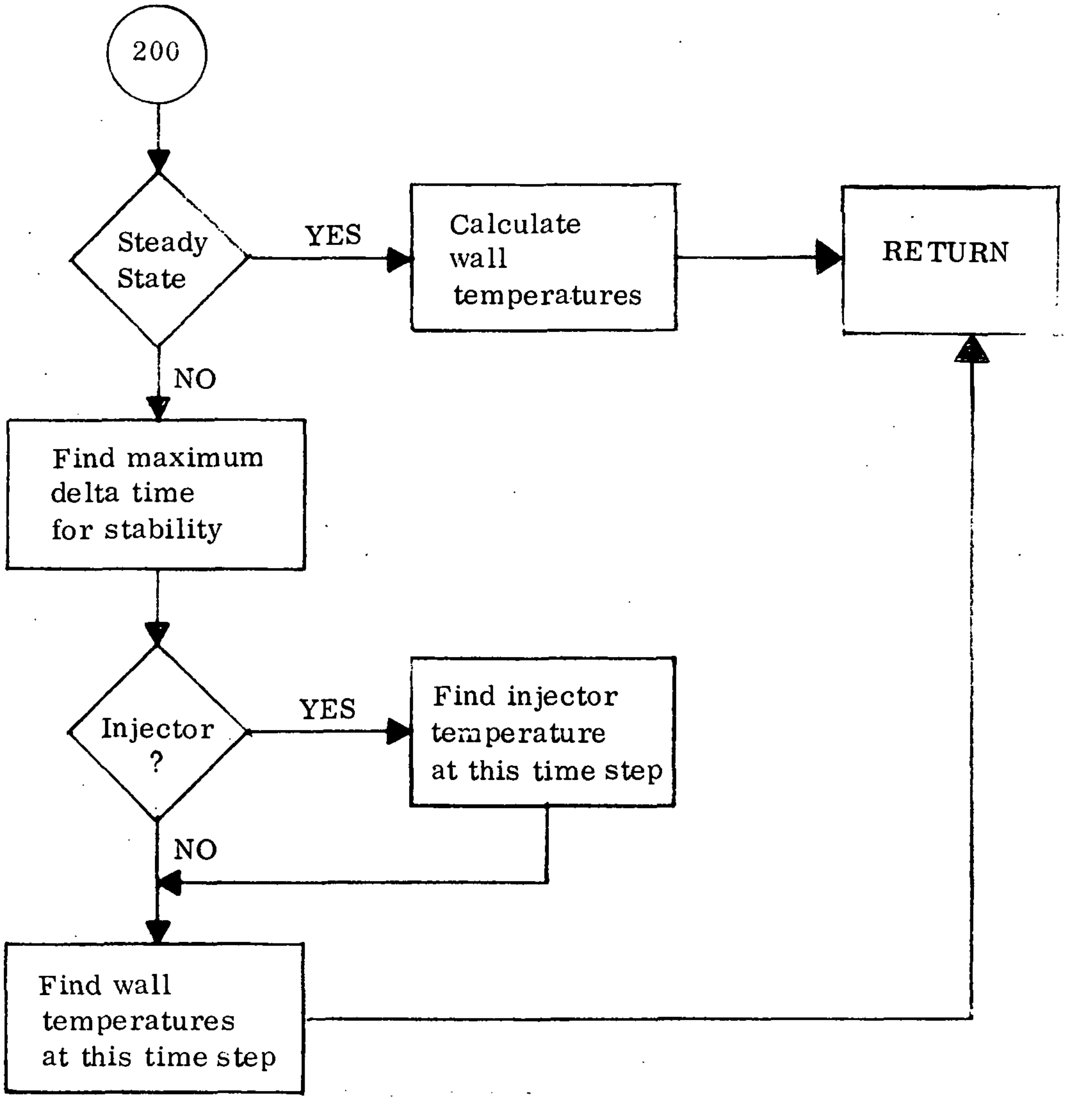
SUBROUTINE HEATT


From MAIN
Initialize some parameters, mostly for output purposes.
Read number of nodes, flags for type of problem.
Read data common to all types of problems.
FILM
YES
NO
Read film input
REGEN
YES
NO
Read Regen input
LINER
YES
NO
Read liner input
Do all regen stations match nodes ?
YES
NO
Write error message
Call exit
Do all liner stations match nodes ?
YES
NO
Write error message
Call exit
Find the end of the liner
Make sure liner flow is injected as film cooling flow whether input as such or not
Match film cooling stations with nodes for output purposes
Injector
YES
NO
Read injector input
Write the input
Combustor geometric calculations
Set up x, y coordinates for wall. Interpolate between nodes to define elements. First and last elements include end node, next adjacent node, and end half way to 2nd adjacent node. Other elements end half way between nodes. All surfaces between nodes are considered to be frustrums of cones.
Calculate for each wall element, inner surface areas, outer surface areas, volumes, average cross-section area, accumulated inner surface areas.
liner ?
YES
NO
Set up x,y coordinates for liner. Treat them same as wall coordinates
Calculate for each liner element, inner surface area, outer surface area, accumulated inner surface area
Find the throat. Calculate A*
Combustor heat transfer characyeristics
Conduction admittance
100

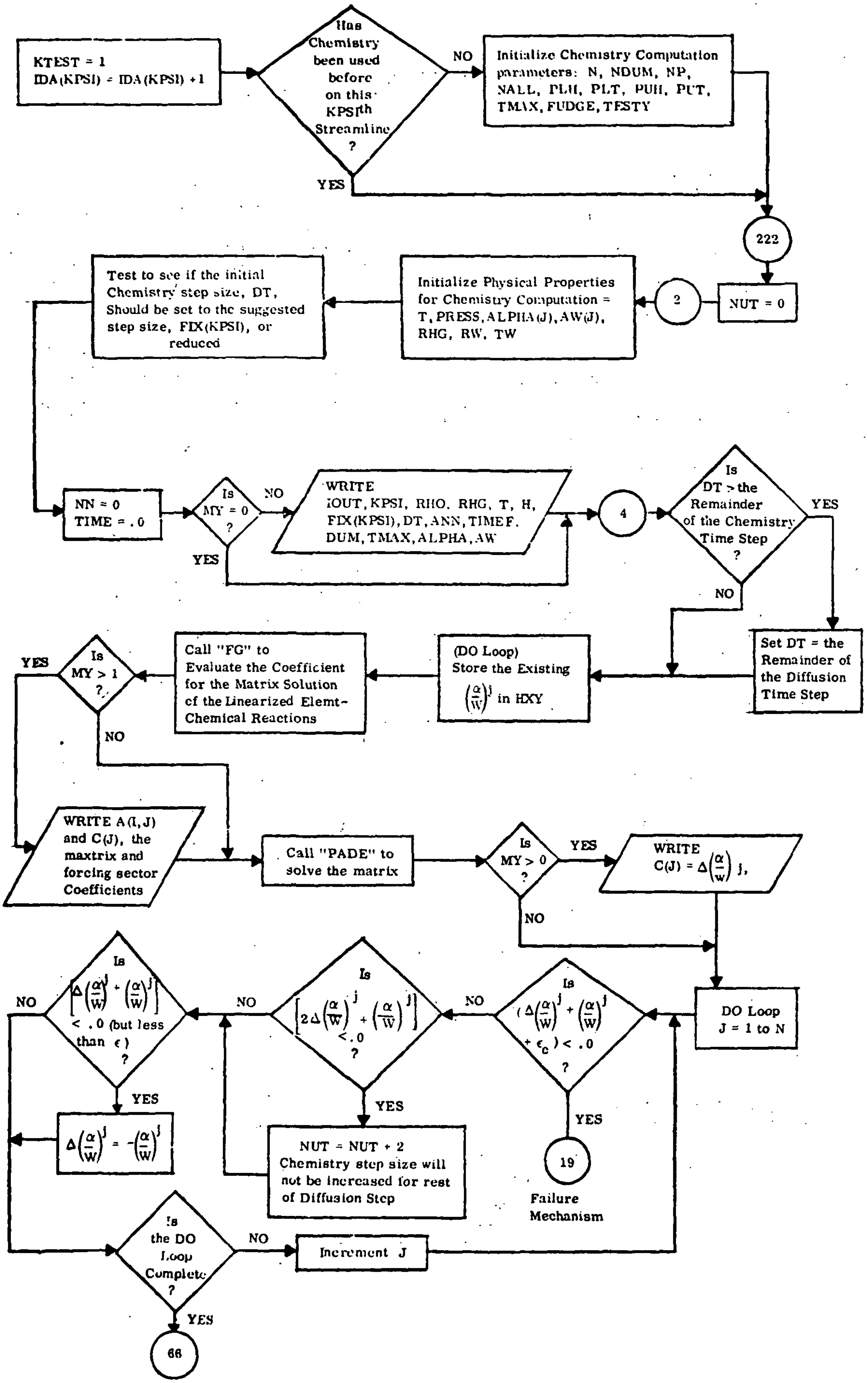
SUBROUTINE HEATT (Cont.)


100
Capacities
Injector ?
YES
NO
Calculate injector parameters
Steady State ?
YES
NO
Calculate injector temperature
1st TIME THRU ?
YES
NO
Steady State
YES
NO
Write Steady State output at NPRINT intervals.
Enough times thru ?
YES
NO
Read next case
Write transient output at TPRINT intervals
TIME greater than TSTOP ?
YES
NO
Read next case
Calculate radiant admittances
Calculate view factor
1st TIME THRU ?
YES
NO
Evaluate functions for: Cp gas, Molecular weight, Chamber temperature, Prandtl number, Viscosity
Calculate weight flow
Film Cooling ?
YES
NO
Do all film stations match nodes ?
YES
NO
Calculate V gas V coolant f (V_g/V_c)
Calculate Heat Transfer Coefficients
Calculate film temperature
Write error message
Call exit
Calculate heat transfer coeffs and sums of admittances
REGEN ?
YES
NO
Calculate regen passage areas, admittances
Evaluate conductivity, emissivity of hydrogen.
Find T wall up to regen injection point
Evaluate conductivity, emissivity, and Cp of hydrogen.
Find H_2 temp in regen passage
Find T wall aft of regen injection point
LINER ?
NO
200
Heat transfer coefficients for the liner
McCarthy-Wolf H_2 convections
Find T_L, TH_L, T_W up to end of liner
Find T wall aft of liner

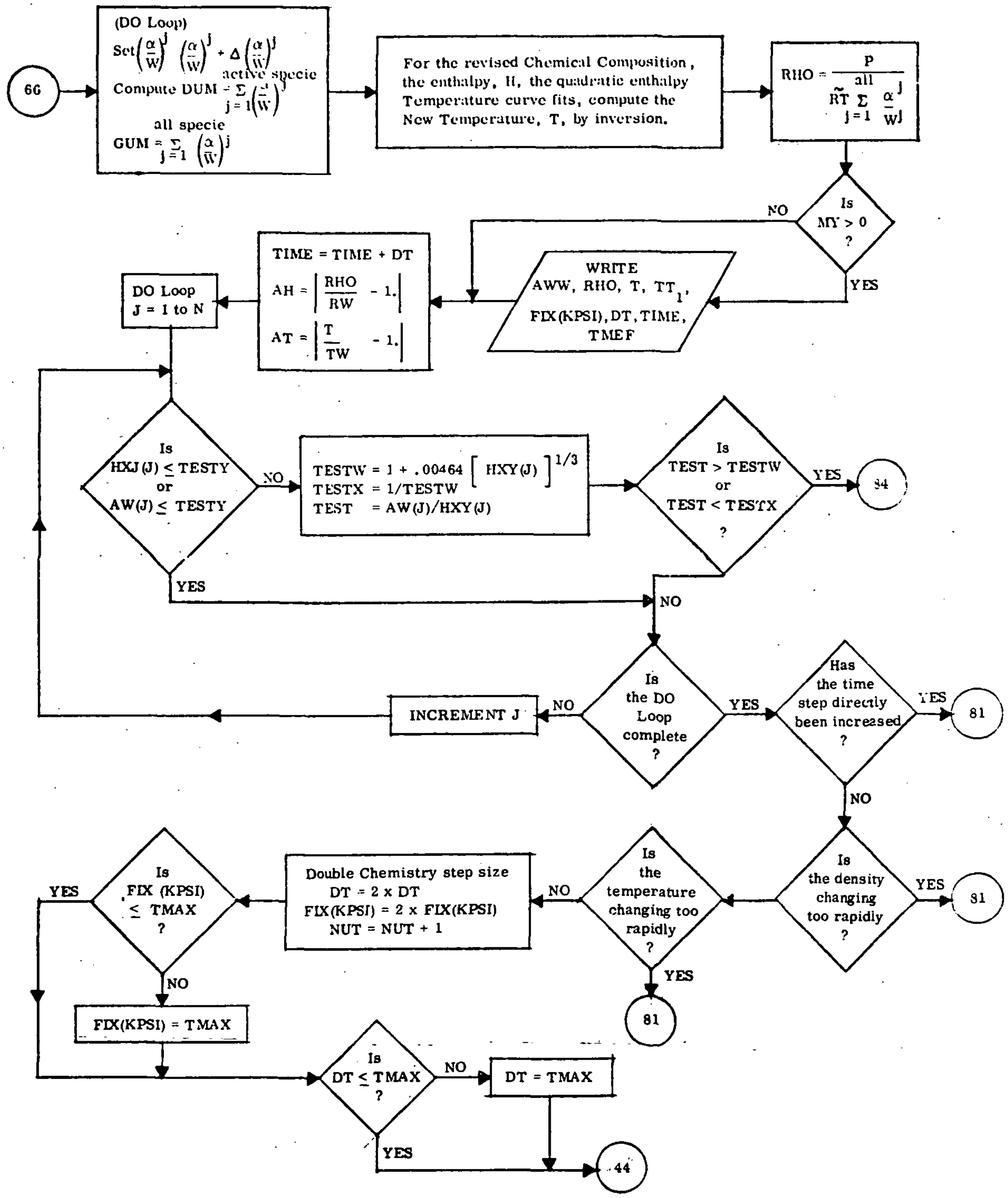
SUBROUTINE HEATT (Cont.)


200
Steady State
YES
Calculate wall temperatures
RETURN
NO
Find maximum delta time for stability
Injector ?
YES
Find injector temperature at this time step
NO
Find wall temperatures at this time step

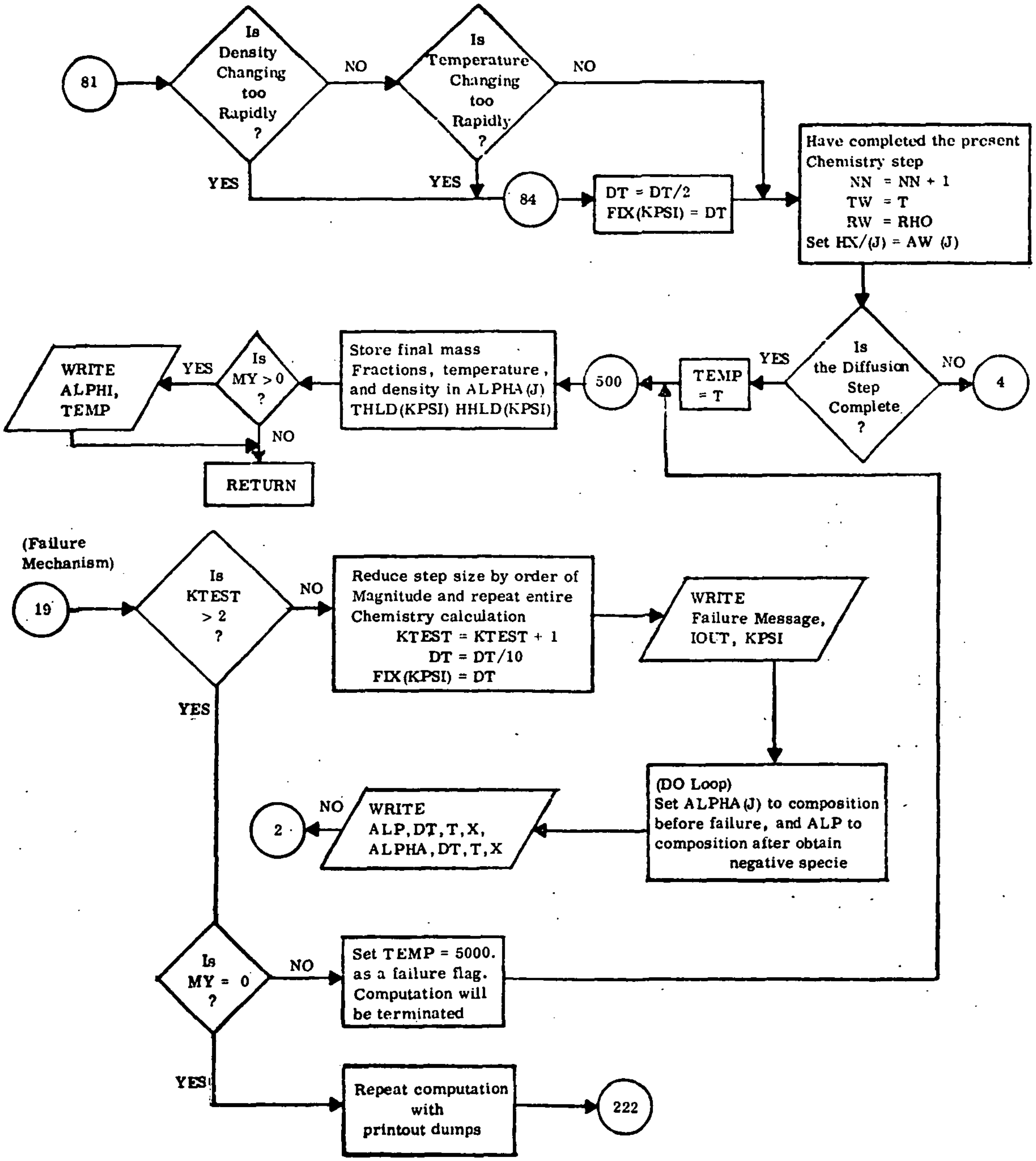
SUBROUTINE HONC


KTEST = 1
IDA(KPSI) = IDA(KPSI) + 1
Has Chemistry been used before on this KPSIth Streamline ?
NO
YES
Initialize Chemistry Computation parameters: N, NDUM, NP, NALL, PLH, PLT, PUH, PUT, TMAX, FUDGE, TESTY
222
NUT = 0
2
Initialize Physical Properties for Chemisury Computation = T, PRESS, ALPHA(J), AW(J), RHG, RW, TW
Test to see if the initial Chemistry step size, DT, Should be set to the suggested step size, FIX(KPSI), or reduced
NN = 0
TIME = .0
Is MY = 0 ?
NO
YES
WRITE IOUT, KPSI, RHO, RHG, T, H, FIX(KPSI), DT, ANN, TIMEF, DUM, TMAX, ALPHA, AW
4
Is DT > the Remainder of the Chemistry Time Step ?
YES
NO
Set DT = the Remainder of the Diffusion Time Step
(DO Loop) Store the Existing $\left(\frac{\alpha}{W}\right)^j$ in HXY
Call "FG" to Evaluate the Coefficient for the Matrix Solution of the Linearized Elemt-Chemical Reactions
Is MY > 1 ?
YES
NO
WRITE A(I, J) and C(J), the maxtrix and forcing sector Coefficients
Call "PADE" to solve the matrix
Is MY > 0 ?
YES
NO
WRITE C(J) = $\Delta\left(\frac{\alpha}{w}\right)$ j,
DO Loop J = 1 to N
Is $(\Delta\left(\frac{\alpha}{W}\right)^j + \left(\frac{\alpha}{W}\right)^j + \epsilon_c) < .0$?
YES
NO
19
Failure Mechanism
Is $\left[2\Delta\left(\frac{\alpha}{W}\right)^j + \left(\frac{\alpha}{W}\right)^j\right] < .0$?
YES
NO
NUT = NUT + 2 Chemistry step size will not be increased for rest of Diffusion Step
Is $\left[\Delta\left(\frac{\alpha}{W}\right)^j + \left(\frac{\alpha}{W}\right)^j\right] < .0$ (but less than ϵ) ?
NO
YES
$\Delta\left(\frac{\alpha}{W}\right)^j = -\left(\frac{\alpha}{W}\right)^j$
Is the DO Loop Complete ?
NO
YES
Increment J
66

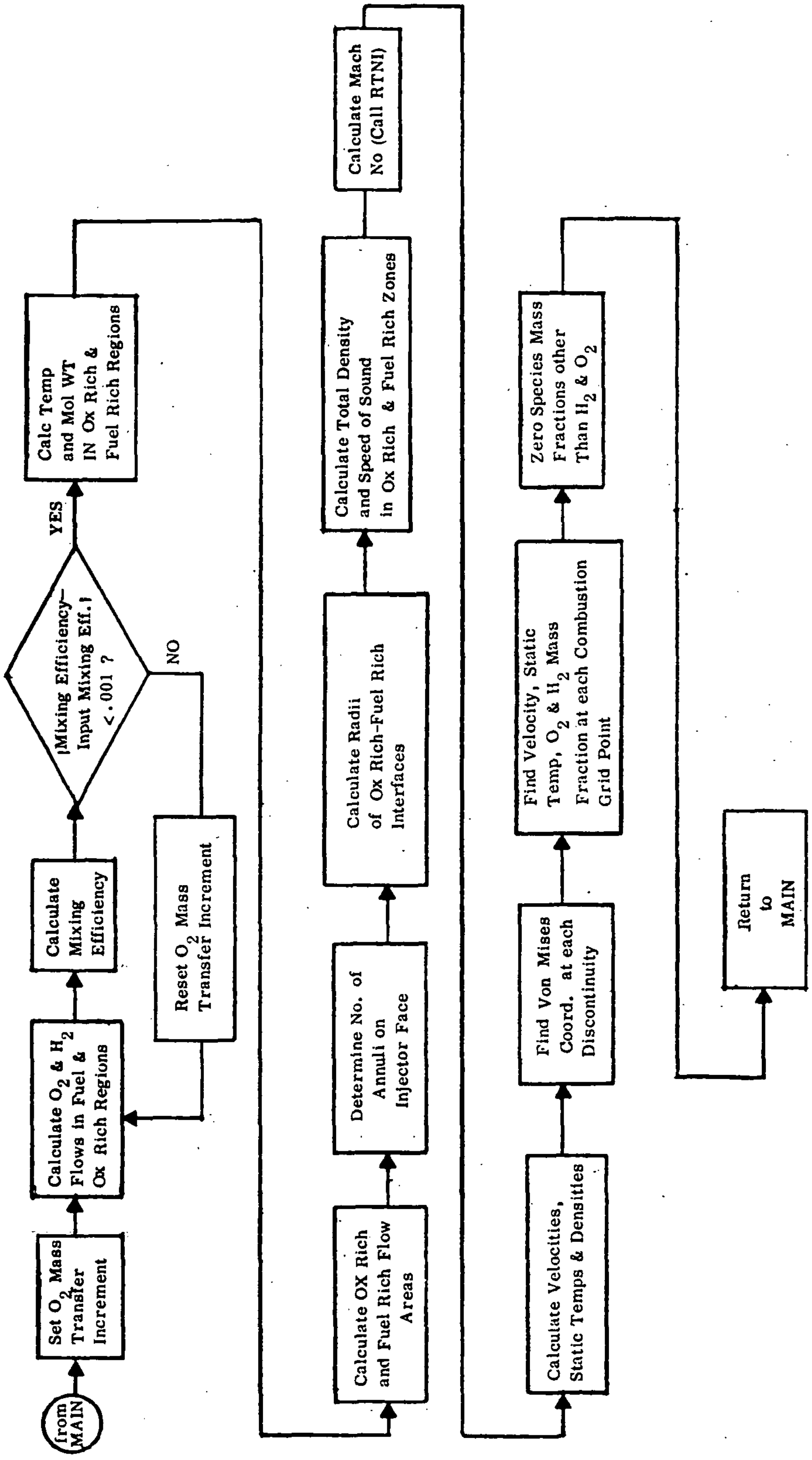
SUBROUTINE HONC (Cont.)


66
(DO Loop)
Set $\left(\frac{\alpha}{W}\right)^j \left(\frac{\alpha}{W}\right)^j + \Delta\left(\frac{\alpha}{W}\right)^j$
Compute DUM = $\sum_{j=1}^{\text{active specie}} \left(\frac{\alpha}{W}\right)^j$
GUM = $\sum_{j=1}^{\text{all specie}} \left(\frac{\alpha}{W}\right)^j$
For the revised Chemical Composition, the enthalpy, H, the quadratic enthalpy Temperature curve fits, compute the New Temperature, T, by inversion.
RHO = $\frac{P}{\tilde{R}T \sum_{j=1}^{\text{all}} \frac{\alpha}{W}^j}$
Is MY > 0 ?
NO
YES
WRITE AWW, RHO, T, TT_1, FIX(KPSI), DT, TIME, TMEF
TIME = TIME + DT
AH = $\left|\frac{RHO}{RW} - 1.\right|$
AT = $\left|\frac{T}{TW} - 1.\right|$
DO Loop J = 1 to N
Is HXJ(J) ≤ TESTY or AW(J) ≤ TESTY
NO
YES
TESTW = 1 + .00464 $[\text{HXY(J)}]^{1/3}$
TESTX = 1/TESTW
TEST = AW(J)/HXY(J)
Is TEST > TESTW or TEST < TESTX ?
YES
34
NO
Is the DO Loop complete ?
NO
INCREMENT J
YES
Has the time step directly been increased ?
YES
81
NO
Is the density changing too rapidly ?
YES
81
Is the temperature changing too rapidly ?
YES
81
NO
Double Chemistry step size
DT = 2 x DT
FIX(KPSI) = 2 x FIX(KPSI)
NUT = NUT + 1
Is FIX (KPSI) ≤ TMAX ?
YES
NO
FIX(KPSI) = TMAX
Is DT ≤ TMAX ?
NO
DT = TMAX
YES
44

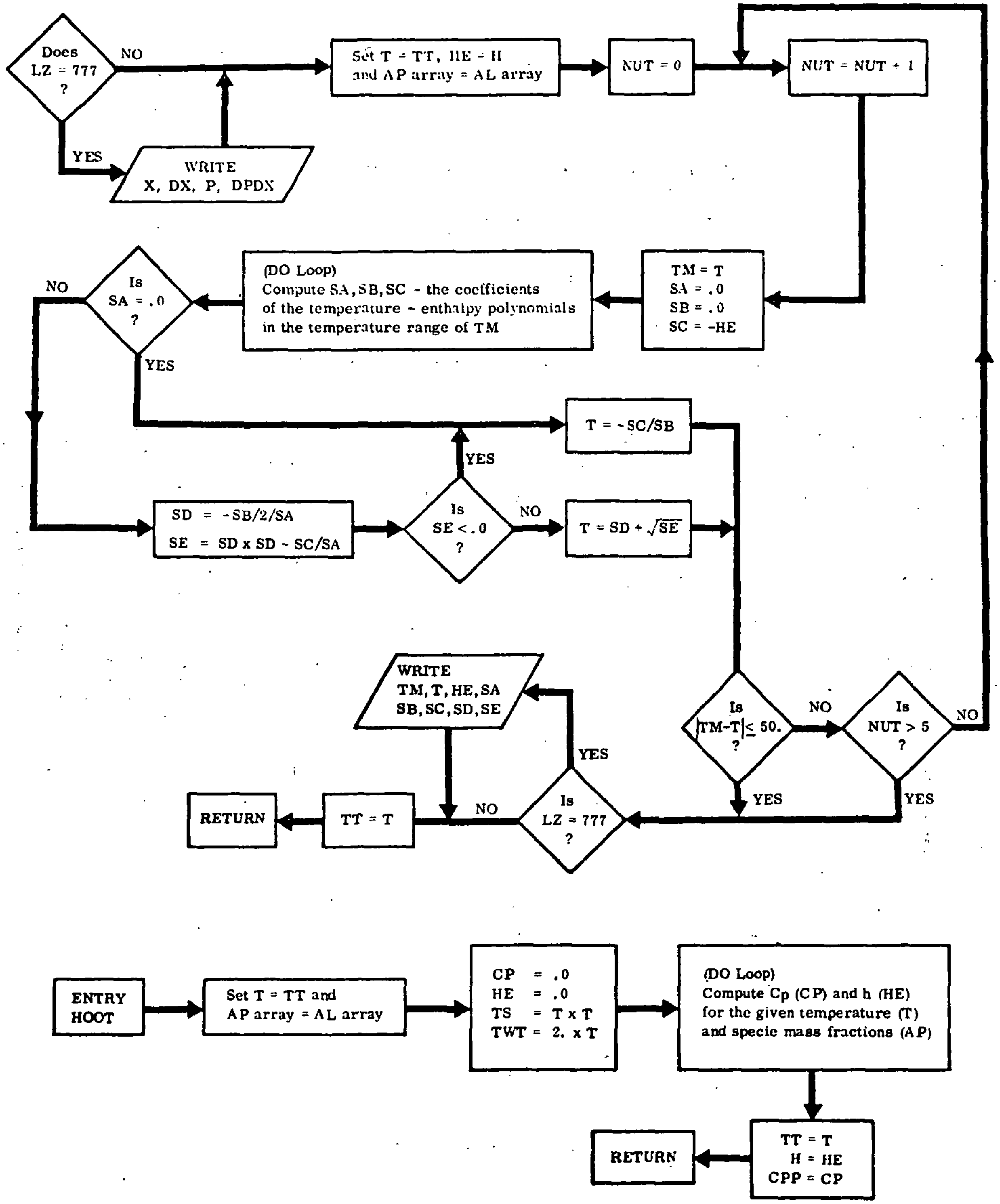
SUBROUTINE HONC (Cont.)


81
Is Density Changing too Rapidly ?
NO
YES
Is Temperature Changing too Rapidly ?
NO
YES
84
DT = DT/2
FIX(KPSI) = DT
Have completed the present Chemistry step
NN = NN + 1
TW = T
RW = RHO
Set HX/(J) = AW (J)
Is the Diffusion Step Complete ?
YES
NO
4
TEMP = T
500
Store final mass Fractions, temperature, and density in ALPHA (J) THLD(KPSI) HHLD(KPSI)
Is MY > 0 ?
YES
NO
WRITE ALPHI, TEMP
RETURN
(Failure Mechanism)
19
Is KTEST > 2 ?
NO
YES
Reduce step size by order of Magnitude and repeat entire Chemistry calculation
KTEST = KTEST + 1
DT = DT/10
FIX(KPSI) = DT
WRITE Failure Message, IOUT, KPSI
(DO Loop)
Set ALPHA (J) to composition before failure, and ALP to composition after obtain negative specie
WRITE ALP,DT,T,X, ALPHA,DT,T,X
NO
2
Is MY = 0 ?
NO
YES
Set TEMP = 5000. as a failure flag. Computation will be terminated
Repeat computation with printout dumps
222

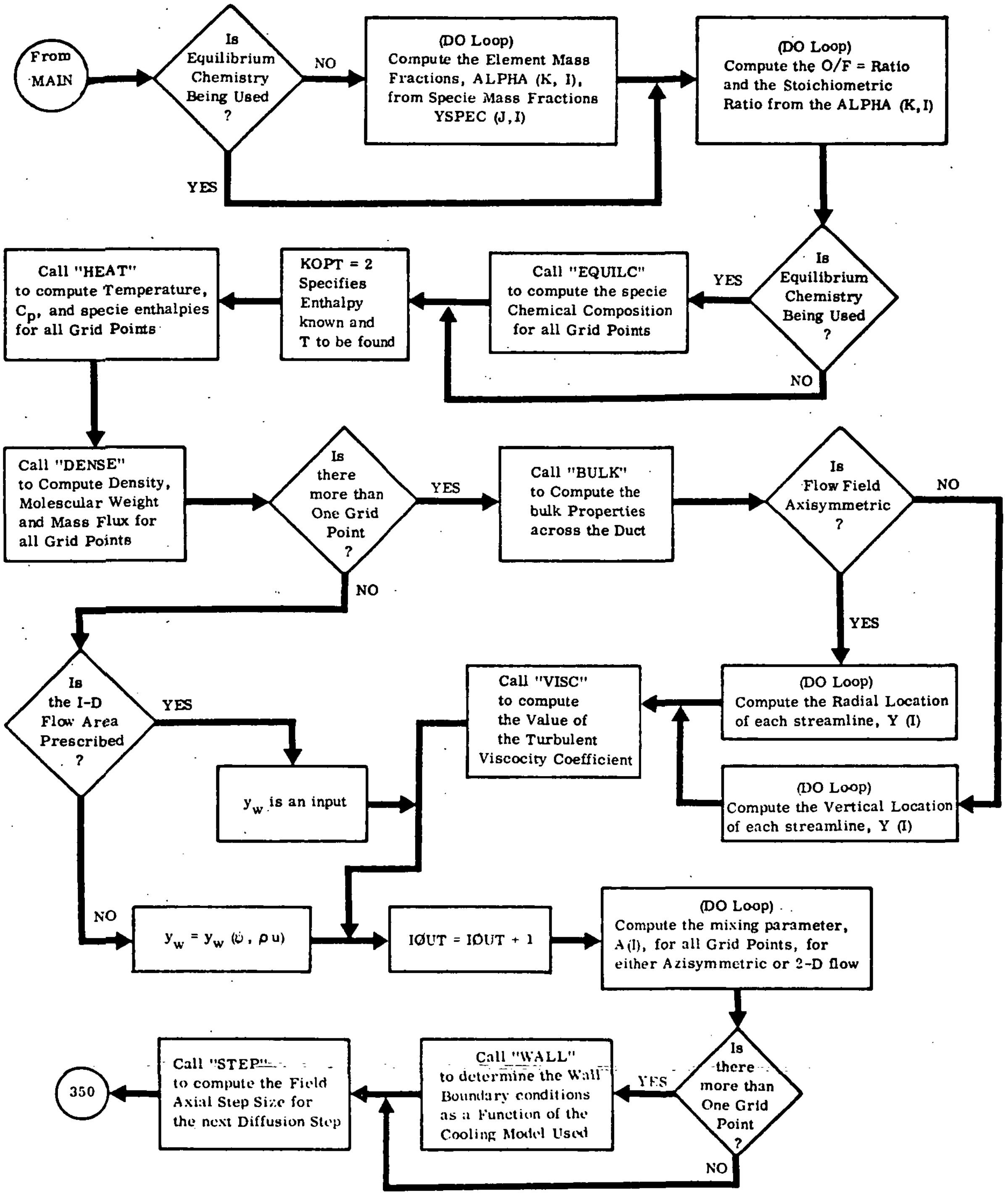
SUBROUTINE INJECT


from MAIN
Set O_2 Mass Transfer Increment
Calculate O_2 & H_2 Flows in Fuel & Ox Rich Regions
Calculate Mixing Efficiency
|Mixing Efficiency— Input Mixing Eff.| <.001 ?
NO
Reset O_2 Mass Transfer Increment
YES
Calc Temp and Mol WT IN Ox Rich & Fuel Rich Regions
Calculate OX Rich and Fuel Rich Flow Areas
Determine No. of Annuli on Injector Face
Calculate Radii of Ox Rich-Fuel Rich Interfaces
Calculate Total Density and Speed of Sound in Ox Rich & Fuel Rich Zones
Calculate Mach No (Call RTNI)
Calculate Velocities, Static Temps & Densities
Find Von Mises Coord. at each Discontinuity
Find Velocity, Static Temp, O_2 & H_2 Mass Fraction at each Combustion Grid Point
Zero Species Mass Fractions other Than H_2 & O_2
Return to MAIN

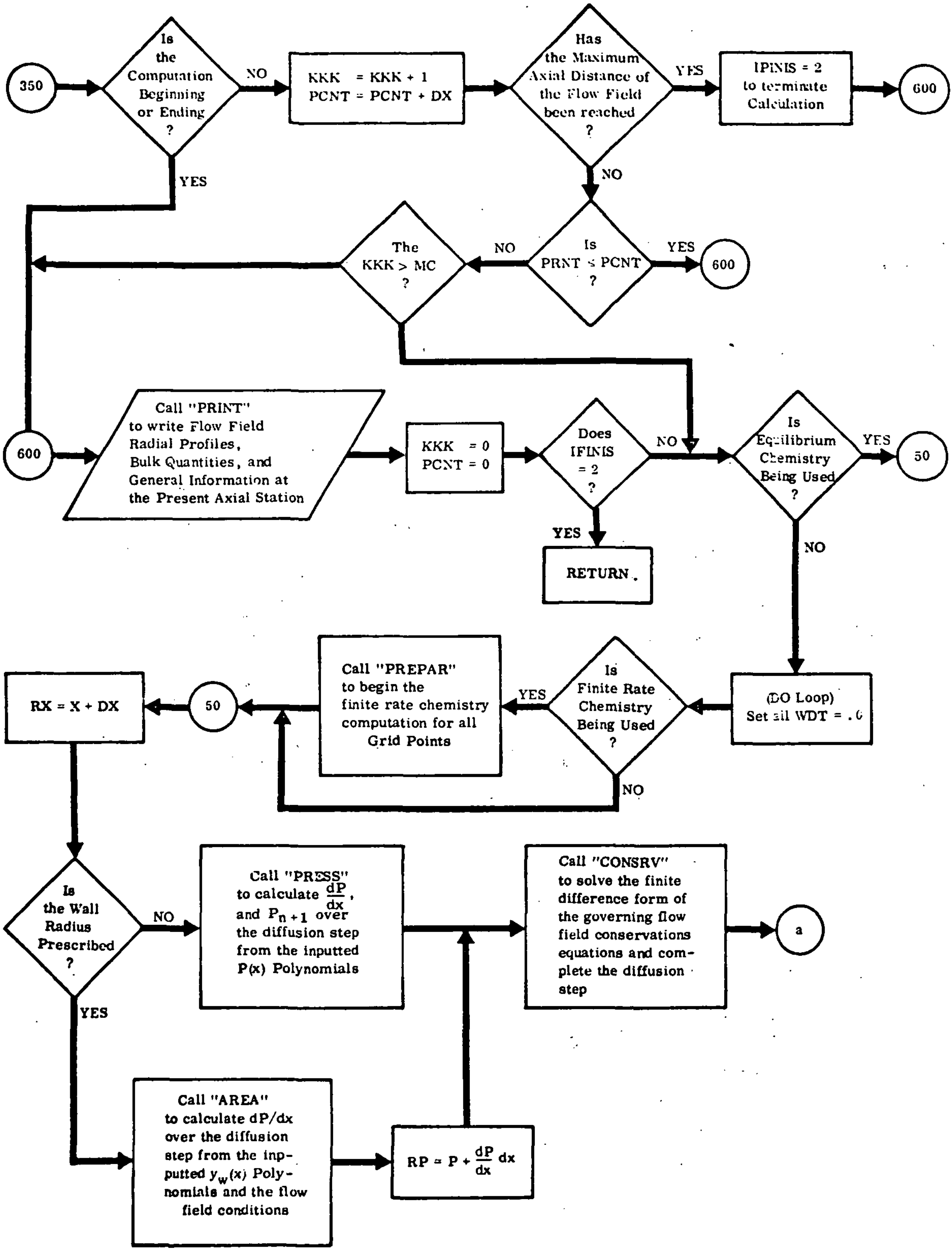
SUBROUTINE INVERT (TT, H, AL, CPP)



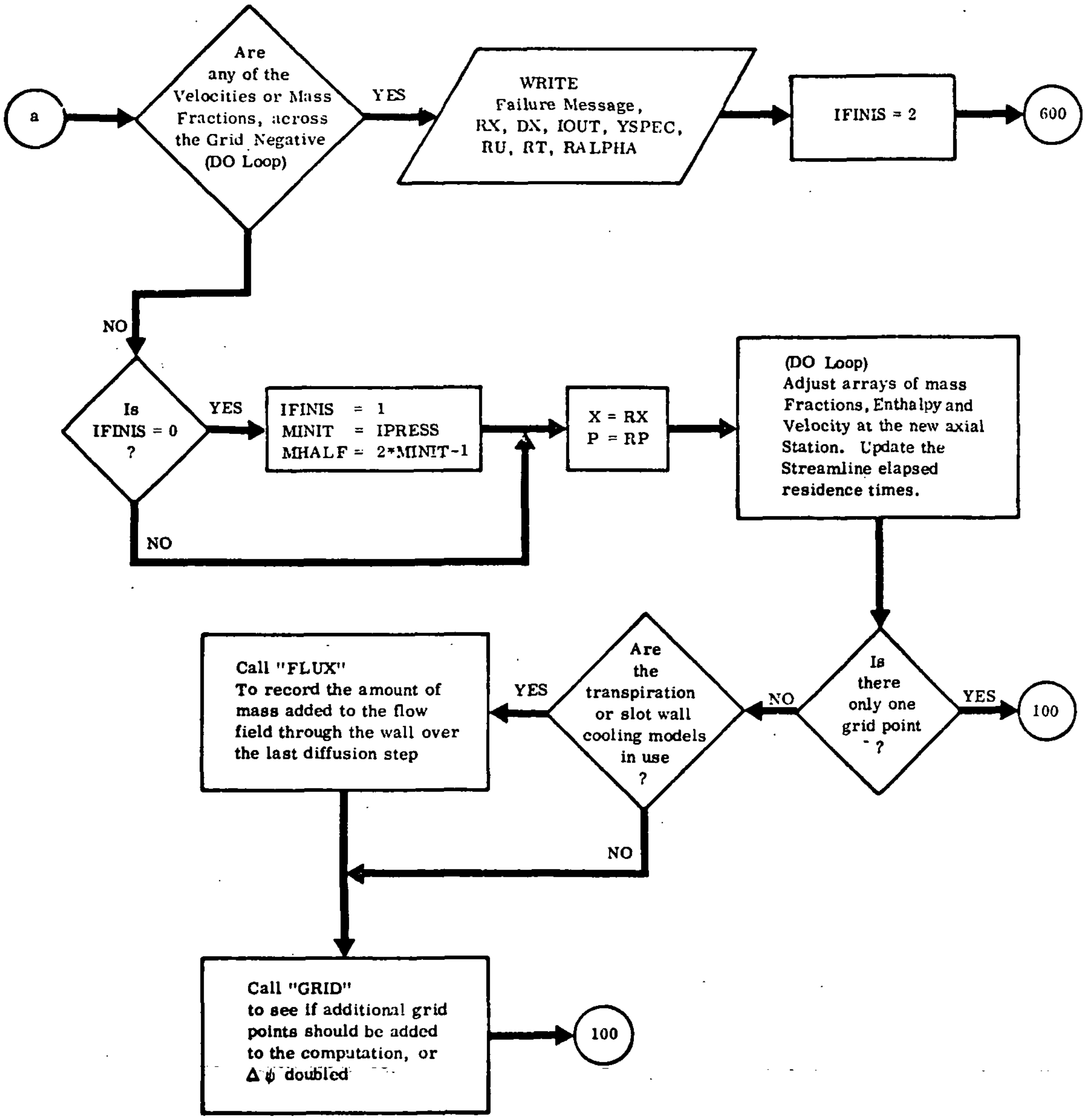
SUBROUTINE MARCH


From MAIN
Is Equilibrium Chemistry Being Used ?
NO
YES
(DO Loop) Compute the Element Mass Fractions, ALPHA (K, I), from Specie Mass Fractions YSPEC (J,I)
(DO Loop) Compute the O/F = Ratio and the Stoichiometric Ratio from the ALPHA (K,I)
Is Equilibrium Chemistry Being Used ?
YES
NO
Call "EQUILC" to compute the specie Chemical Composition for all Grid Points
KOPT = 2 Specifies Enthalpy known and T to be found
Call "HEAT" to compute Temperature, C$_p$, and specie enthalpies for all Grid Points
Call "DENSE" to Compute Density, Molescular Weight and Mass Flux for all Grid Points
Is there more than One Grid Point ?
YES
NO
Call "BULK" to Compute the bulk Properties across the Duct
Is Flow Field Axisymmetric ?
NO
YES
(DO Loop) Compute the Radial Location of each streamline, Y (I)
(DO Loop) Compute the Vertical Location of each streamline, Y (I)
Call "VISC" to compute the Value of the Turbulent Viscocity Coefficient
Is the I-D Flow Area Prescribed ?
YES
NO
y$_w$ is an input
y$_w$ = y$_w$ (ψ, ρ u)
IØUT = IØUT + 1
(DO Loop) Compute the mixing parameter, A(I), for all Grid Points, for either Azisymmetric or 2-D flow
Is there more than One Grid Point ?
YES
NO
Call "WALL" to determine the Wall Boundary conditions as a Function of the Cooling Model Used
Call "STEP" to compute the Field Axial Step Size for the next Diffusion Step
350

SUBROUTINE MARCH (Cont.)


350
Is the Computation Beginning or Ending ?
NO
KKK = KKK + 1
PCNT = PCNT + DX
Has the Maximum Axial Distance of the Flow Field been reached ?
YES
IFINIS = 2 to terminate Calculation
600
YES
NO
Is PRNT ≤ PCNT ?
YES
600
NO
The KKK > MC ?
600
Call "PRINT" to write Flow Field Radial Profiles, Bulk Quantities, and General Information at the Present Axial Station
KKK = 0
PCNT = 0
Does IFINIS = 2 ?
NO
YES
RETURN.
Is Equilibrium Chemistry Being Used ?
YES
50
NO
(DO Loop) Set all WDT = .0
Is Finite Rate Chemistry Being Used ?
YES
NO
Call "PREPAR" to begin the finite rate chemistry computation for all Grid Points
50
RX = X + DX
Is the Wall Radius Prescribed ?
NO
YES
Call "PRESS" to calculate dP/dx, and P_n+1 over the diffusion step from the inputted P(x) Polynomials
Call "CONSRV" to solve the finite difference form of the governing flow field conservations equations and complete the diffusion step
a
Call "AREA" to calculate dP/dx over the diffusion step from the inputted y_w(x) Polynomials and the flow field conditions
RP = P + dP/dx dx

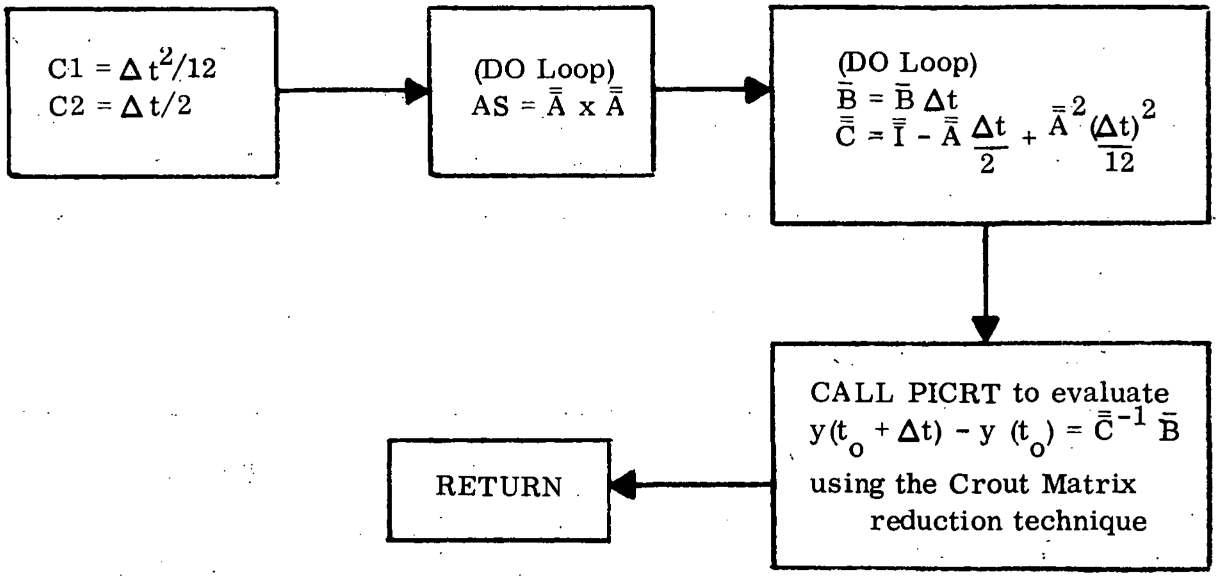
SUBROUTINE MARCH (Cont.)


a
Are any of the Velocities or Mass Fractions, across the Grid Negative (DO Loop)
YES
WRITE Failure Message, RX, DX, IOUT, YSPEC, RU, RT, RALPHA
IFINIS = 2
600
NO
Is IFINIS = 0 ?
YES
IFINIS = 1
MINIT = IPRESS
MHALF = 2*MINIT-1
NO
X = RX
P = RP
(DO Loop)
Adjust arrays of mass Fractions, Enthalpy and Velocity at the new axial Station. Update the Streamline elapsed residence times.
Is there only one grid point ?
YES
100
NO
Are the transpiration or slot wall cooling models in use ?
YES
Call "FLUX"
To record the amount of mass added to the flow field through the wall over the last diffusion step
NO
Call "GRID"
to see if additional grid points should be added to the computation, or
Δ φ doubled
100

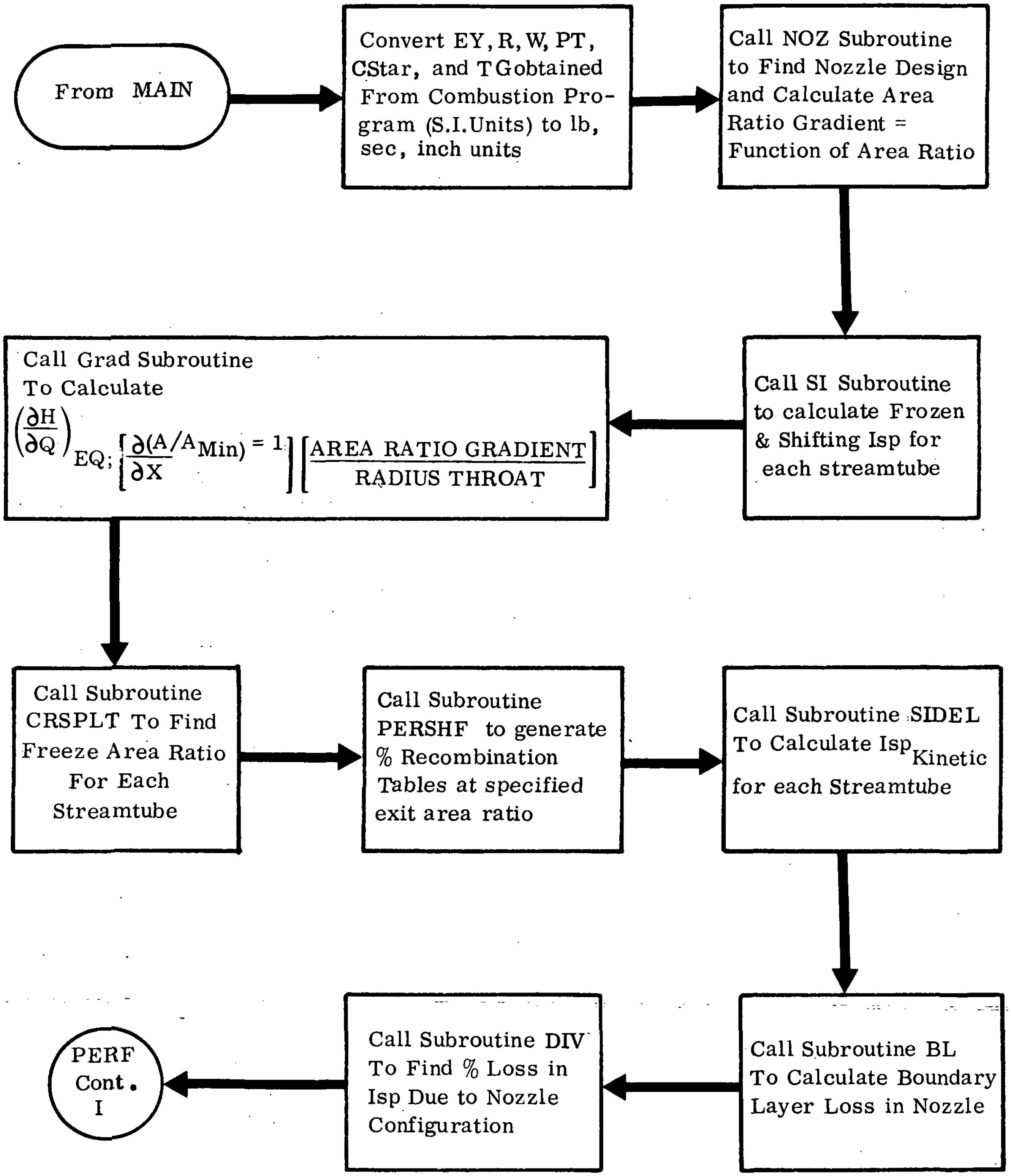
SUBROUTINE PADE

This subroutine prepares for the solution of the integrated form of the chemical kinetic, linearized, ordinary differential equations:

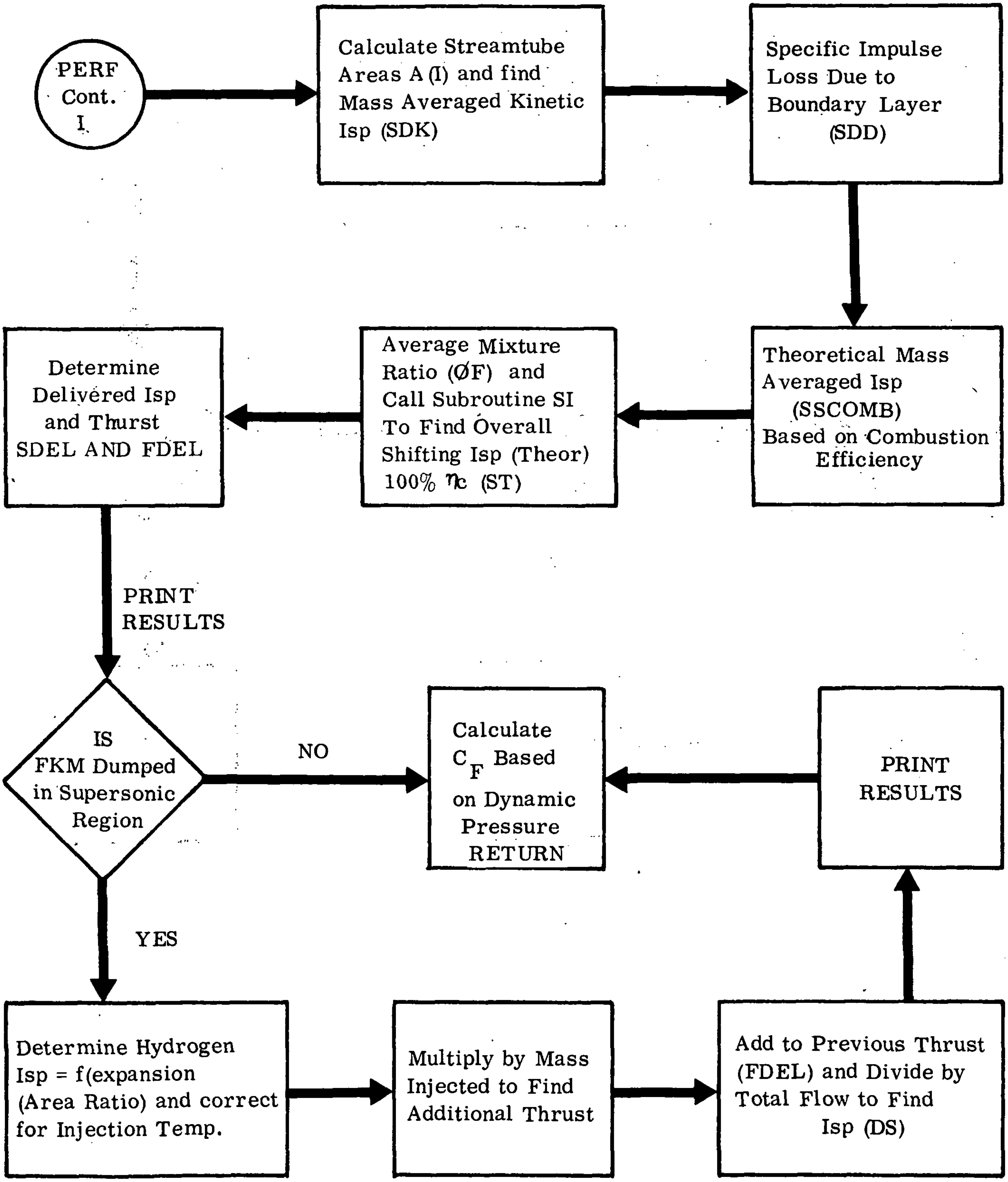
$$y\,(t_o + \Delta t) - y(t_o) = \left[\bar{\bar{I}} - \bar{\bar{A}}\,\frac{\Delta t}{2} + \bar{\bar{A}}^2\,\frac{(\Delta t)^2}{12}\right]^{-1} \Delta t\,\bar{B}$$


$C1 = \Delta t^2/12$
$C2 = \Delta t/2$
(DO Loop)
$AS = \bar{\bar{A}} \times \bar{\bar{A}}$
(DO Loop)
$\bar{B} = \bar{B}\,\Delta t$
$\bar{\bar{C}} = \bar{\bar{I}} - \bar{\bar{A}}\,\frac{\Delta t}{2} + \bar{\bar{A}}^2\,\frac{(\Delta t)^2}{12}$
CALL PICRT to evaluate
$y(t_o + \Delta t) - y\,(t_o) = \bar{\bar{C}}^{-1}\,\bar{B}$
using the Crout Matrix reduction technique
RETURN

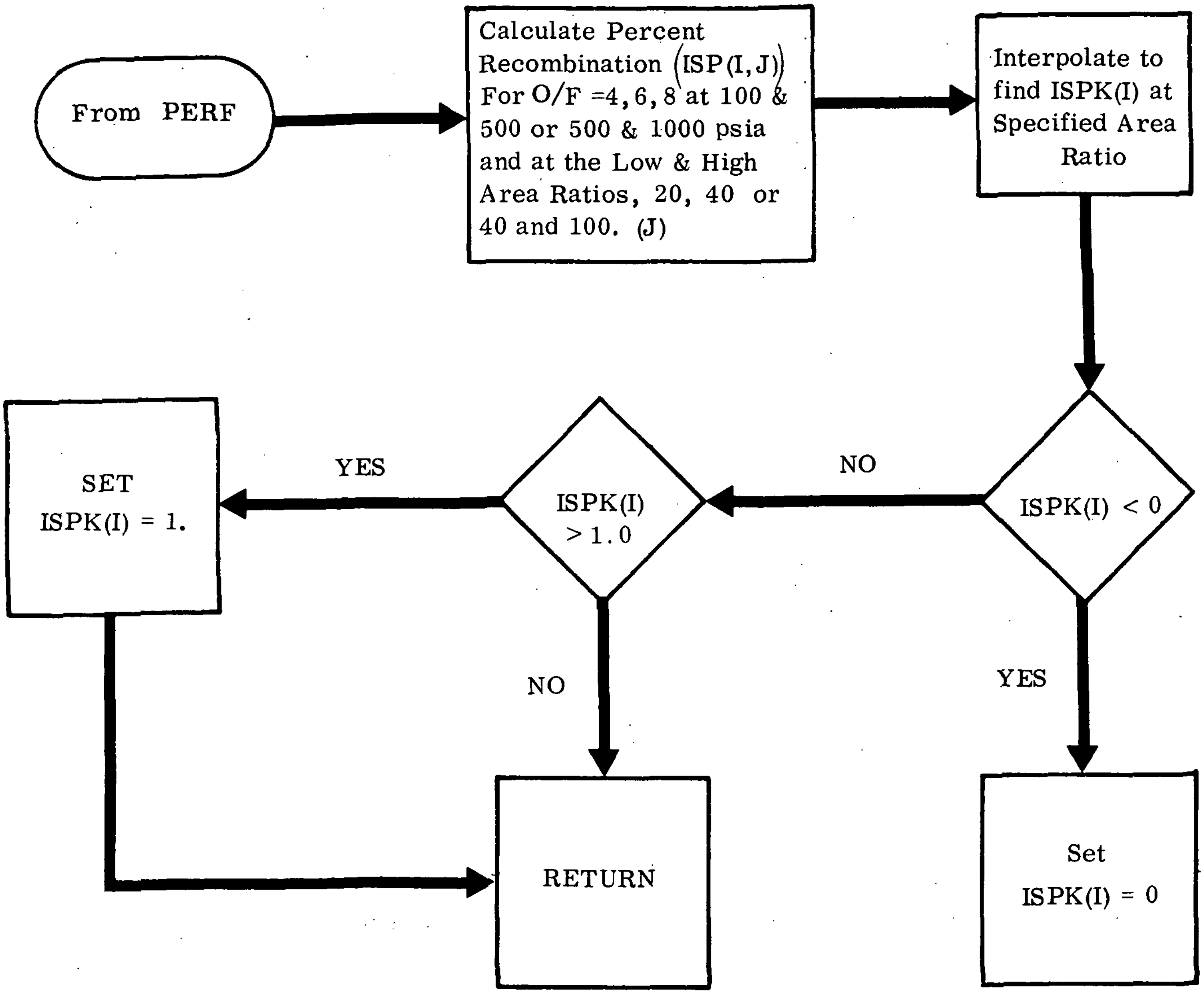
SUBROUTINE PERF


From MAIN
Convert EY, R, W, PT, CStar, and TGobtained From Combustion Program (S.I.Units) to lb, sec, inch units
Call NOZ Subroutine to Find Nozzle Design and Calculate Area Ratio Gradient = Function of Area Ratio
Call SI Subroutine to calculate Frozen & Shifting Isp for each streamtube
Call Grad Subroutine To Calculate
$\left(\frac{\partial H}{\partial Q}\right)_{EQ}; \left[\frac{\partial(A/A_{Min}) = 1}{\partial X}\right] \left[\frac{\text{AREA RATIO GRADIENT}}{\text{RADIUS THROAT}}\right]$
Call Subroutine CRSPLT To Find Freeze Area Ratio For Each Streamtube
Call Subroutine PERSHF to generate % Recombination Tables at specified exit area ratio
Call Subroutine SIDEL To Calculate $Isp_{Kinetic}$ for each Streamtube
Call Subroutine BL To Calculate Boundary Layer Loss in Nozzle
Call Subroutine DIV To Find % Loss in Isp Due to Nozzle Configuration
PERF Cont. I

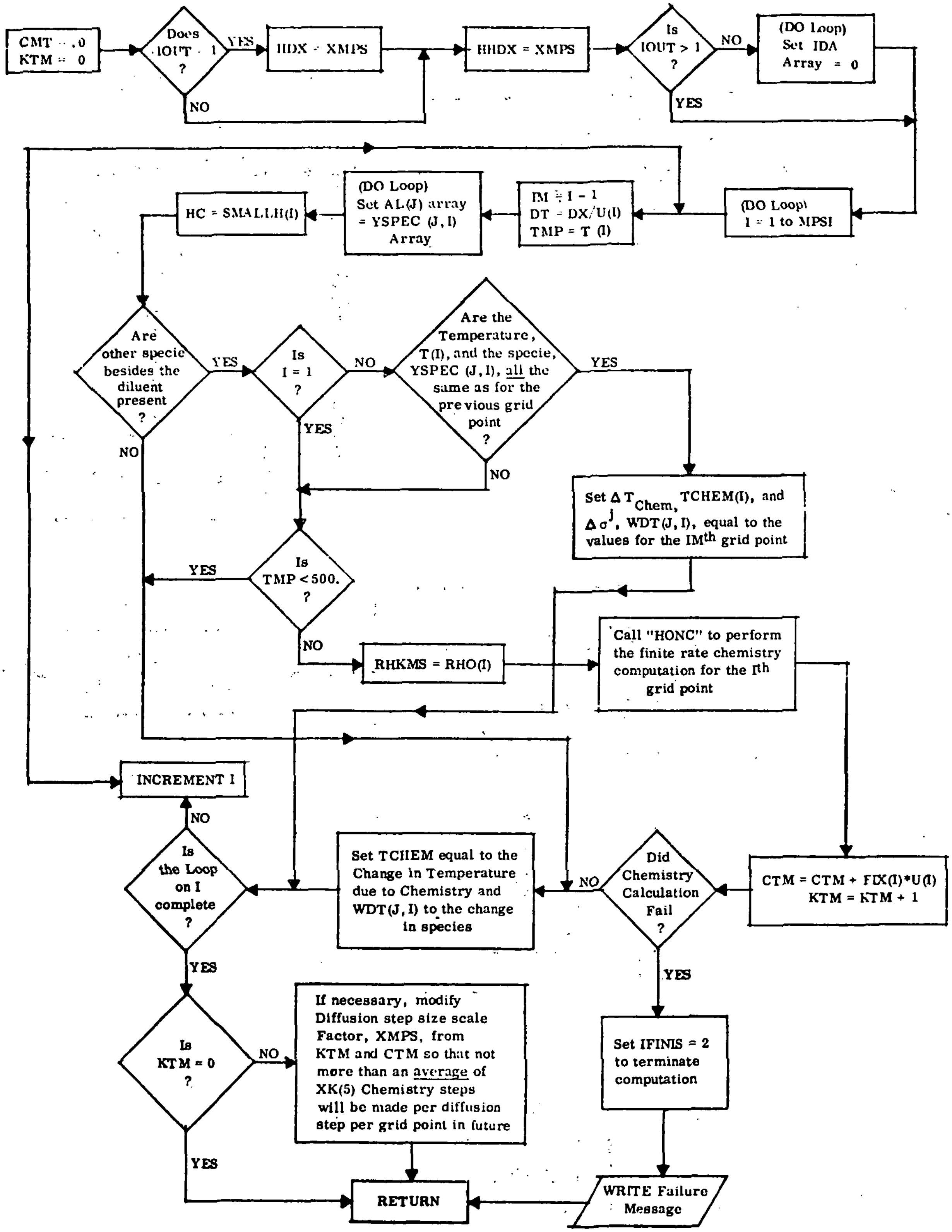
SUBROUTINE PERF (Cont.)


PERF Cont. I
Calculate Streamtube Areas A (I) and find Mass Averaged Kinetic Isp (SDK)
Specific Impulse Loss Due to Boundary Layer (SDD)
Theoretical Mass Averaged Isp (SSCOMB) Based on Combustion Efficiency
Average Mixture Ratio (ØF) and Call Subroutine SI To Find Overall Shifting Isp (Theor) 100% ηc (ST)
Determine Delivered Isp and Thurst SDEL AND FDEL
PRINT RESULTS
IS FKM Dumped in Supersonic Region
NO
Calculate C F Based on Dynamic Pressure RETURN
PRINT RESULTS
YES
Determine Hydrogen Isp = f(expansion (Area Ratio) and correct for Injection Temp.
Multiply by Mass Injected to Find Additional Thrust
Add to Previous Thrust (FDEL) and Divide by Total Flow to Find Isp (DS)

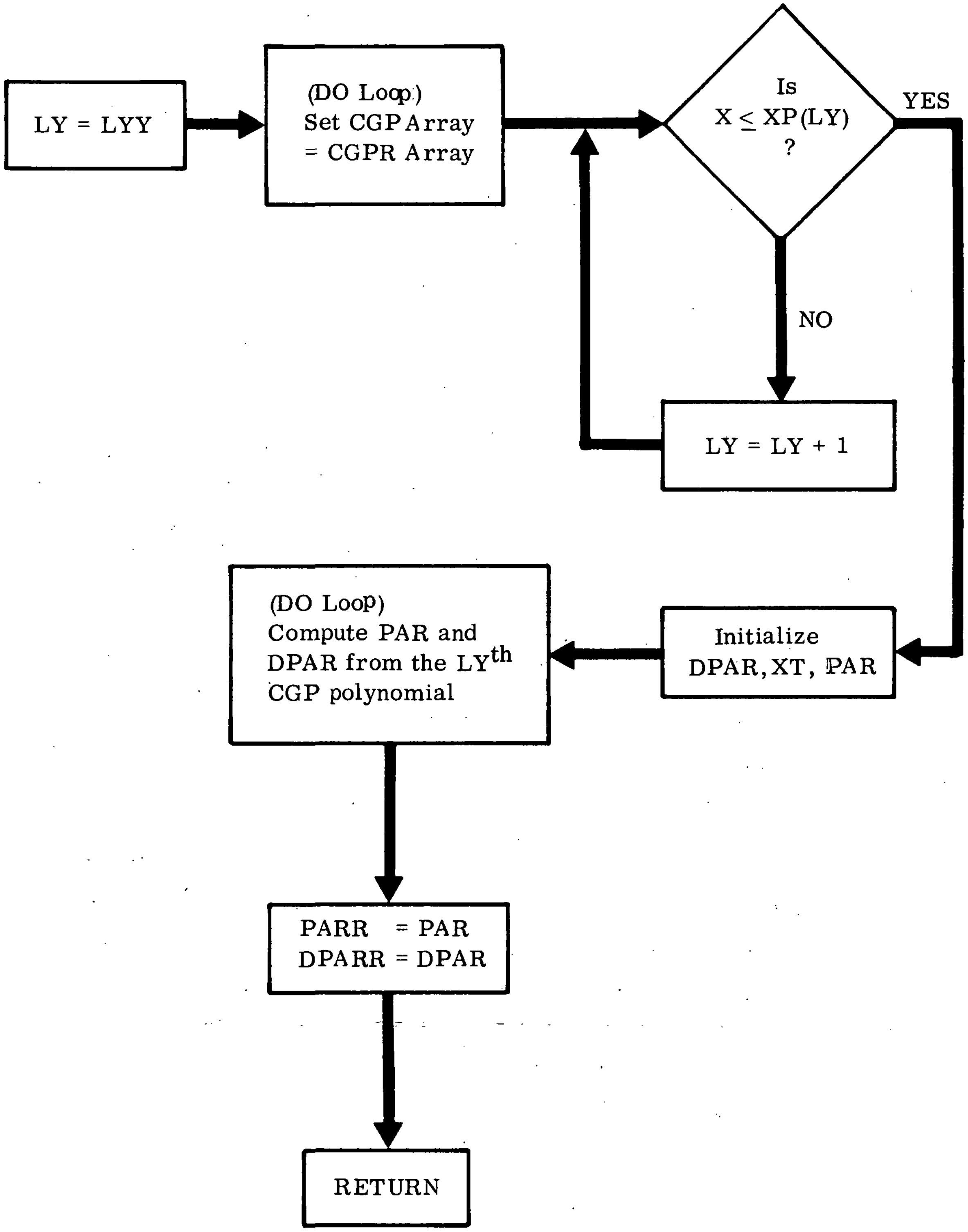
SUBROUTINE PERSHF


From PERF
Calculate Percent Recombination (ISP(I,J)) For O/F =4,6,8 at 100 & 500 or 500 & 1000 psia and at the Low & High Area Ratios, 20, 40 or 40 and 100. (J)
Interpolate to find ISPK(I) at Specified Area Ratio
ISPK(I) < 0
NO
YES
ISPK(I) > 1.0
YES
NO
SET ISPK(I) = 1.
Set ISPK(I) = 0
RETURN

SUBROUTINE PREPAR

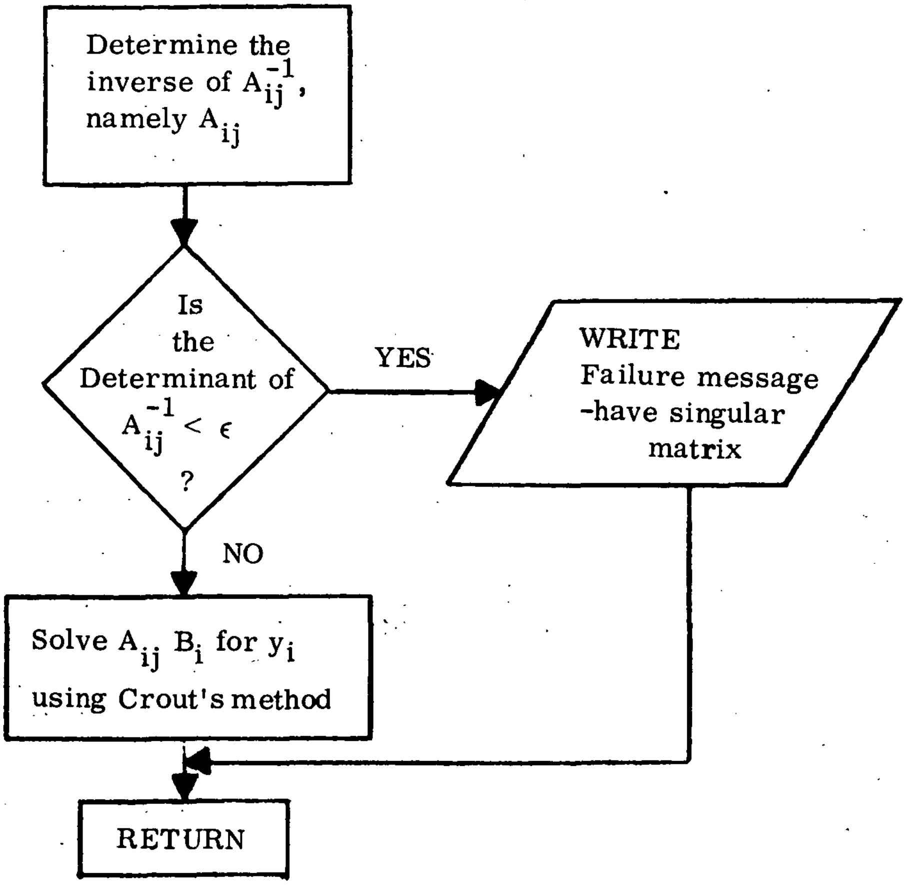

CMT = .0
KTM = 0
Does IOUT = 1 ?
YES
NO
HDX = XMPS
HHDX = XMPS
Is IOUT > 1 ?
NO
YES
(DO Loop) Set IDA Array = 0
(DO Loop) I = 1 to MPSI
IM = I - 1
DT = DX/U(I)
TMP = T (I)
(DO Loop) Set AL(J) array = YSPEC (J,I) Array
HC = SMALLH(I)
Are other specie besides the diluent present ?
YES
NO
Is I = 1 ?
NO
YES
Are the Temperature, T(I), and the specie, YSPEC (J,I), all the same as for the previous grid point ?
YES
NO
Set Δ T Chem, TCHEM(I), and Δσ^j, WDT(J,I), equal to the values for the IMth grid point
Is TMP < 500. ?
YES
NO
RHKMS = RHO(I)
Call "HONC" to perform the finite rate chemistry computation for the Ith grid point
INCREMENT I
NO
Is the Loop on I complete ?
YES
Set TCHEM equal to the Change in Temperature due to Chemistry and WDT(J,I) to the change in species
NO
Did Chemistry Calculation Fail ?
YES
CTM = CTM + FDX(I)*U(I)
KTM = KTM + 1
Is KTM = 0 ?
NO
YES
If necessary, modify Diffusion step size scale Factor, XMPS, from KTM and CTM so that not more than an average of XK(5) Chemistry steps will be made per diffusion step per grid point in future
Set IFINIS = 2 to terminate computation
RETURN
WRITE Failure Message

SUBROUTINE PRESS (PARR, DPARR, X, CGPR, XP, LYY)

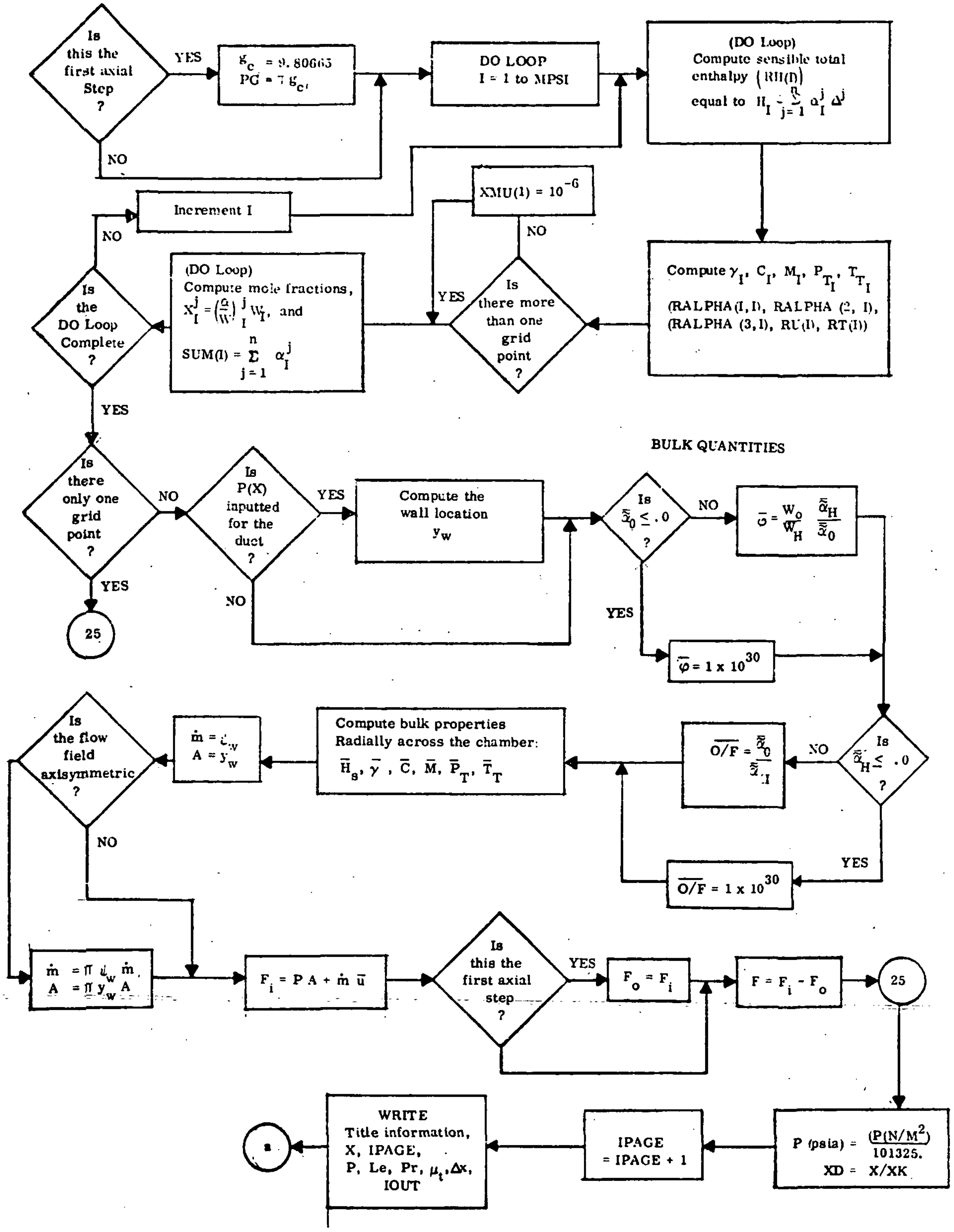


SUBROUTINE PICRT

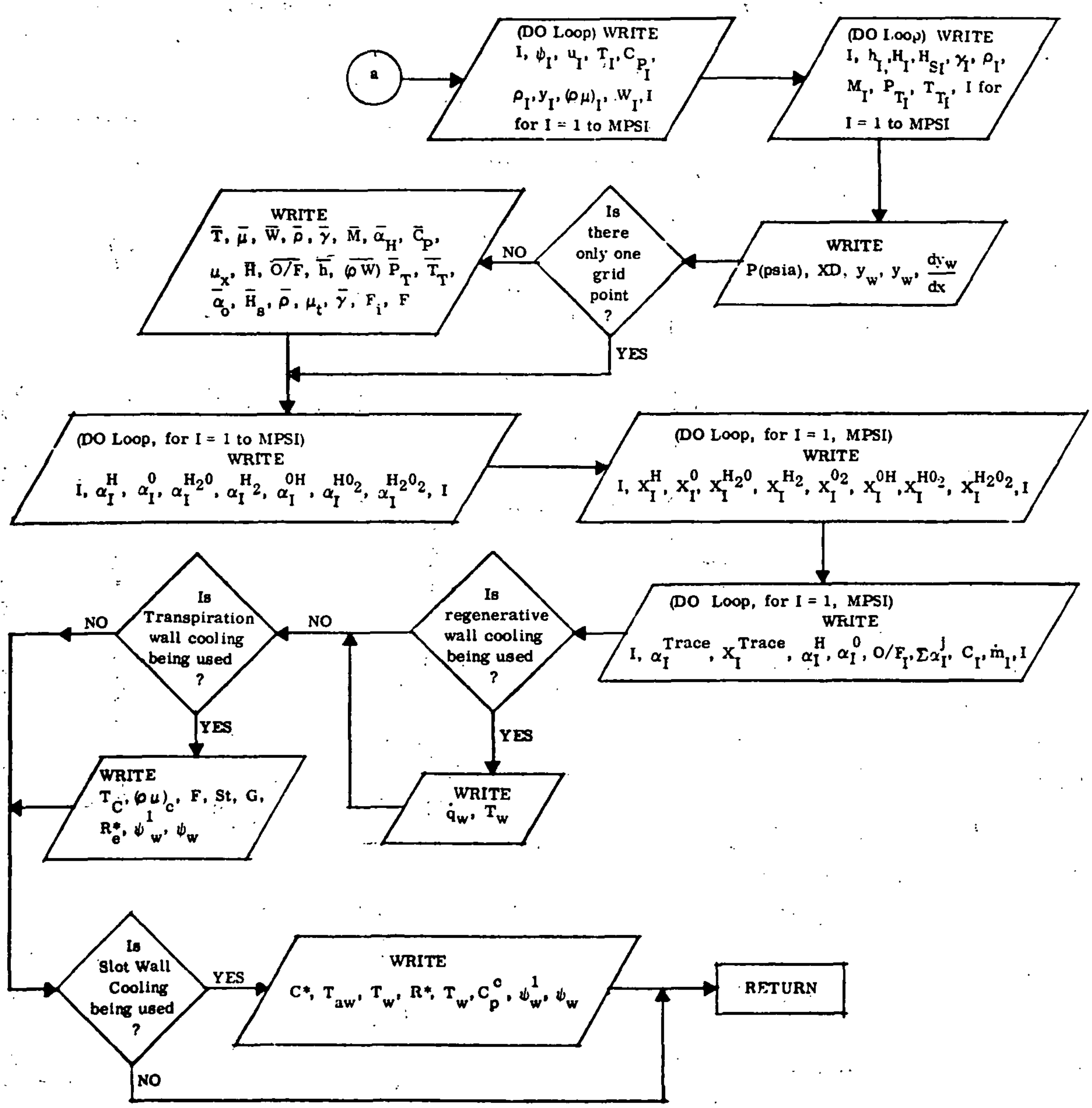
In this subroutine, the solution of the chemical kinetics matrix, $y_i = A_{ij}^{-1} B_i$ is obtained:


Determine the inverse of A_{ij}^{-1}, namely A_{ij}
Is the Determinant of $A_{ij}^{-1} < \epsilon$?
YES
WRITE Failure message -have singular matrix
NO
Solve A_{ij} B_i for y_i using Crout's method
RETURN

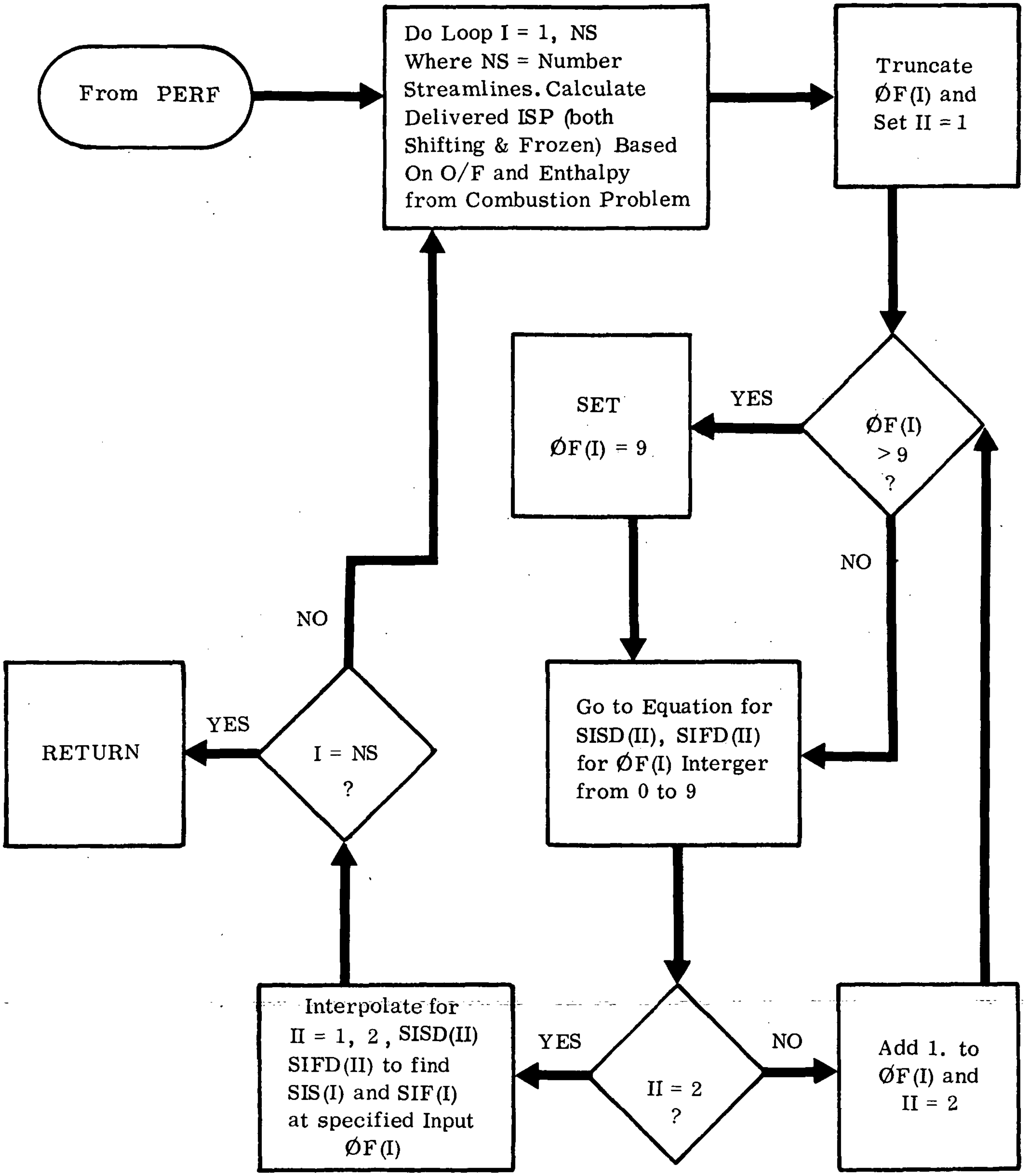
SUBROUTINE PRINT


Is this the first axial Step ?
YES
NO
g_c = 9.80665
PG = 7 g_c,
DO LOOP
I = 1 to MPSI
(DO Loop)
Compute sensible total enthalpy (RH(I))
equal to $H_I - \sum_{j=1}^{n} \alpha_I^j \Delta^j$
Increment I
NO
XMU(1) = 10^{-6}
NO
YES
Is there more than one grid point ?
Compute γ_I, C_I, M_I, P_{T_I}, T_{T_I}
(RALPHA(I,I), RALPHA (2, I),
(RALPHA (3,I), RU(I), RT(I))
(DO Loop)
Compute mole fractions,
$X_I^j = \left(\frac{\alpha}{W}\right)_I^j W_I$, and
SUM(I) = $\sum_{j=1}^{n} \alpha_I^j$
Is the DO Loop Complete ?
YES
BULK QUANTITIES
Is there only one grid point ?
NO
YES
25
Is P(X) inputted for the duct ?
YES
NO
Compute the wall location y_w
Is $\bar{\bar{\alpha}}_0 \leq .0$?
NO
YES
$\bar{\varphi} = \frac{W_0}{W_H} \frac{\bar{\bar{\alpha}}_H}{\bar{\bar{\alpha}}_0}$
$\bar{\varphi} = 1 \times 10^{30}$
Is $\bar{\bar{\alpha}}_H \leq .0$?
NO
YES
$\overline{O/F} = \frac{\bar{\bar{\alpha}}_0}{\bar{\bar{\alpha}}_I}$
$\overline{O/F} = 1 \times 10^{30}$
Compute bulk properties Radially across the chamber:
$\bar{H}_s$, $\bar{\gamma}$, $\bar{C}$, $\bar{M}$, $\bar{P}_T$, $\bar{T}_T$
$\dot{m} = \dot{m}_w$
$A = y_w$
Is the flow field axisymmetric ?
NO
$\dot{m} = \pi y_w \dot{m}$
$A = \pi y_w A$
$F_i = P A + \dot{m} \bar{u}$
Is this the first axial step ?
YES
$F_o = F_i$
$F = F_i - F_o$
25
P (psia) = $\frac{(P(N/M^2)}{101325.}$
XD = X/XK
IPAGE = IPAGE + 1
WRITE
Title information,
X, IPAGE,
P, Le, Pr, μ_t, Δx,
IOUT
a

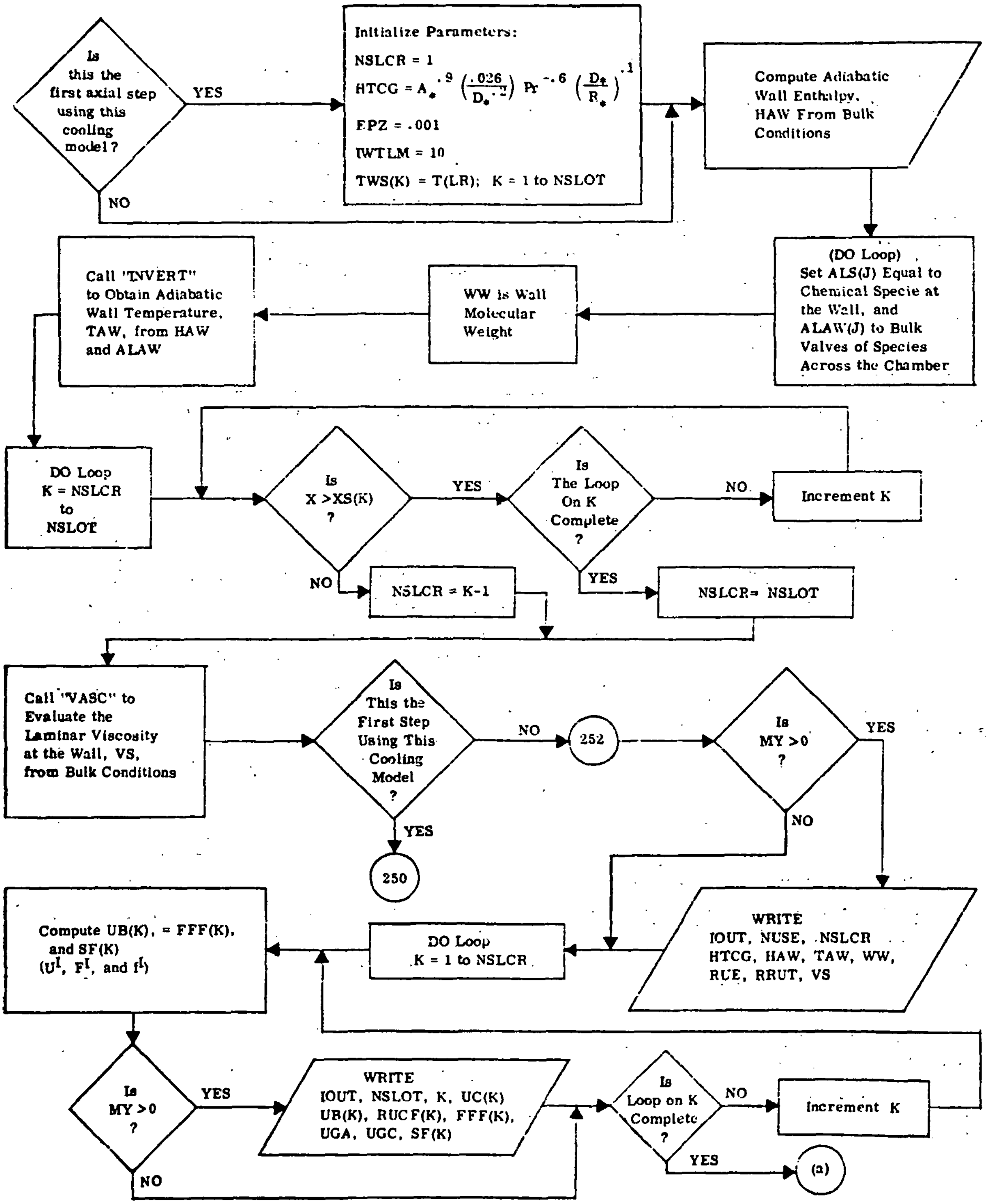
SUBROUTINE PRINT (Cont.)


a
(DO Loop) WRITE
$I, \phi_I, u_I, T_I, C_{P_I}, \rho_I, y_I, (\rho\mu)_I, W_I, I$
for I = 1 to MPSI
(DO Loop) WRITE
$I, h_I, H_I, H_{SI}, \gamma_I, \rho_I, M_I, P_{T_I}, T_{T_I}$, I for
I = 1 to MPSI
WRITE
P(psia), XD, y_w, y_w, $\frac{dy_w}{dx}$
Is there only one grid point ?
NO
YES
WRITE
$\bar{T}, \bar{\mu}, \bar{W}, \bar{\rho}, \bar{\gamma}, \bar{M}, \bar{\alpha}_H, \bar{C}_P, u_x, \bar{H}, \overline{O/F}, \bar{h}, (\overline{\rho W}) \bar{P}_T, \bar{T}_T, \bar{\alpha}_o, \bar{H}_s, \bar{\rho}, \mu_t, \bar{\gamma}, F_i, F$
(DO Loop, for I = 1 to MPSI)
WRITE
$I, \alpha_I^H, \alpha_I^0, \alpha_I^{H_20}, \alpha_I^{H_2}, \alpha_I^{0H}, \alpha_I^{H0_2}, \alpha_I^{H_20_2}, I$
(DO Loop, for I = 1, MPSI)
WRITE
$I, X_I^H, X_I^0, X_I^{H_20}, X_I^{H_2}, X_I^{0_2}, X_I^{0H}, X_I^{H0_2}, X_I^{H_20_2}, I$
(DO Loop, for I = 1, MPSI)
WRITE
$I, \alpha_I^{Trace}, X_I^{Trace}, \alpha_I^H, \alpha_I^0, O/F_I, \Sigma\alpha_I^j, C_I, \dot{m}_I, I$
Is regenerative wall cooling being used ?
NO
YES
WRITE
$\dot{q}_w, T_w$
Is Transpiration wall cooling being used ?
NO
YES
WRITE
$T_C, (\rho u)_c, F, St, G, R_e^*, \psi_w^1, \psi_w$
Is Slot Wall Cooling being used ?
YES
NO
WRITE
$C^*, T_{aw}, T_w, R^*, T_w, C_p^c, \psi_w^1, \psi_w$
RETURN

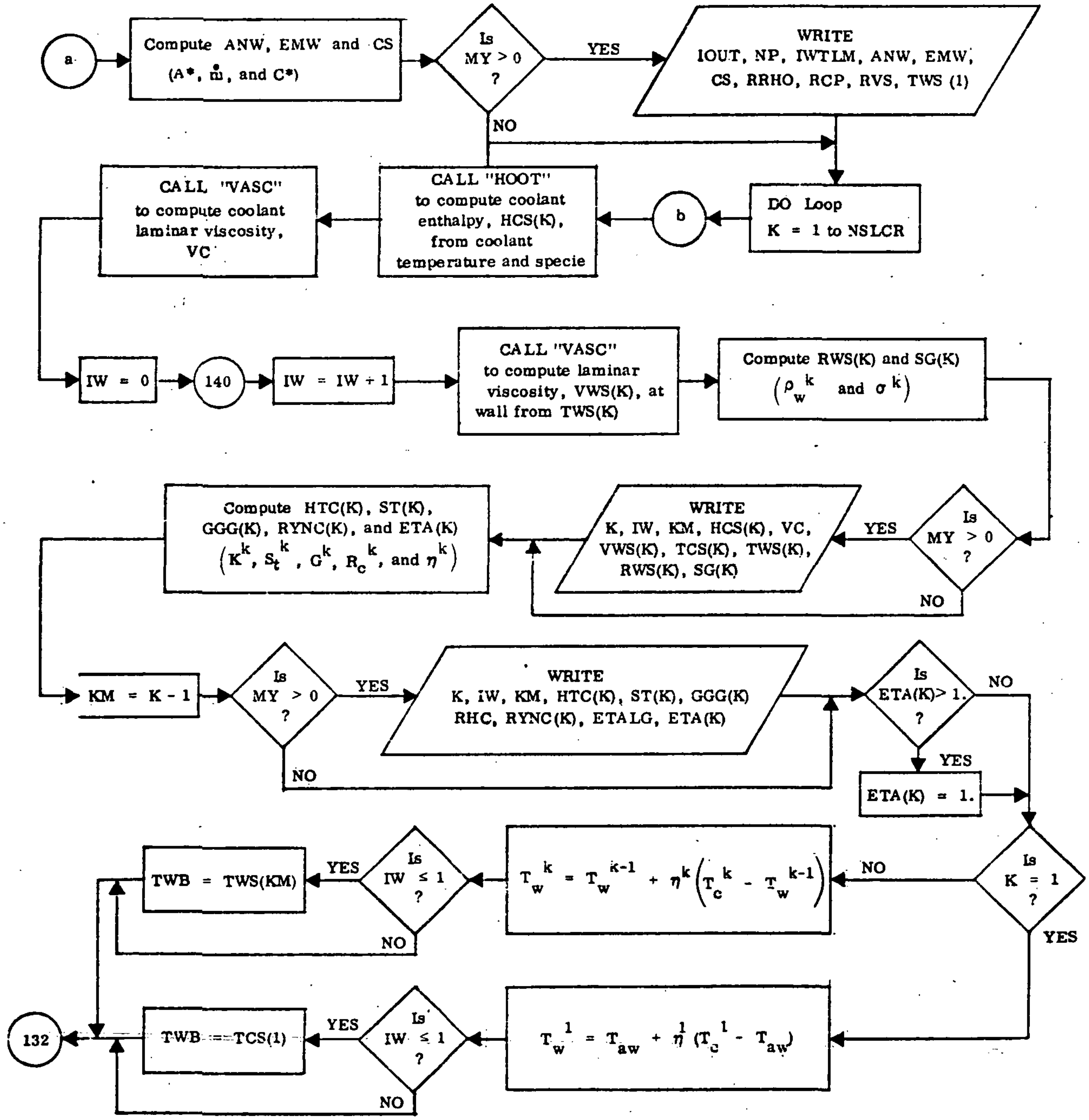
SUBROUTINE SI


From PERF
Do Loop I = 1, NS Where NS = Number Streamlines. Calculate Delivered ISP (both Shifting & Frozen) Based On O/F and Enthalpy from Combustion Problem
Truncate ØF(I) and Set II = 1
ØF(I) > 9 ?
YES
SET ØF(I) = 9
NO
Go to Equation for SISD(II), SIFD(II) for ØF(I) Interger from 0 to 9
II = 2 ?
YES
NO
Add 1. to ØF(I) and II = 2
Interpolate for II = 1, 2, SISD(II) SIFD(II) to find SIS(I) and SIF(I) at specified Input ØF(I)
I = NS ?
YES
RETURN
NO

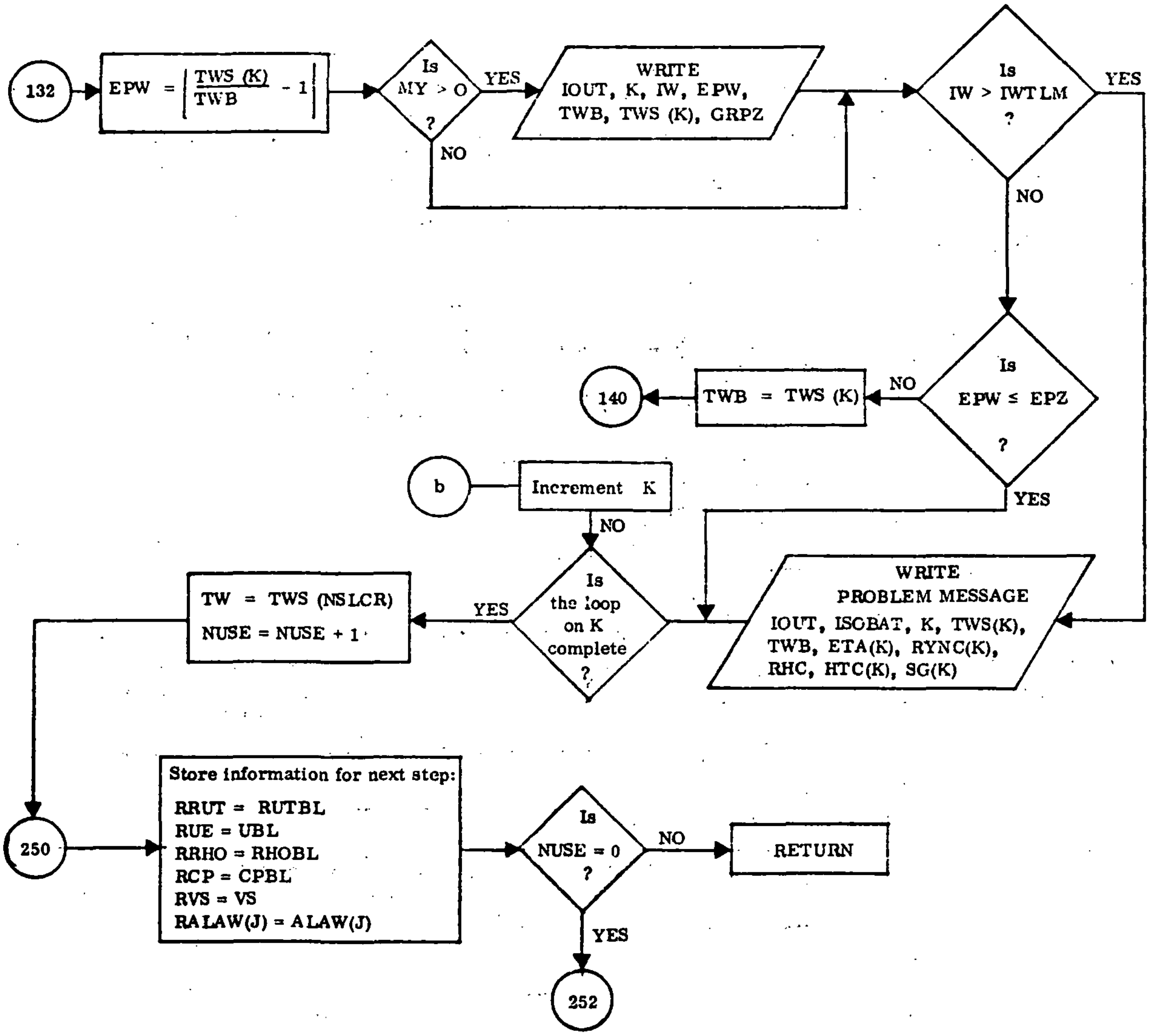
SUBROUTINE SLOT


Is this the first axial step using this cooling model ?
YES
NO
Initialize Parameters:
NSLCR = 1
HTCG = $A_*^{.9} \left(\frac{.026}{D_*^{.2}}\right) Pr^{-.6} \left(\frac{D_*}{R_*}\right)^{.1}$
EPZ = .001
IWTLM = 10
TWS(K) = T(LR); K = 1 to NSLOT
Compute Adiabatic Wall Enthalpy, HAW From Bulk Conditions
(DO Loop) Set ALS(J) Equal to Chemical Specie at the Wall, and ALAW(J) to Bulk Valves of Species Across the Chamber
WW is Wall Molecular Weight
Call "INVERT" to Obtain Adiabatic Wall Temperature, TAW, from HAW and ALAW
DO Loop K = NSLCR to NSLOT
Is X >XS(K) ?
YES
NO
Is The Loop On K Complete ?
NO
YES
Increment K
NSLCR = K-1
NSLCR= NSLOT
Call "VASC" to Evaluate the Laminar Viscosity at the Wall, VS, from Bulk Conditions
Is This the First Step Using This Cooling Model ?
NO
YES
252
250
Is MY >0 ?
YES
NO
WRITE IOUT, NUSE, NSLCR HTCG, HAW, TAW, WW, RUE, RRUT, VS
DO Loop K = 1 to NSLCR
Compute UB(K), = FFF(K), and SF(K) (U^I, F^I, and f^I)
Is MY >0 ?
YES
NO
WRITE IOUT, NSLOT, K, UC(K) UB(K), RUCF(K), FFF(K), UGA, UGC, SF(K)
Is Loop on K Complete ?
NO
YES
Increment K
(a)

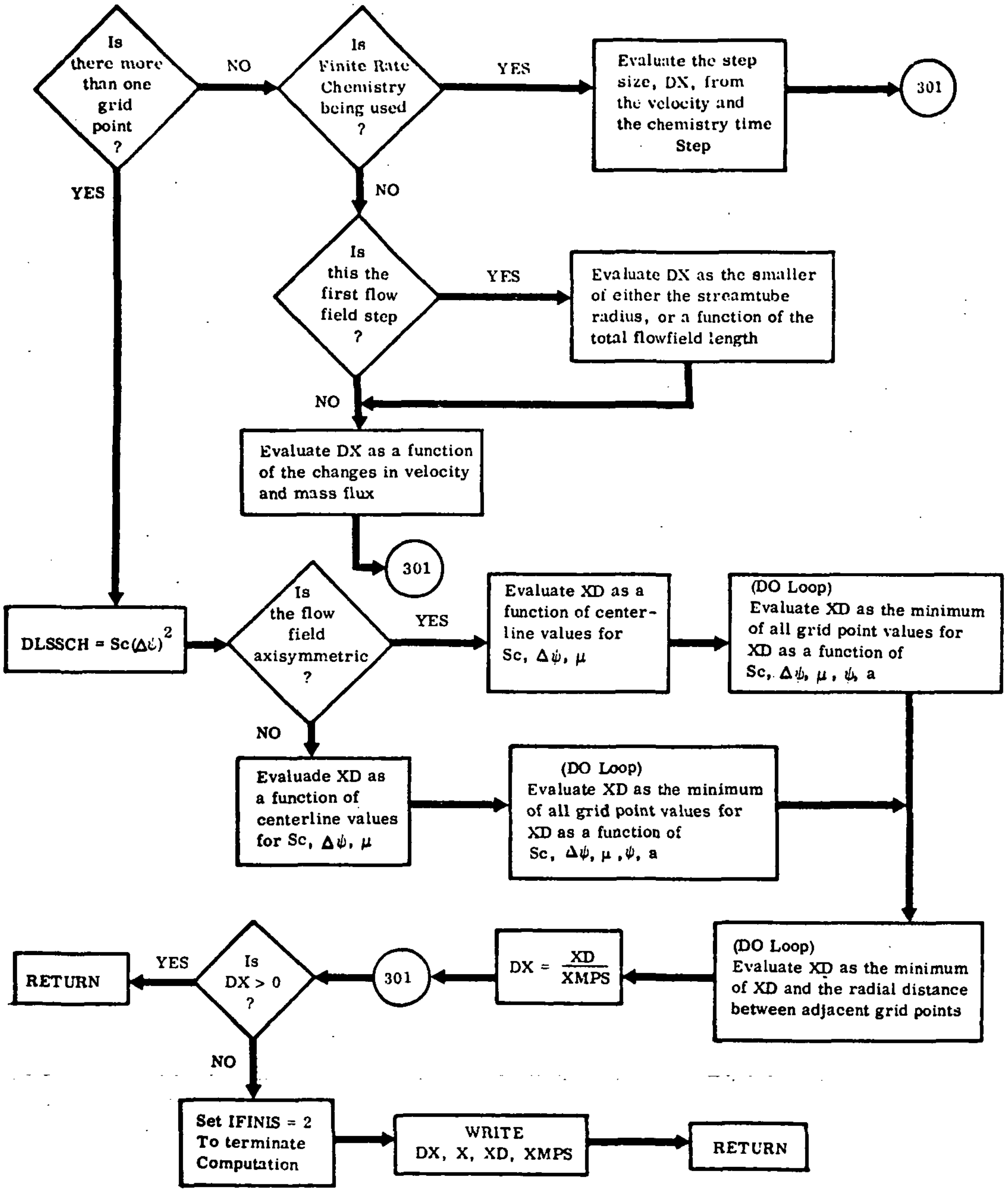
SUBROUTINE SLOT (Cont.)


a
Compute ANW, EMW and CS
(A*, $\dot{m}$, and C*)
Is MY > 0 ?
YES
WRITE IOUT, NP, IWTLM, ANW, EMW, CS, RRHO, RCP, RVS, TWS (1)
NO
DO Loop K = 1 to NSLCR
b
CALL "HOOT" to compute coolant enthalpy, HCS(K), from coolant temperature and specie
CALL "VASC" to compute coolant laminar viscosity, VC
IW = 0
140
IW = IW + 1
CALL "VASC" to compute laminar viscosity, VWS(K), at wall from TWS(K)
Compute RWS(K) and SG(K)
$\left(\rho_w^{\ k}\right.$ and $\left.\sigma^k\right)$
Is MY > 0 ?
YES
NO
WRITE K, IW, KM, HCS(K), VC, VWS(K), TCS(K), TWS(K), RWS(K), SG(K)
Compute HTC(K), ST(K), GGG(K), RYNC(K), and ETA(K)
$\left(K^k, S_t^{\ k}, G^k, R_c^{\ k}, \text{and } \eta^k\right)$
KM = K - 1
Is MY > 0 ?
YES
NO
WRITE K, IW, KM, HTC(K), ST(K), GGG(K) RHC, RYNC(K), ETALG, ETA(K)
Is ETA(K)> 1. ?
NO
YES
ETA(K) = 1.
Is K = 1 ?
NO
YES
$T_w^{\ k} = T_w^{\ k-1} + \eta^k\left(T_c^{\ k} - T_w^{\ k-1}\right)$
Is IW ≤ 1 ?
YES
NO
TWB = TWS(KM)
$T_w^{\ 1} = T_{aw} + \eta^1 (T_c^{\ 1} - T_{aw})$
Is IW ≤ 1 ?
YES
NO
TWB = TCS(1)
132

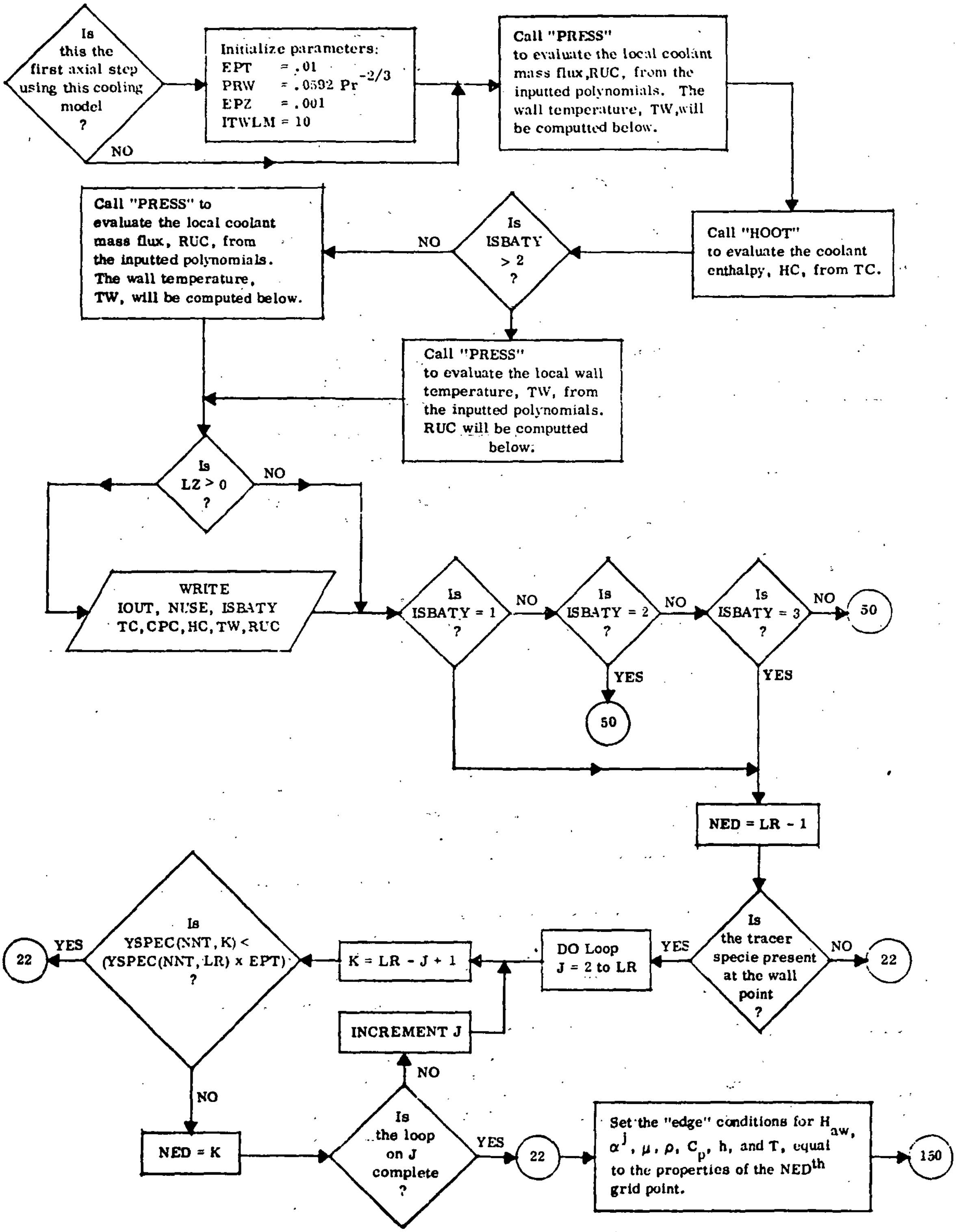
SUBROUTINE SLOT (Cont.)



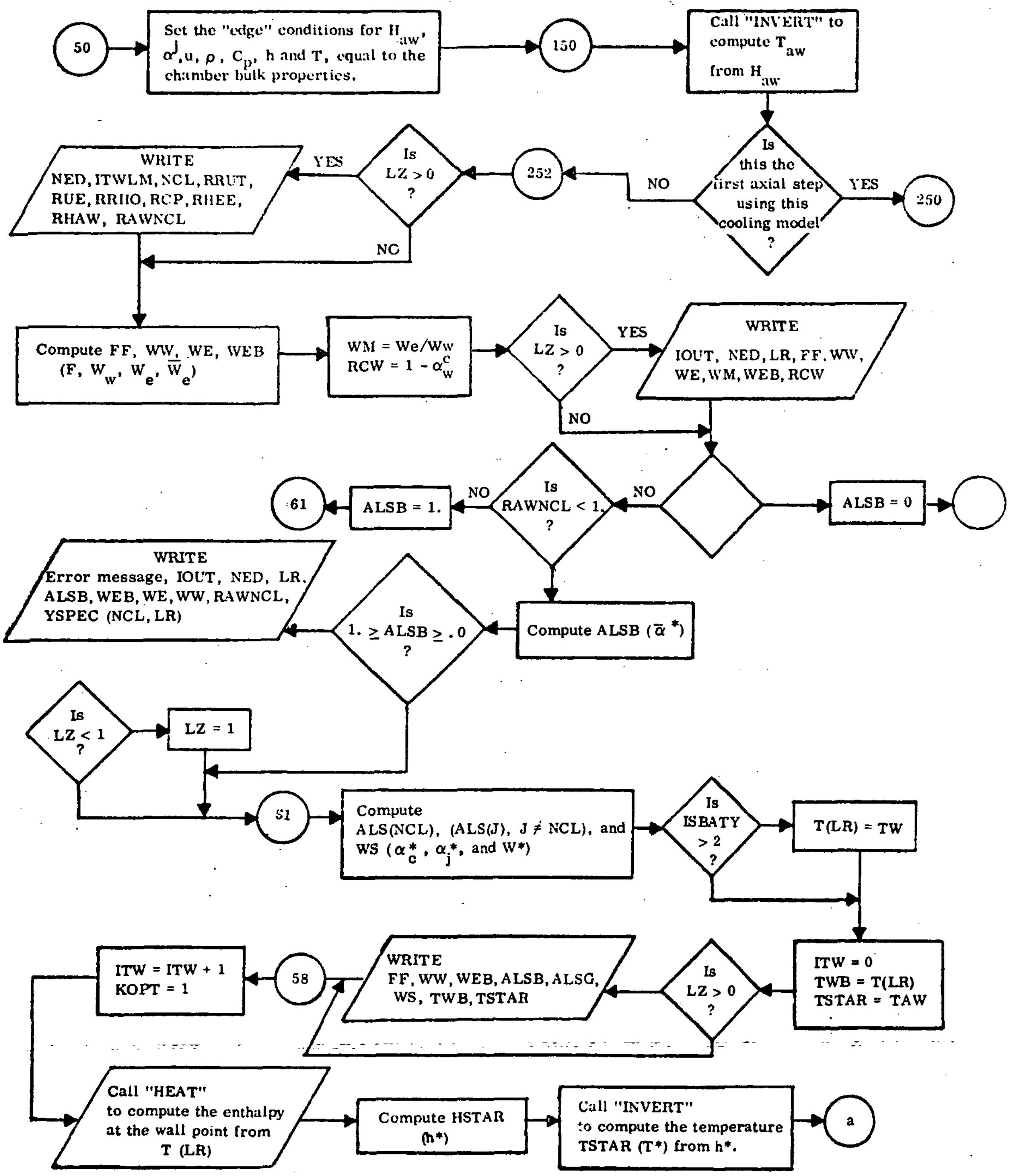
SUBROUTINE STEP


Is there more than one grid point ?
NO
Is Finite Rate Chemistry being used ?
YES
Evaluate the step size, DX, from the velocity and the chemistry time Step
301
YES
NO
Is this the first flow field step ?
YES
Evaluate DX as the smaller of either the streamtube radius, or a function of the total flowfield length
NO
Evaluate DX as a function of the changes in velocity and mass flux
301
DLSSCH = Sc(Δψ)²
Is the flow field axisymmetric ?
YES
Evaluate XD as a function of center-line values for Sc, Δψ, μ
(DO Loop) Evaluate XD as the minimum of all grid point values for XD as a function of Sc, Δψ, μ, ψ, a
NO
Evaluade XD as a function of centerline values for Sc, Δψ, μ
(DO Loop) Evaluate XD as the minimum of all grid point values for XD as a function of Sc, Δψ, μ, ψ, a
(DO Loop) Evaluate XD as the minimum of XD and the radial distance between adjacent grid points
DX = XD / XMPS
301
Is DX > 0 ?
YES
RETURN
NO
Set IFINIS = 2 To terminate Computation
WRITE DX, X, XD, XMPS
RETURN

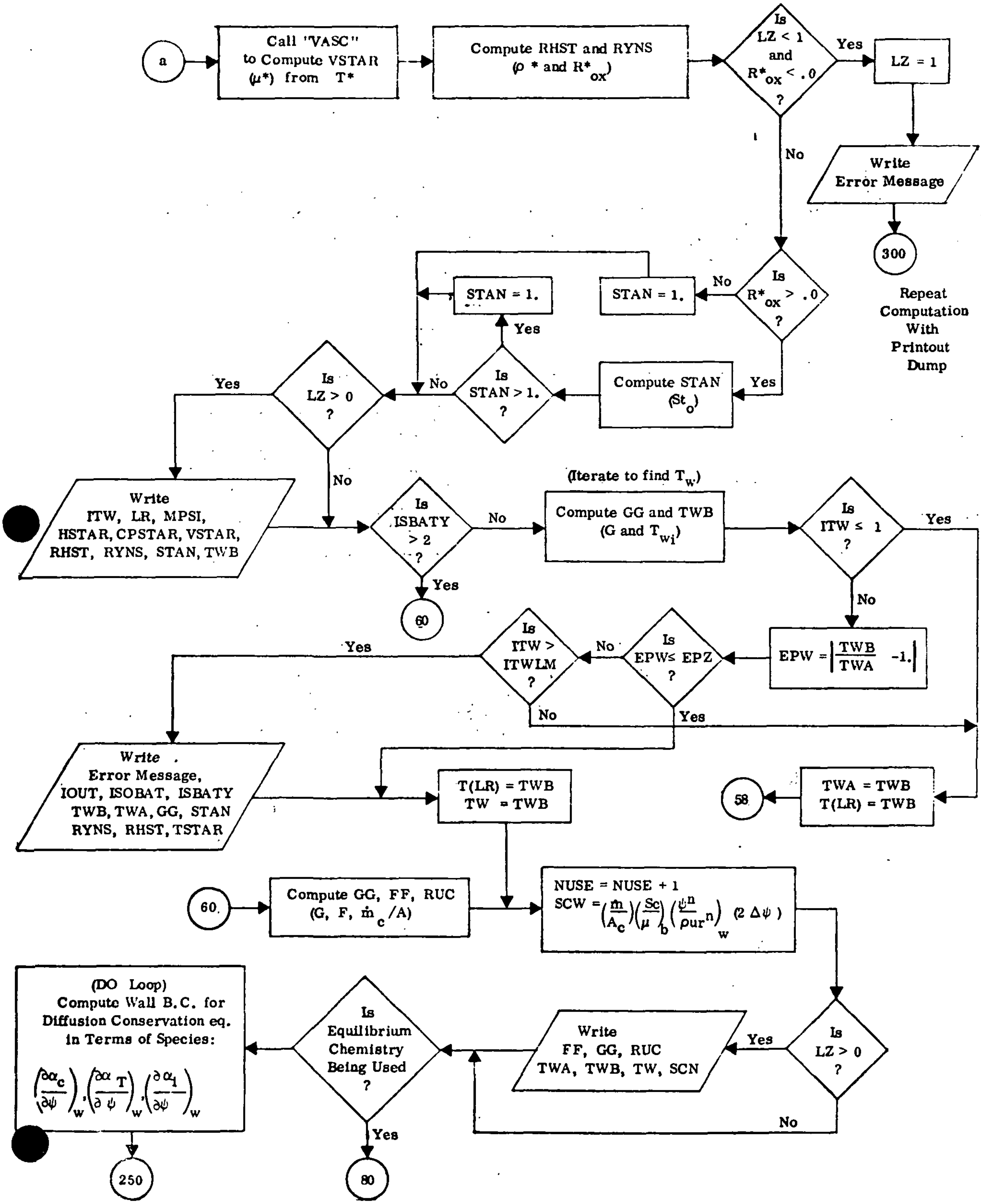
SUBROUTINE TRANSP


Is this the first axial step using this cooling model ?
NO
Initialize parameters:
EPT = .01
PRW = .0592 Pr$^{-2/3}$
EPZ = .001
ITWLM = 10
Call "PRESS" to evaluate the local coolant mass flux,RUC, from the inputted polynomials. The wall temperature, TW,will be computted below.
Call "HOOT" to evaluate the coolant enthalpy, HC, from TC.
Is ISBATY > 2 ?
NO
Call "PRESS" to evaluate the local coolant mass flux, RUC, from the inputted polynomials. The wall temperature, TW, will be computed below.
Call "PRESS" to evaluate the local wall temperature, TW, from the inputted polynomials. RUC will be computted below.
Is LZ > 0 ?
NO
WRITE IOUT, NUSE, ISBATY TC,CPC,HC,TW,RUC
Is ISBATY = 1 ?
NO
Is ISBATY = 2 ?
NO
YES
50
Is ISBATY = 3 ?
NO
50
YES
NED = LR - 1
Is the tracer specie present at the wall point ?
NO
22
YES
DO Loop J = 2 to LR
K = LR - J + 1
INCREMENT J
Is YSPEC(NNT,K) < (YSPEC(NNT,LR) x EPT) ?
YES
22
NO
NED = K
Is the loop on J complete ?
NO
YES
22
Set the "edge" conditions for H_{aw}, α^j, μ, ρ, C_p, h, and T, equal to the properties of the NEDth grid point.
150

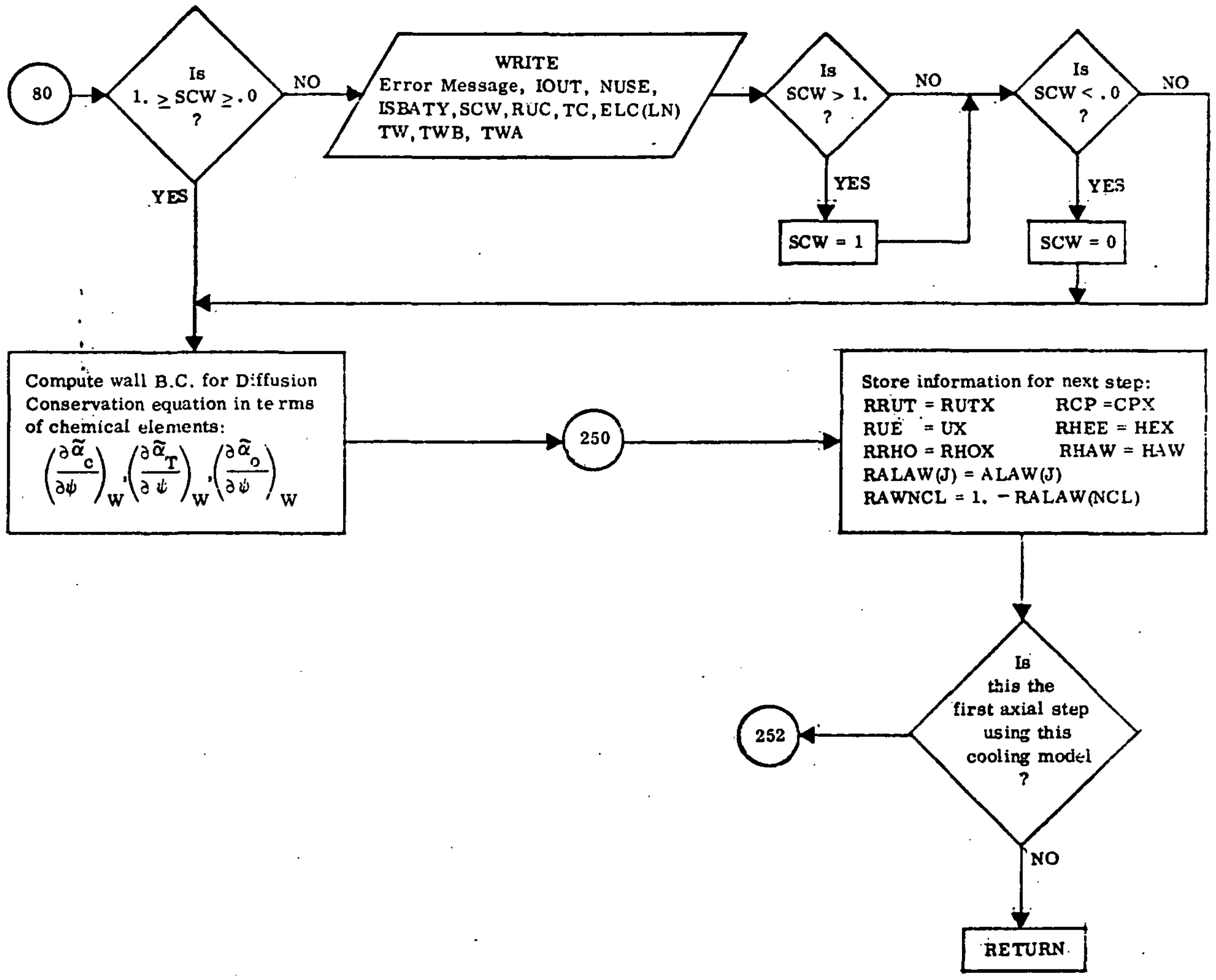
SUBROUTINE TRANSP (Cont.)


50
Set the "edge" conditions for H_{aw}, α^j, u, ρ, C_p, h and T, equal to the chamber bulk properties.
150
Call "INVERT" to compute T_{aw} from H_{aw}
Is this the first axial step using this cooling model ?
YES
250
NO
252
Is LZ > 0 ?
YES
WRITE NED, ITWLM, NCL, RRUT, RUE, RRHO, RCP, RHEE, RHAW, RAWNCL
NO
Compute FF, WW, WE, WEB (F, W_w, W_e, $\overline{W}_e$)
WM = We/Ww RCW = $1 - \alpha_w^c$
Is LZ > 0 ?
YES
WRITE IOUT, NED, LR, FF, WW, WE, WM, WEB, RCW
NO
ALSB = 0
NO
Is RAWNCL < 1. ?
NO
ALSB = 1.
61
Compute ALSB ($\overline{\alpha}$ *)
Is 1. ≥ ALSB ≥ .0 ?
WRITE Error message, IOUT, NED, LR, ALSB, WEB, WE, WW, RAWNCL, YSPEC (NCL, LR)
Is LZ < 1 ?
LZ = 1
51
Compute ALS(NCL), (ALS(J), J ≠ NCL), and WS (α_c^*, α_j^*, and W*)
Is ISBATY > 2 ?
T(LR) = TW
ITW = 0 TWB = T(LR) TSTAR = TAW
Is LZ > 0 ?
WRITE FF, WW, WEB, ALSB, ALSG, WS, TWB, TSTAR
58
ITW = ITW + 1 KOPT = 1
Call "HEAT" to compute the enthalpy at the wall point from T (LR)
Compute HSTAR (h*)
Call "INVERT" to compute the temperature TSTAR (T*) from h*.
a

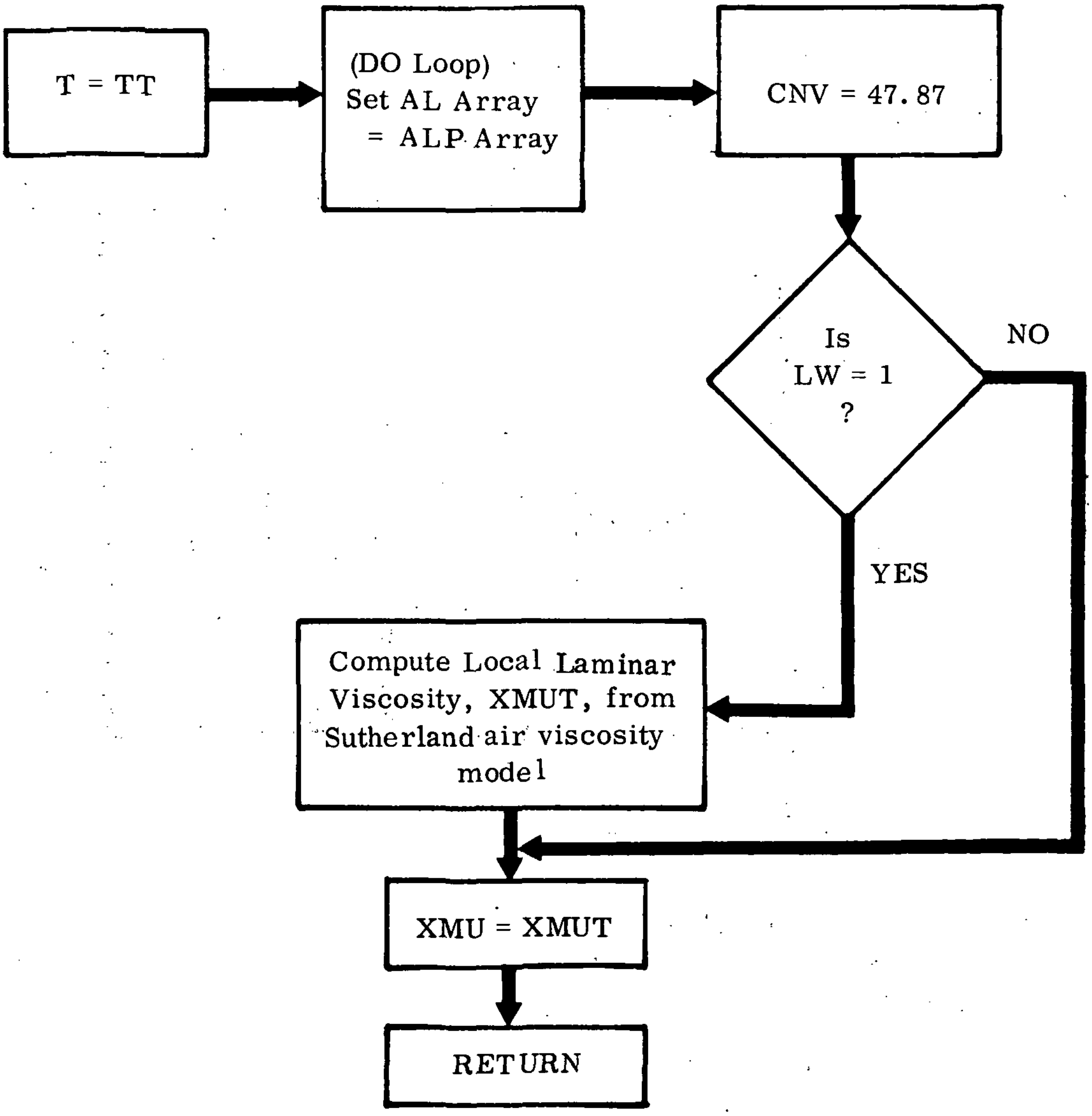
SUBROUTINE TRANSP (Cont.)


a
Call "VASC" to Compute VSTAR (μ*) from T*
Compute RHST and RYNS (ρ * and R*ox)
Is LZ < 1 and R*ox < .0 ?
Yes
LZ = 1
Write Error Message
300
Repeat Computation With Printout Dump
No
Is R*ox > .0 ?
No
STAN = 1.
STAN = 1.
Yes
Compute STAN (Sto)
Is STAN > 1. ?
Yes
No
Is LZ > 0 ?
Yes
Write ITW, LR, MPSI, HSTAR, CPSTAR, VSTAR, RHST, RYNS, STAN, TWB
No
Is ISBATY > 2 ?
Yes
60
No
(Iterate to find Tw)
Compute GG and TWB (G and Twi)
Is ITW ≤ 1 ?
Yes
No
EPW = |TWB/TWA -1.|
Is EPW≤ EPZ ?
No
Is ITW > ITWLM ?
Yes
No
Yes
Write Error Message, IOUT, ISOBAT, ISBATY TWB, TWA, GG, STAN RYNS, RHST, TSTAR
T(LR) = TWB TW = TWB
TWA = TWB T(LR) = TWB
58
60
Compute GG, FF, RUC (G, F, ṁc/A)
NUSE = NUSE + 1
SCW = (ṁ/Ac)(Sc/μ)b (ψn/ρurn)w (2 Δψ)
Is LZ > 0 ?
Yes
No
Write FF, GG, RUC TWA, TWB, TW, SCN
Is Equilibrium Chemistry Being Used ?
Yes
80
(DO Loop) Compute Wall B.C. for Diffusion Conservation eq. in Terms of Species:
(∂αc/∂ψ)w, (∂αT/∂ψ)w, (∂α1/∂ψ)w
250

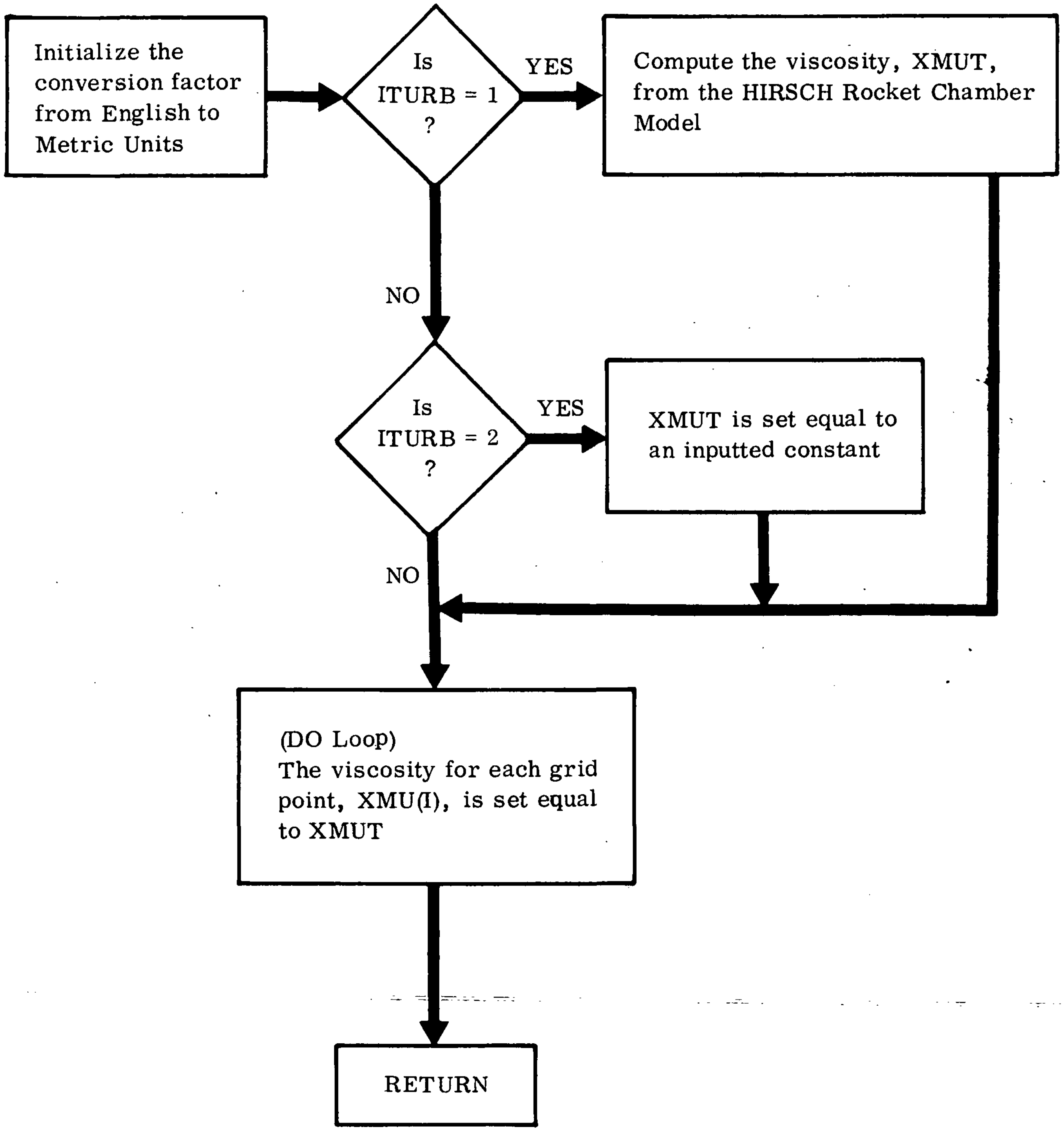
SUBROUTINE TRANSP (Cont.)


80
Is 1. ≥ SCW ≥ .0 ?
NO
YES
WRITE Error Message, IOUT, NUSE, ISBATY, SCW, RUC, TC, ELC(LN) TW, TWB, TWA
Is SCW > 1. ?
NO
YES
SCW = 1
Is SCW < .0 ?
NO
YES
SCW = 0
Compute wall B.C. for Diffusion Conservation equation in terms of chemical elements:
$\left(\frac{\partial\tilde{\alpha}_c}{\partial\psi}\right)_W, \left(\frac{\partial\tilde{\alpha}_T}{\partial\psi}\right)_W, \left(\frac{\partial\tilde{\alpha}_o}{\partial\psi}\right)_W$
250
Store information for next step:
RRUT = RUTX
RCP = CPX
RUE = UX
RHEE = HEX
RRHO = RHOX
RHAW = HAW
RALAW(J) = ALAW(J)
RAWNCL = 1. − RALAW(NCL)
Is this the first axial step using this cooling model ?
252
NO
RETURN

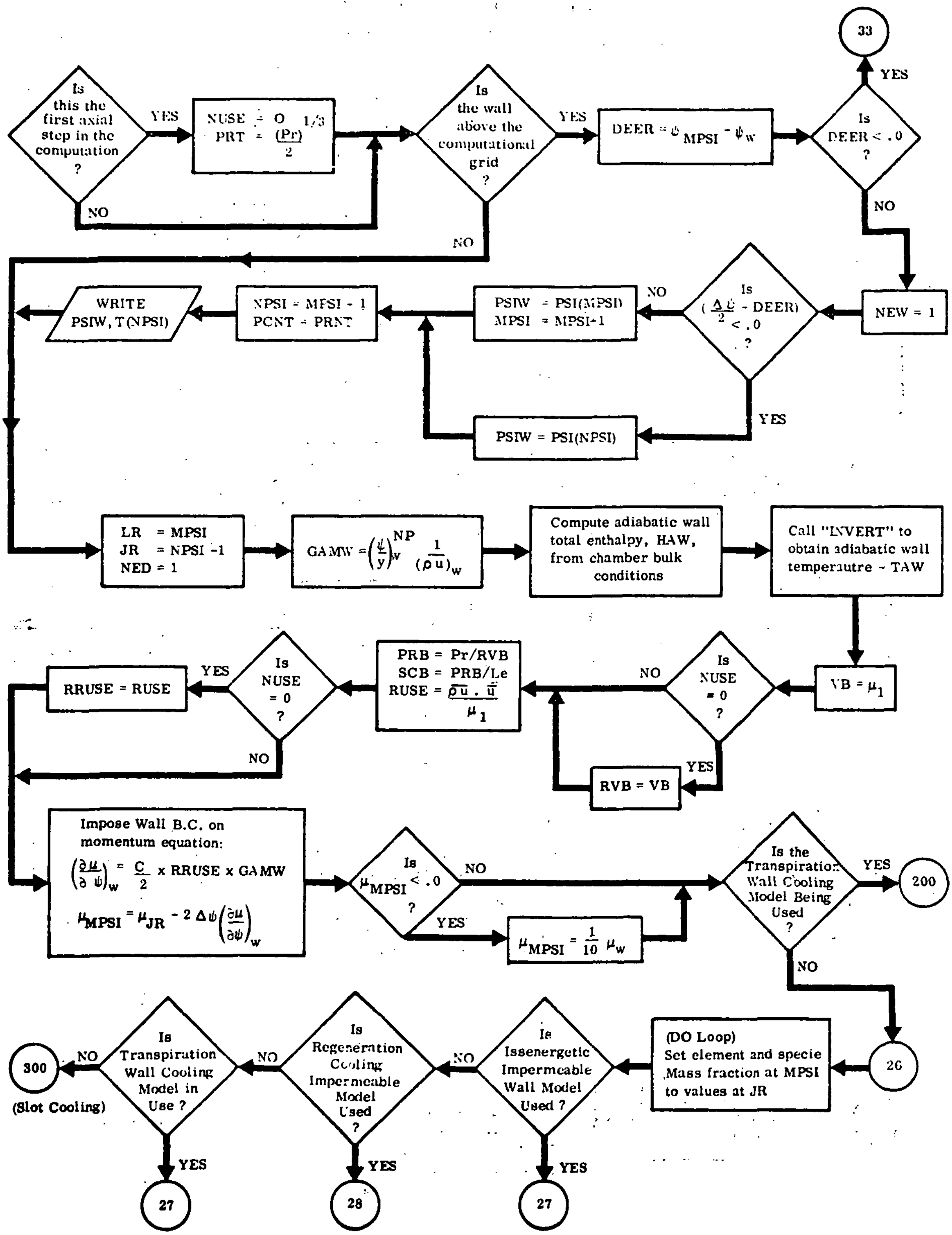
SUBROUTINE VASC (XMU, TT, ALP)



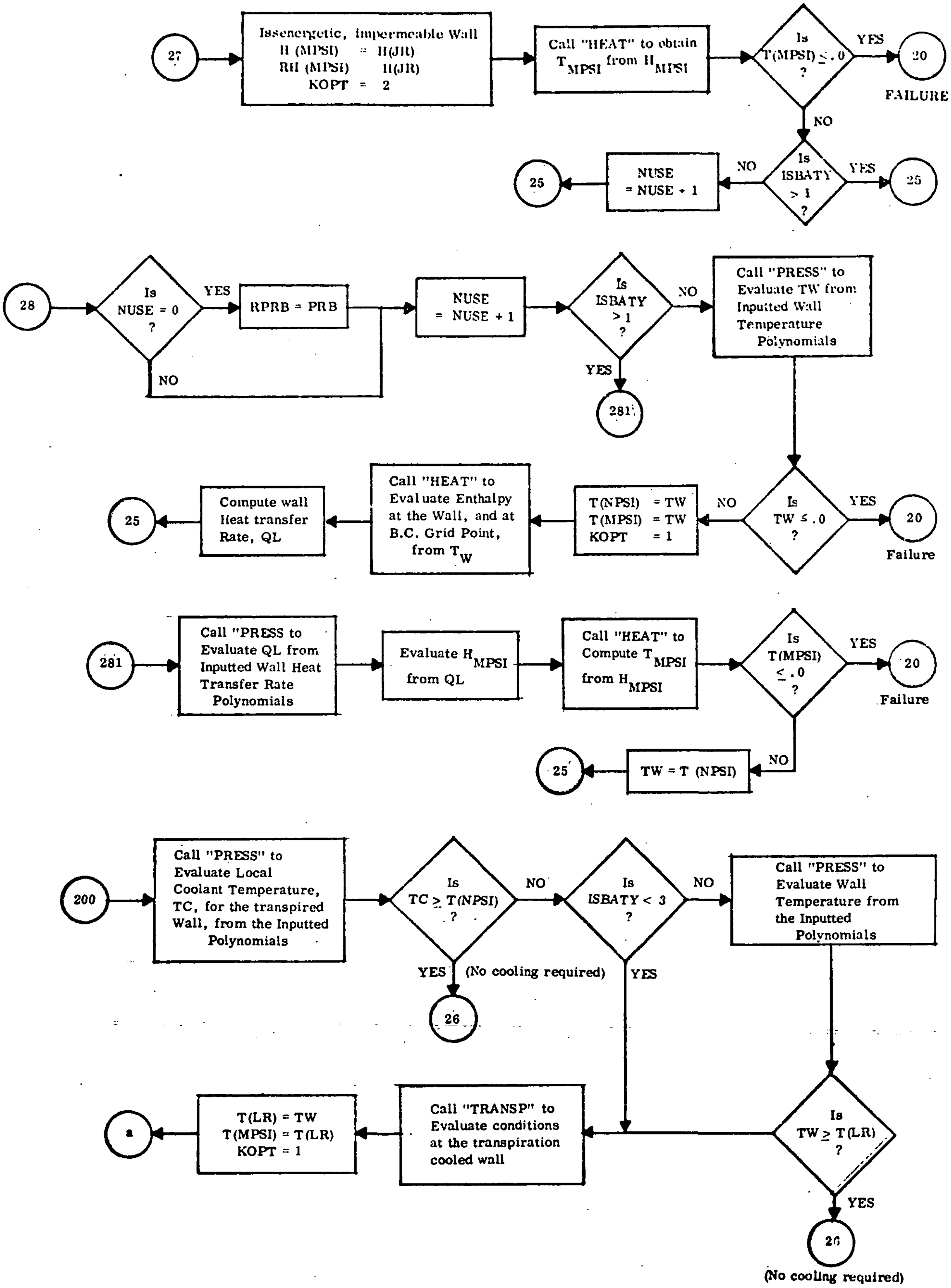
SUBROUTINE VISC


Initialize the conversion factor from English to Metric Units
Is ITURB = 1 ?
YES
Compute the viscosity, XMUT, from the HIRSCH Rocket Chamber Model
NO
Is ITURB = 2 ?
YES
XMUT is set equal to an inputted constant
NO
(DO Loop) The viscosity for each grid point, XMU(I), is set equal to XMUT
RETURN

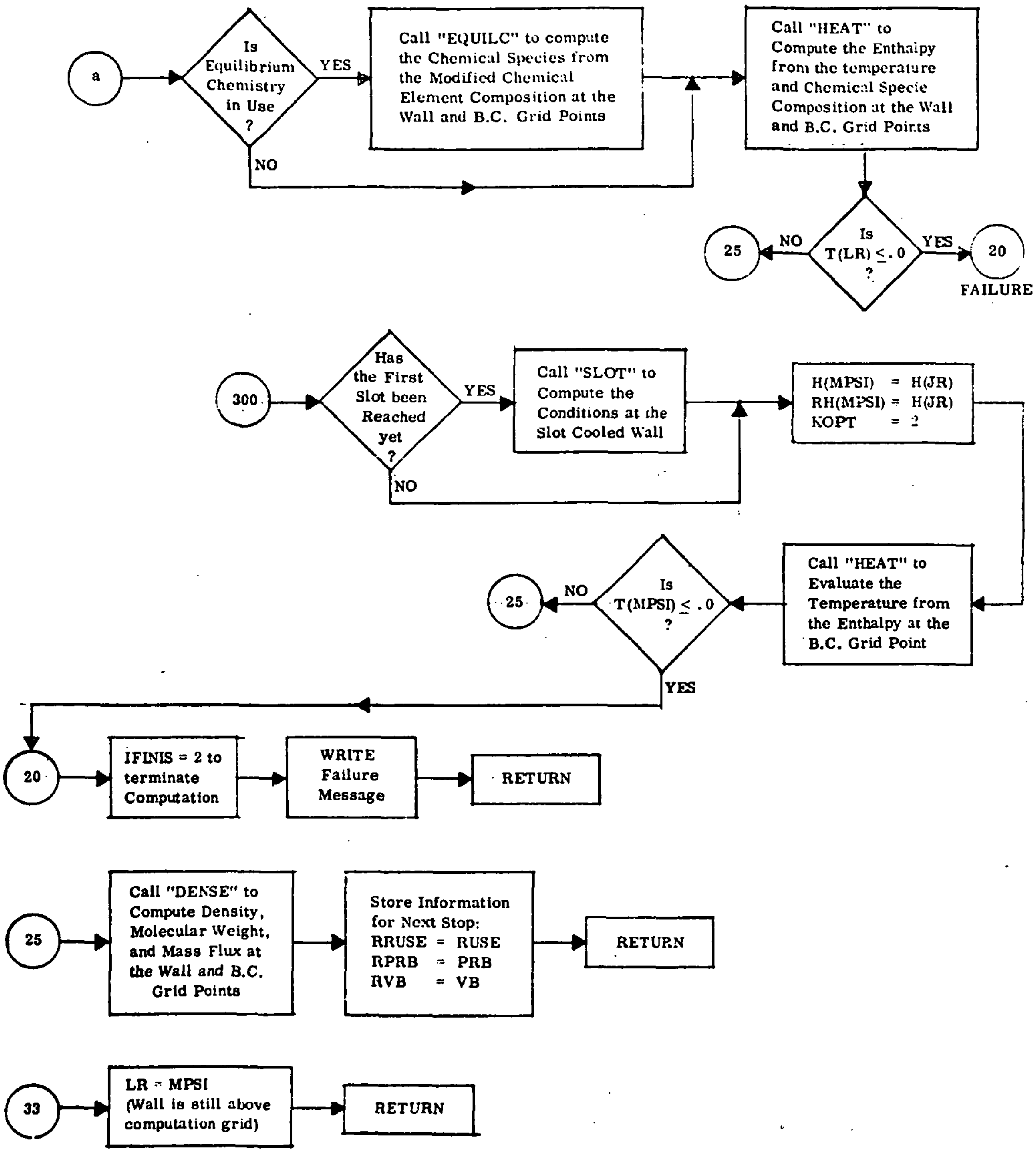
SUBROUTINE WALL


Is this the first axial step in the computation ?
YES
NO
NUSE = O
PRT = (Pr)^1/3 / 2
Is the wall above the computational grid ?
YES
NO
DEER = ψ MPSI - ψw
Is DEER < .0 ?
YES
NO
33
NEW = 1
Is (Δψ/2 - DEER) < .0 ?
NO
YES
PSIW = PSI(MPSI)
MPSI = MPSI+1
PSIW = PSI(NPSI)
NPSI = MPSI - 1
PCNT = PRNT
WRITE PSIW, T(NPSI)
LR = MPSI
JR = NPSI -1
NED = 1
GAMW = (ψ/y)w^NP 1/(ρu)w
Compute adiabatic wall total enthalpy, HAW, from chamber bulk conditions
Call "INVERT" to obtain adiabatic wall temperautre - TAW
VB = μ1
Is NUSE = 0 ?
NO
YES
RVB = VB
PRB = Pr/RVB
SCB = PRB/Le
RUSE = ρu . u / μ1
Is NUSE = 0 ?
YES
NO
RRUSE = RUSE
Impose Wall B.C. on momentum equation:
(∂μ/∂ψ)w = C/2 x RRUSE x GAMW
μMPSI = μJR - 2 Δψ (∂μ/∂ψ)w
Is μMPSI < .0 ?
NO
YES
μMPSI = 1/10 μw
Is the Transpiration Wall Cooling Model Being Used ?
YES
200
NO
26
(DO Loop) Set element and specie Mass fraction at MPSI to values at JR
Is Issenergetic Impermeable Wall Model Used ?
NO
YES
27
Is Regeneration Cooling Impermeable Model Used ?
NO
YES
28
Is Transpiration Wall Cooling Model in Use ?
NO
YES
27
300
(Slot Cooling)

SUBROUTINE WALL (Cont.)


27
Issenergetic, Impermeable Wall
H (MPSI) = H(JR)
RH (MPSI) H(JR)
KOPT = 2
Call "HEAT" to obtain T MPSI from H MPSI
Is T(MPSI) ≤ . 0 ?
YES
20
FAILURE
NO
Is ISBATY > 1 ?
YES
25
NO
NUSE = NUSE + 1
25
28
Is NUSE = 0 ?
YES
RPRB = PRB
NO
NUSE = NUSE + 1
Is ISBATY > 1 ?
NO
YES
281
Call "PRESS" to Evaluate TW from Inputted Wall Temperature Polynomials
Is TW ≤ . 0 ?
YES
20
Failure
NO
T(NPSI) = TW
T(MPSI) = TW
KOPT = 1
Call "HEAT" to Evaluate Enthalpy at the Wall, and at B.C. Grid Point, from T W
Compute wall Heat transfer Rate, QL
25
281
Call "PRESS to Evaluate QL from Inputted Wall Heat Transfer Rate Polynomials
Evaluate H MPSI from QL
Call "HEAT" to Compute T MPSI from H MPSI
Is T(MPSI) ≤ . 0 ?
YES
20
Failure
NO
TW = T (NPSI)
25
200
Call "PRESS" to Evaluate Local Coolant Temperature, TC, for the transpired Wall, from the Inputted Polynomials
Is TC ≥ T(NPSI) ?
YES
(No cooling required)
26
NO
Is ISBATY < 3 ?
YES
NO
Call "PRESS" to Evaluate Wall Temperature from the Inputted Polynomials
Is TW ≥ T(LR) ?
YES
26
(No cooling required)
Call "TRANSP" to Evaluate conditions at the transpiration cooled wall
T(LR) = TW
T(MPSI) = T(LR)
KOPT = 1
a

SUBROUTINE WALL (Cont.)


a
Is Equilibrium Chemistry in Use ?
YES
NO
Call "EQUILC" to compute the Chemical Species from the Modified Chemical Element Composition at the Wall and B.C. Grid Points
Call "HEAT" to Compute the Enthalpy from the temperature and Chemical Specie Composition at the Wall and B.C. Grid Points
Is T(LR) ≤ .0 ?
NO
25
YES
20
FAILURE
300
Has the First Slot been Reached yet ?
YES
NO
Call "SLOT" to Compute the Conditions at the Slot Cooled Wall
H(MPSI) = H(JR)
RH(MPSI) = H(JR)
KOPT = 2
Call "HEAT" to Evaluate the Temperature from the Enthalpy at the B.C. Grid Point
Is T(MPSI) ≤ .0 ?
NO
25
YES
20
IFINIS = 2 to terminate Computation
WRITE Failure Message
RETURN
25
Call "DENSE" to Compute Density, Molecular Weight, and Mass Flux at the Wall and B.C. Grid Points
Store Information for Next Stop:
RRUSE = RUSE
RPRB = PRB
RVB = VB
RETURN
33
LR = MPSI (Wall is still above computation grid)
RETURN

PRECEDING PAGE BLANK NOT FILMED

SUBROUTINE LISTINGS

```
00100   C     PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
00100   C DRIVER FOR O2/H2 ENGINE MODEL
00100   C
00100   C INPUT UNITS ARE INCHES,POUNDS,SEC
00100   C
00101         COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
00103         COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
00104         COMMON/ZL/GLX(7,4),          TCX(7,4),GL,TC,PRR,GAMW,RUC,RUCX(7,4)
00104        1,SCB,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
00104        2,GG,UB(21),UBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
00104        3,VWS(21),FWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,GGG(21)
00104        4,CPB,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
00104        5,NCL,NP,PYE,ANP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),HC,HCS(21)
00105         COMMON/STORNP/RRUSE,NUSE,RPRB,RRUT,RALAW(9),RHEE,RHAW,RUE,RRHO,RCP
00105        1,RVB,RVS
00106         COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
00106        1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XMU(200),U(200)
00106        2,A(200),RHC(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
00106        3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
00106        4,FEE(200)
00107         COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
00107        1,IDA(200),TCHEM(200),WDT(9,200)
00110         COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
00110        1,GSCALE,TWX(7,4)
00111         COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,IOUT,IPAGE,MY,NTYPE,
00111        1IR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
00111        2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
00111        3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
00111        4,NEW,MG,MF,LN,NNT,JR,NSLCR
00112         COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
00112        1OR,USTCR,RAY,RG,AK,AKA
00113         COMMON/ZB/CF(200),HSEN(200)
```

```
00114         COMMON/ZF/SBL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL,UTBL,CPRI,HEBL
00114        1,RHOBL,RUTBL,WMBL,TBL,GMBI,SSBL,EMBL,PTBL,TTBL
00115         COMMON/ADYN/NNODE,NCONN,CD,WOUTC,NOINJ(24),LPRTF,
00115        1NVOL(60),ICOMB(60),PRES(60),TEMP(60),RMIX(60),VOL(60),
00115        2IADMIT(60,2),IRTYPE(60),CAREA(60),CCOEFF(60),DLEN(60),TIMON(60),
00115        3TIMOFF(60),TIMOPN(60),TIMCLS(60),NNCINJ,TFSTST,
00115        4SPTIME,SPKTL,SPKF,SPGAP,SPARKP,SPARKE,DC,NCHEK,TIME,L,ICVOL,DPRES
00116         COMMON/NJECT/WT,PO,FM,PSTAT                                                  *NEW
00117         COMMON/CONP/NCOMBU,NINJE                                                     ***I
00120         DIMENSION NBLOW(24)
00121         ICAS=0
00122     751 ICAS=ICAS+1
00122   C READ MODEL CONTROL FLAGS
00123         READ (5,900) NDYNA,NCOMBU,NINJE,NHEAT,NCAS,NBLFG
00133         IF (NDYNA.EQ.0)  GO TO 1101
00135         IF (NCOMBU.EQ.0)  GO TO 1102
00137         IF (NINJE.EQ.1)  GO TO 1102
00141         WRITE (6,901)
00143         STOP
00144    1102 CONTINUE
00144   C READ INPUT FOR DYNAMICS MODEL
00145         WRITE (6,903)
00147         READ (5,930) NNODE,NCONN,NITER,LPRTF,NCHEK,PCHEK,PSST,CD
00161         WRITE (6,904) NNODE,NCONN,NITER,NCHEK,PCHEK
00161   C SKIP DYNAMICS INPUT RELATING TO COMBUSTION IF NCOMBU EQUALS 0
00170         IF (NBLFG.EQ.0)  GO TO 999
00172         READ (5,900) NBLOW
00200     999 IF (NCOMBU.EQ.0)  GO TO 103
00203         READ (5,905) NCOMCL,DPCOMC
00207         WRITE (6,906) NCOMCL,DPCOMC,CD
00214         READ (5,900) NOINJ
00222         DO 104 I=1,24
00225         IF (NOINJ(I).EQ.0)  GO TO 105
00227     104 CONTINUE
00231     900 FORMAT (24I3)
00232     901 FORMAT ('0',6X,'INJECTION MODEL MUST BE USED')
00233     930 FORMAT (5I4,3F10.2)
00234     903 FORMAT ('1',52X,'* * * DYNAMICS INPUT * * *')
00235     904 FORMAT (//7X,'NNODE   NCONN   NITER   NCHEK  PCHEK',//2X4I8,F10.3)
00236     905 FORMAT (1I4,1F10.0)
00237     906 FORMAT (//6X,'NCOMCL  DPCOMC        CD',//2X1I8,2F10.3)
00240     907 FORMAT (//7X,'NOINJ'//)
00241     908 FORMAT (2X1I8)
00242     909 FORMAT (2I3,4E10.0)
00243     910 FORMAT (//7X,'NVOL   ICOMB    VOL       PRES       TEMP       RMIX',
00243        1//)
00244     911 FORMAT (2X2I8,4F10.3)
00245     912 FORMAT (3I3,6F10.0)
00246     913 FORMAT (//7X,'IADMIT  IADMIT  IRTYPE  CAREA   CCOEFF    TIMON
00246        1 TIMOFF    TIMOPN    TIMCLS')
00247     914 FORMAT (2X3I8,2F10.3,4F10.4)
00250     915 FORMAT (7F10.0)
00251     916 FORMAT (//5X,'SPTIME    SPKTL     SPKF')
00252     917 FORMAT (//6X,'SPGAP     SPARKP    SPARKE     DC')
00253     918 FORMAT (2X,7F10.4)
00254    9918 FORMAT (2X,F10.4,F10.1,2F10.4)
00255     919 FORMAT ('1',51X,'* * * COMBUSTION INPUT * * *')
00256     105 NNOINJ=I-1
00257         WRITE (6,907)
00261         DO 106 I=1,NNOINJ
```

```
00264          WRITE (6,908) NOINJ(I)
00267      106 CONTINUE
00271      103 CONTINUE
00271     C READ VOLUME DATA
00272          DO 107 I=1,NNODE
00275          READ (5,909) NVOL(I),ICOMB(I),VOL(I),PRES(I),TEMP(I),RMIX(I)
00305      107 CONTINUE
00307          WRITE (6,910)
00311          DO 180 I=1,NNODE
00314          WRITE (6,911) NVOL(I),ICOMB(I),VOL(I),PRES(I),TEMP(I),RMIX(I)
00324      180 CONTINUE
00326          WRITE (6,913)
00330          DO 1109 I=1,NCONN
00333          READ (5,912) IADMIT(I,1),IADMIT(I,2),IRTYPE(I),CAREA(I),CCOEFF(I),
00333         1TIMON(I),TIMOFF(I),TIMOPN(I),TIMCLS(I)
00346          WRITE (6,914) IADMIT(I,1),IADMIT(I,2),IRTYPE(I),CAREA(I),CCOEFF(I)
00346         1,TIMON(I),TIMOFF(I),TIMOPN(I),TIMCLS(I)
00361     1109 CONTINUE
00363          TFSTST=TIMOFF(1)
00364          VACLST=TIMOFF(1)+TIMCLS(1)
00365          DO 777 I=2,NCONN
00370          IF (TFSTST.LT. .00001)   TFSTST=10000.
00372          TFSTST=AMIN1(TIMOFF(I),TFSTST)
00373          VACLST=AMAX1((TIMOFF(I)+TIMCLS(I)),VACLST)
00374      777 CONTINUE
00376          IF (TFSTST.GT. 9999.)  TFSTST=0.0
00400          DO 778 I=1,NCONN
00403          IF (IRTYPE(I).EQ.1)  GO TO 779
00405          CAREA(I)=CAREA(I)*CCOEFF(I)
00406      779 CONTINUE
00407      778 CONTINUE
00407     C READ SPARK IGNITION PARAMETERS
00411          READ (5,915) SPTIME,SPKTL,SPKF
00416          WRITE (6,916)
00420          WRITE (6,918) SPTIME,SPKTL,SPKF
00425          READ (5,915) SPGAP,SPARKP,SPARKE,DC
00433          WRITE (6,917)
00435          WRITE (6,5918) SPGAP,SPARKP,SPARKE,DC
00435     C
00435     C END OF DYNAMICS INPUT
00435     C
00443     1101 CONTINUE
00444          IF (NCOMBL.EQ.0) GO TO 300
00444     C READ INPUT FOR COMBUSTION MODEL
00446          WRITE (6,919)
00450          LW=1
00451          LC=2
00452          LN=3
00453          NHTO=3
00454          NHT=4
00455          NOT=5
00456          NNT=9
00457          NSPC=9
00460          NGAS=NSPC
00461          NFL=3
00462          MF=200
00463          NREAC=17
00464          NCL=NHT
00465          RG=8314.3
00466          AKA=1.
```

```
00467          PYE=3.141592
00467    C
00470        4 IFINIS=0
00471          ICUT=0
00472          IPAGE=
00473          DX=.0
00474          ICHEM=0
00475          NEW=0
00476          DO 33 I=1,NREAC
00501       33 DAMP(I)=1.
00501    C
00501    C     INPUTS
00501    C
00503          READ (5,333)  TITLE
00511      333 FORMAT (18A4)
00512          GO TO 24
00513       23 STOP
00514       24 CONTINUE
00514    C     TITLE=I.D. CARD  ANY 72 CHARACTERS
00515          WRITE (6,6)  TITLE
00523        6 FORMAT (//18A4)
00523    C
00524          READ(5,100)MPSI,IPRESS,ITURB,LW,NTYPE,ISOBAT,MB,ICHEM,MC,MG,LZ,MA
00524         1,MY,MH,ISEATY,NSLOT
00546      100 FORMAT(16I5)
00546    C     MPSI=INITIAL NUMBER OF INPUT POINTS IN PSI COORDINATES
00547          MP=MPSI+1
00550          IF(MP.GT.MF)MP=MF
00552          MQ=MP+1
00553          IF(MQ.GT.MF)MQ=MF
00555          MR=2*MPSI
00556          IF(MR.GT.MF)MR=MF
00560          NPSI=MPSI-1
00561          LR=MPSI
00561    C     MAX.NUMBER OF GRID PTS.=2*IPRESS-1
00561    C     ITURB CHOOSES THE TURBULENT VISCOSITY MODEL
00561    C     ITURB=1 - HIRSCH ROCKET CHAMBER MODEL
00561    C     ITURB=2 - VISCOSITY IS AN INPUTTED CONSTANT
00561    C     LW CHOOSE THE LAMINAR VISCOSITY MODEL FOR LOCAL PROPERTIES
00562          IF(LW.LT.1)LW=1
00562    C     NTYPE=1-AXISYMMETRIC, NOT=0-2-DIMENSIONAL
00564          NP=1-NTYPE
00565          ANP=NP
00565    C     ISOBAT=1-ISOENERGETIC WALL 2-TW(X) 3-TRANSPIRATION 4-SLOT COOLING
00565    C     MB=0-FREE JET  1=P(X) WALL  2=RW(X) WALL
00565    C     ICHEM=1-FROZEN FLOW   =2-EQUILIBRIUM CHEM  =3-FINITE RATE CHEM
00565    C     PRINTOUT AT LEAST EVERY MC STEPS
00566          IF(MC.LE.0)MC=10
00566    C     MG=1-N2 DILUENT  =2-HE  =3-AR
00570          IF(MG.LE.0)MG=1
00570    C     LZ IS PRINTOUT DUMP FOR MIXING
00570    C     MA IS PRINTOUT DUMP FOR EQUILIBRIUM CHEMISTRY AND THERMO DATA
00570    C     MY IS PRINTOUT DUMP FOR FINITE RATE CHEMISTRY AND THERMO DATA
00570    C     MH IS PRINTOUT DUMP FOR REACTION RATES
00570    C     ISBATY DETERMINES ALTERNATIVES FOR IN WALL COOLING MODELS
00570    C     NSLOT=TOTAL NUMBER OF SLOTS
00572          WRITE(6,100)MPSI,IPRESS,ITURB,LW,NTYPE,ISOBAT,MB,ICHEM,MC,MG,LZ,MA
00572         1,MY,MH,ISEATY,NSLOT
00572    C
00572    C
```

```
00614    213.          READ(5,1000)PRNT,XMAX,X
00621    214.     1000 FORMAT(7E10.8)
00621    215.     C     PRNT IS PRINTOUT INTERVAL( M)
00621    216.     C     XMAX IS FINAL X( M)
00621    217.     C     X IS INITIAL X( M)
00622    218.          WRITE(6,1)PRNT,XMAX,X
00627    219.        1 FORMAT(7E17.8)
00630    220.          PRNT=PRNT*.0254
00631    221.          XMAX=XMAX*.0254
00632    222.          X=X*.0254
00632    223.     C
00632    224.     C
00633    225.          READ (5,1000)XLE(1),SIGMA(1),DELPSI,XMPS,XK2,PSI(1)
00633    226.     C     XLE = LEWIS NUMBER
00633    227.     C     SIGMA= PRANDTL NUMBER
00633    228.     C     DELPSI= SPACING IN PSI DIRECTION
00633    229.     C     XMPS=MAXIMUM DELTA X/(XMPS)
00633    230.     C     XK2 IS DIMENSIONLESS COEFFICIENT IN TURBULENT VISCOSITY MODELS
00633    231.     C     PSI(1)=INITIAL VALUE OF PSI
00643    232.          WRITE (6,1)XLE(1),SIGMA(1),DELPSI,XMPS,XK2,PSI(1)
00653    233.          IF (NTYPE)701,701,702
00656    234.      701 DELPSI=DELPSI*.673492
00657    235.          PSI(1)=PSI(1)*.673492
00660    236.          GO TO 703
00661    237.      702 DELPSI=DELPSI*.453592
00662    238.          PSI(1)=PSI(1)*.453592
00663    239.      703 CONTINUE
00663    240.     C INPUT UNITS FOR XK2 FOR ITURB=2 ARE LBM/IN SEC
00663    241.     C CONVERSION BELOW TO NEWTON SEC/ METER**2
00664    242.          IF (ITURB.EQ.1)  GO TO 7313
00666    243.          XK2=XK2*17.858
00667    244.     7313 CONTINUE
00670    245.          IF(PSI(1).LT..0)GO TO 23
00670    246.     C
00672    247.          CALL BEGIN
00672    248.     C
00673    249.          READ(5,1000)XK
00673    250.     C     XK(1) TO XK(3) USED IN VISC MODELS
00673    251.     C     XK(5)=MEAN MAX.NO.OF CHEM.STEPS/DIFFUSION STEP
00673    252.     C     XK(6)=CF/2
00673    253.     C     XK(7)=D FOR X/D PRINTOUT
00701 *DIAGNOSTIC*     THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00701    254.          IF(XK(5).EQ.0.0)  XK(5)=10.
00703    255.          IF(XK(7).LE..0)XK(7)=1.
00705    256.          WRITE(6,1)XK
00713    257.          XK(3)=XK(3)*.0254
00714    258.          XK(7)=XK(7)*.0254
00714    259.     C
00715    260.          READ(5,1000)TAR,(XP(J),J=1,6)
00715    261.     C     USED WITH PRESSURE POLYNOMIALS
00724    262.          XP(4)=2.*XMAX
00725    263.          IY=1
00726    264.          IF(XP(5).LE..0)XP(5)=1.E-5
00726    265.     C     XP(6) IS LOWER CHEMISTRY T CHANGE TOLERANCE
00730    266.          IF(XP(6).LE..0)XP(6)=5.E-3
00732    267.          WRITE(6,503)
00734    268.      503 FORMAT(43X,33HINITIAL YW OR P,POLYNOMIAL LIMITS)
00735    269.          WRITE(6,1)TAR,(XP(J),J=1,6)
00744    270.          IF (MB-1)704,704,705
00747    271.      704 TAR=TAR*.0254
```

UNIVERSITY COMPUTING COMPANY

```
00750    272*          GO TO 706
00751    273*      705 TAR=TAR*6894.757
00752    274*      706 DO 707 I=1,8
00755    275*          XP(I)=XP(I)*.0254
00756    276*      707 CONTINUE
00756    277*   C
00760    278*          READ(5,1000)CGP
00760    279*   C       FOUR CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
00766    280*          WRITE(6,152)
00770    281*      152 FORMAT(40X,31HWALL RADIUS(INCHES) POLYNOMIALS    )
00770    282*   C      POLYNOMIALS FOR PRESSURE OR WALL RADIUS
00771    283*          WRITE(6,1)CGP
00777    284*          IF (MB-1)708,708,711
01002    285*      708 DO 709 J=1,4
01005    286*          CGP(7,J)=CGP(7,J)*.0254
01006    287*          DO 710 I=1,6
01011    288*      710 CGP(I,J)=CGP(I,J)*6894.757/(.0254**(I-1))
01013    289*      709 CONTINUE
01015    290*          GO TO 3713
01016    291*      711 DO 712 J=1,4
01021    292*          CGP(7,J)=CGP(7,J)*.0254
01022    293*          DO 713 I=1,6
01025    294*      713 CGP(I,J)=CGP(I,J)*.0254/(.0254**(I-1))
01027    295*      712 CONTINUE
01031    296*     3713 CONTINUE
01031    297*   C
01031    298*   C      READ IN WALL BOUNDARY CONDITION INPUTS
01032    299*          IF(MB.EQ.0)GO TO 50
01034    300*          GO TO(51,52,53,54),ISOBAT
01035    301*       51 CONTINUE
01035    302*   C      ISOENERGETIC WALL -IMPERMEABLE WALL
01036    303*          GO TO 50
01037    304*       52 CONTINUE
01037    305*   C      EXTERNAL REGENERATIVE COOLING - IMPERMEABLE WALL
01040    306*          IF(ISBATY.GT.1)GO TO 521
01042    307*          READ(5,1000)TWX
01042    308*   C       FOUR CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
01050    309*          WRITE(6,153)
01052    310*      153 FORMAT(40X,44HREGERATIVE COOLING - WALL TEMP(R) PROFILES    )
01053    311*            WRITE(6,1)TWX
01061    312*          DO 714 J=1,4
01064    313*          TWX(7,J)=.0254*TWX(7,J)
01065    314*          DO 715 I=1,6
01070    315*      715 TWX(I,J)=TWX(I,J)*.55556/(.0254**(I-1))
01072    316*      714 CONTINUE
01074    317*          GO TO 50
01075    318*      521 CONTINUE
01076    319*          READ(5,1000)QLX
01076    320*   C       FOUR CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
01104    321*          WRITE(6,154)
01106    322*      154 FORMAT(40X,38HREGEN. WALL COOLING HEAT TRANSFER RATE      )
01107    323*            WRITE(6,1)QLX
01115    324*          DO 716 J=1,4
01120    325*          QLX(7,J)=.0254*QLX(7,J)
01121    326*          DO 717 I=1,6
01124    327*      717 QLX(I,J)=QLX(I,J)*1.635317E+6/(.0254**(I-1))
01126    328*      716 CONTINUE
01130    329*          GO TO 50
01130    330*   C      TRANSPIRATION COOLING AT WALL
01131    331*       53 CONTINUE
```

```
01132   332.          IF(ISBATY.GT.2)GO TO 533
01134   333.          READ(5,1000)RUCX,TCX
01134   334.   C      EIGHT CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
01146   335.          WRITE(6,155)
01150   336.      155 FORMAT(4 X,45HCOOLANT MASS FLUX AND TEMPERATURE POLYNOMIALS      )
01151   337.          WRITE(6,1)RUCX
01157   338.          WRITE(6,1)TCX
01165   339.          DO 718 J=1,4
01170   340.          RUCX(7,J)=.0254*RUCX(7,J)
01171   341.          TCX(7,J)=.0254*TCX(7,J)
01172   342.          DO 719 I=1,6
01175   343.          RUCX(I,J)=RUCX(I,J)*703.07/(.0254**(I-1))
01176   344.      719 TCX(I,J)=TCX(I,J)*.55556/(.0254**(I-1))
01200   345.      718 CONTINUE
01202   346.          GO TO 50
01203   347.      533 CONTINUE
01204   348.          READ(5,1000)TWX,TCX
01204   349.   C      EIGHT CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
01216   350.          WRITE(6,156)
01220   351.      156 FORMAT(4 X,42HWALL AND COOLANT TEMPERAT, (R) POLYNOMIALS    )
01221   352.          WRITE(6,1)TWX
01227   353.          WRITE(6,1)TCX
01235   354.          DO 721 J=1,4
01240   355.          TCX(7,J)=.0254*TCX(7,J)
01241   356.          TWX(7,J)=.0254*TWX(7,J)
01242   357.          DO 722 I=1,6
01245   358.          TWX(I,J)=TWX(I,J)*.55556/(.0254**(I-1))
01246   359.      722 TCX(I,J)=TCX(I,J)*.55556/(.0254**(I-1))
01250   360.      721 CONTINUE
01252   361.          GO TO 50
01252   362.   C      SLOT COOLING AT WALL
01253   363.       54 CONTINUE
01254   364.          READ(5,100)NSLOT
01257   365.          IF(NSLOT.GT.21)NSLOT=21
01261   366.          WRITE(6,100)NSLOT
01264   367.          WRITE(6,157)
01266   368.      157 FORMAT(4 X,45HSLOT X(IN),H(IN),U(IN/S),RHO-U(LB/IN2-S),T(R))
01267   369.          DO 541 K=1,NSLOT
01272   370.          READ(5,1000)XS(K),SH(K),UC(K),RUCF(K),TCS(K)
01301   371.          WRITE(6,1)XS(K),SH(K),UC(K),RUCF(K),TCS(K)
01310   372.          XS(K)=XS(K)*.0254
01311   373.          SH(K)=SH(K)*.0254
01312   374.          UC(K)=UC(K)*.0254
01313   375.          RUCF(K)=RUCF(K)*703.07
01314   376.          TCS(K)=TCS(K)*.55556
01315   377.      541 CONTINUE
01317   378.          READ(5,1000)RSTAR,DST,PRC
01324   379.          AST=PYE*.25*DST*DST
01325   380.          WRITE(6,158)
01327   381.      158 FORMAT(4 X,21HR*(M),D*(M),PR,A*(M2))
01330   382.            WRITE(6,1)RSTAR,DST,PRC,AST
01336   383.          RSTAR=.0254*RSTAR
01337   384.          DST=.0254*DST
01340   385.          AST=PYE*.25*DST*DST
01341   386.       50 CONTINUE
01341   387.   C
01341   388.   C      COMPUTE PRESSURE
01342   389.          IF(MB.EQ.2)GO TO 41
01344   390.          DYWDX=.0
01345   391.          CALL PRESS(P,CPDX,X,CGP,XP,LY)
```

```
01346           IF(MB.EQ.1)GO TO 43
01350           YW=1.E+6
01351           GO TO 902
01352        43 YW=TAR
01353           GO TO 902
01354        41 P=TAR
01355           PNEW=P
01356           DPDX=.0
01357           CALL PRESS(YW,DYWDX,X,CGP,XP,LY)
01360           NRETRN=0
01361           IF (NINJE.EQ.0) GO TO 902
01361     C READ INJECTION MODEL INPUT
01363           READ (5,921) NELEM,DCHAMB,TO,TH,EMR
01372           IF (NDYNA.EQ.1) GO TO 191
01374           READ (5,915) WT,PO,FM
01401       998 CALL INJECT (NELEM,DCHAMB,TO,TH,EMR,MPSI,DELPSI)
01402           GO TO 723
01403       902 READ (5,1000)(T(I),I=1,MPSI)
01411           READ(5,1000)(L(I),I=1,MPSI)
01411     C     U= VELOCITY ARRAY
01417       723 CONTINUE
01420           WRITE(6,504)
01422       504 FORMAT(5 X,20HTEMPERATURE(R) ARRAY)
01423           WRITE (6,1)(T(I),I=1,MPSI)
01431           WRITE(6,159)
01433       159 FORMAT(4 X,24HVELOCITY PROFILES (IN/S))
01434           WRITE (6,1)(U(I),I=1,MPSI)
01442           DO 7723 I=1,MPSI
01445           U(I)=U(I)*.0254
01446           T(I)=.55556*T(I)
01447      7723 CONTINUE
01451       191 CONTINUE
01451     C                                                                          00003790
01452           IF (NRETRN.EQ.1)  GO TO 232
01454           IF(ICHEM.NE.3)GO TO 220
01456           READ(5,1000)(FIX(I),I=1,MPSI)
01456     C     CHEMISTRY TIME STEPS IN SECONDS
01464           WRITE(6,160)
01466       160 FORMAT(4 X,46HINITIAL CHEMICAL KINETIC TIME STEP PROFILES(S)    )
01467           WRITE(6,1)(FIX(I),I=1,MPSI)
01475       220 CONTINUE
01475     C                                                                          00000920
01476           IF(X.LE..0)GO TO 230
01476     C     ELAPSED TIME
01500           READ(5,1000)(TELAP(I),I=1,MPSI)
01506           WRITE(6,161)
01510       161 FORMAT(40X,37HSTREAMLINE RESIDENCE TIME PROFILES(S))
01511           WRITE(6,1)  (TELAP(I),I=1,MPSI)
01517           GO TO 232
01520       230 DO 231 I=1,MPSI
01523       231 TELAP(I)=.0
01525       232 CONTINUE
01526           IF (NINJE.EQ.1.AND.NRETRN.EQ.0)  GO TO 3121
01530           IF (NINJE.EQ.1)  GO TO 2121
01530     C
01532           DO 30 I=1,MPSI
01535        30 READ(5,1000)(YSPEC(J,I),J=1,NSPC)
01544      2121 CONTINUE
01545           WRITE(6,162)
01547       162 FORMAT(40X,29HSPECIE MASS FRACTION PROFILES    )
```

```
C1550          DO 720 I=1,MPSI
01553      720 WRITE(6,1)(YSPEC(J,I),J=1,NSPC)
C1562          WRITE(6,163)
C1564      163 FORMAT(4 X,30HELEMENT MASS FRACTION PROFILES        )
C1565          DO 31 I=1,MPSI
C1570          DO 39 K=1,NEL
C1573          ALPHA(K,I)=.0
01574          DO 39 J=1,NSPC
01577          ALPHA(K,I)=ALPHA(K,I)+GAM(K,J)*YSPEC(J,I)
C1600       39 CONTINUE
C1603       31 WRITE(6,1)(ALPHA(J,I),J=1,NEL)
01603    C
01603    C     INPUTS ARE CHECKED FOR OBVIOUS ERRORS HERE
01612          IF (P.LE. .0)  GO TO 444                                        *NEW
01614          DO 121 I=1,MPSI                                                 **-1
01617          IF(FIX(I).LE..0)FIX(I)=1.E-8
C1621          IF (T(I).LE. .0)  GO TO 444                                     *NEW
C1623          IF (U(I).LE. .0)  GO TO 444                                     *NEW
01625          DO 121 J=1,NSPC                                                 **-9
01630          IF (YSPEC(J,I).LT. .0)  GO TO 444                               *NEW
C1632      121 CONTINUE                                                        **-1
C1635     3121 CONTINUE
C1636          IF (NRETRN.EQ.1)  GO TO 690
C1640          IF (NINJE.EQ.0) GO TO 300
C1642          WRITE (6,920)
C1644          WRITE (6,922)
C1646          WRITE (6,923) NELEM,DCHAMB,TO,TH,EMR
C1655          IF (NDYNA.EQ.1)  GO TO 300
01657          WRITE (6,924)
C1661          WRITE (6,918) WT,PO,FM
C1661    C
01666      300 CONTINUE
01666    C END OF COMBUSTION MODEL INPUT
01666    C
C1667      920 FORMAT ('1',52X,'* * * INJECTION INPUT * * *')
C1670      921 FORMAT (1I3,4F10.0)
01671      922 FORMAT ('0',6X,'NELEM   DCHAMB       TO         TH          EM')
C1672      923 FORMAT (2X,1I8,1F10.3,2F10.2,1F10.3)
C1673      924 FORMAT ('0',4X,'WT          PO           FM')
01674      925 FORMAT ('0',52X,'* * * PERFORMANCE INPUT * * *')
01675      926 FORMAT ('0',6X,'RT          RE          XN        PERBEL     ENTHO      EN
01675         1THH')
C1676      927 FORMAT ('0',5X,'OFINPT    PER       TTT')
01676    C PERFORMANCE INPUT
C1677          IF (NCOMBL.EQ.0) GO TO 193
C1701          READ (5,915) RTTT,RE,XN,PERBEL,ENTHO,ENTHH
01711          WRITE (6,925)
C1713          WRITE (6,926)
C1715          WRITE (6,918) RTTT,RE,XN,PERBEL,ENTHO,ENTHH
01715    C BLANK CARD HERE IF NDYNA=1 AND SUPERSONIC FILM COOLING USED
01725          READ (5,915) OFINPT,PER,TTT
01732          WRITE (6,927)
C1734          WRITE (6,918) OFINPT,PER,TTT
01741          IF (NDYNA.EQ.1) GO TO 193
01743          CD=1.                                                           *NEW
01744          TIME=0.0                                                        **-1
C1745          PC=1.
01746      193 CONTINUE
01746    C
01746    C END OF INPUT
```

```
01746    512•    C
01746    513•    C DRIVER CALLING SEQUENCE
01747    514•          IF (NDYNA.EQ.0)  GO TO 680
01751    515•          TIME=0.0
01751    516•    C SET PRESSURE TO TEST FOR COMBUSTION PROGRAM CALL
01752    517•          IF (NCCMBL.EQ.0)  GO TO 683
01754    518•          PCOCAL=DPCOMC
01755    519•          NITCT=NCOMCL
01756    520•      683 CONTINUE
01757    521•      682 CALL DYNAM
01760    522•          IF (NCCMBL.EQ.0) GO TO 684
01762    523•          IF (L.NE.NITCT.AND.PRES(ICVOL).LT.PCOCAL)  GO TO 684
01764    524•          IF (ICOMB(ICVOL).EQ.2)  GO TO 685
01764    525•    C CALCULATE COLD FLOW PERFORMANCE
01766    526•          OFINPT=(1.-FM)/FM
01767    527•          CALL CCLD (PC,WOUTC,CD,RTTT,RE,XN,PERBEL,WM,TG,TIME,OFINPT)
01770    528•          IF (NITCT.EQ.L)  GO TO 1685
01772    529•          PCOCAL=PCOCAL+DPCOMC
01773    530•          GO TO 2685
01774    531•     1685 NITCT=NITCT+NCOMCL
01775    532•     2685 CONTINUE
01776    533•          GO TO 684
01777    534•      685 CONTINUE
01777    535•    C CALCULATE HOT FLOW PERFORMANCE
02000    536•          NRETRN=1
02001    537•          GO TO 998
02002    538•      690 CONTINUE
02003    539•          P=PSTAT*6894.7572                                                    •NEW
02004    540•          CALL COMB
02005    541•          OFINPT=(1.-FM)/FM
02006    542•          EINPUT=(FM*ENTHO)+((1.-FM)*ENTHH)
02007    543•          CALL PERF(RTTT,RE,XN,PERBEL,PTBL,H,OF,RU,Y,RUT,WMAL,TTBL,MP8I,
02007    544•         1CD,TIME,PRES(ICVOL),OFINPT,EINPUT)                                   •NEW
02010    545•          IF (NITCT.EQ.L)  GO TO 1684                                          •••I
02012    546•          PCOCAL=PCOCAL+DPCOMC
02013    547•          GO TO 2684
02014    548•     1684 NITCT=NITCT+NCOMCL
02015    549•     2684 CONTINUE
02016    550•      684 CONTINUE
02017    551•          IF (L.EQ.NITER)  GO TO 686
02017    552•    C ARE VALVES SHUTTING?
02021    •DIAGNOSTIC•  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
02021    553•          IF (TFSTST.EQ.0.0.OR.TIME.LT.TFSTST)  GO TO 687
02023    554•          IF (PRES(ICVOL).LT.1.0)  GO TO 686
02025    555•          GO TO 688
02026    556•      687 CONTINUE
02026    557•    C MAKE SHUTDOWN TESTS FOR RISING COMBUSTOR PRESSURE
02027    558•          IF (PCHEK.GE.DPRES.AND.PRES(NCHEK).GE.PSST)  GO TO 688
02031    559•          GO TO 689
02032    560•      688 CONTINUE
02033    •DIAGNOSTIC•  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
02033    561•          IF (TFSTST.EQ.0.0)  GO TO 686
02035    562•          IF (TIME.GE.TFSTST)  GO TO 689
02037    563•          TIME=TFSTST+.00005
02037    564•    C RETURN TO DYNAMICS PROGRAM
02040    565•      689 CONTINUE
02041    •DIAGNOSTIC•  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
02041    566•          IF (TIME.LE.VACLST.OR.VACLST.EQ. 0.0)  GO TO 682
02043    567•          IF (NBLFG.EQ.0)  GO TO 680
02045    568•          CALL BLOW (NBLOW)
```

```
C2046    569*           GO TO 686
C2047    570*       680 CONTINUE
C2050    571*           IF (NCCMBL.EQ.0)  GO TO 686
C2050    572*     C COMBUSTION CALCULATIONS FOR CASE WITHOUT DYNAMICS
C2052    573*           CALL CCMB
C2053    574*           FM=OFINPT/(1.+OFINPT)
C2054    575*           EINPUT=(FM*ENTHO)+((1.-FM)*ENTHH)
C2055    576*           CALL PERF(RTTT,RE,XN,PERBEL,PTBL,H,CF,RU,Y,RUT,WMPL,TTBL,MPSI,
C2055    577*          1CD,TIME,PC,OFINPT,EINPUT)                                          *NEW
C2056    578*       686 CONTINUE                                                           ***I
C2056    579*     C DYNAMICS AND PERFORMANCE CALCULATIONS COMPLETE OR BYPASSED
C2057    580*           IF (NHEAT.EQ.0)  GO TO 691
C2061    581*           CALL HEATT
C2062    582*       691 CONTINUE
C2063    583*           IF (ICAS.LT.NCAS)  GO TO 751
C2065    584*       444 CONTINUE                                                           *NEW
C2066    585*           STOP
C2067    586*           END

          END OF UCC 1108 FORTRAN V COMPILATION.     4 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME    00:01.194
PHASE 2 TIME    00:00.085
PHASE 3 TIME    00:02.685
PHASE 4 TIME    00:00.120
PHASE 5 TIME    00:01.541
PHASE 6 TIME    00:01.517

TOTAL COMPILATION TIME =    00:07.148
    MAIN              SYMBOLIC                               (FN5)   23 JUN 72  09:16:11   0  00375226    14  584  (DELETED)
    MAIN     CODE     RELOCATABLE                            (FN5)   23 JUN 72  09:16:11   1  00415206    84    1  (DELETED)
                                                                                           0  00415332    14  244
```

UCC
UNIVERSITY COMPUTING COMPANY

```
• ELT AREA,1,720512, 51591    , 1

                  SUBROUTINE AREA
                  COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
                 1,W(9,200),SIGMA(1),XLE(1),RUR(200),CPBAR(200),XMU(200),U(200)
                 2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLW(200),H(200)
                 3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
                 4,FEF(200)
                  COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
                 1,GSCALE,TKX(7,4)
                  COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
                 1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
                 2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
                 3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
                 4,NEW,MQ,MR,LN,NNT,JR,NSLCR
                  COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,MST
                 1OR,USTOR,RAY,RG,AK,AKA
                  COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
                  COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
                  DIMENSION RHOZ(200),YZ(200),RU(200),RW(200),PAY(4)
                  DIMENSION AP(9)
                  IF(ICUT-1)98,98,99
               98 NUMBER=0
                  LT=40
                  LU=20
                  LX=20
               99 KOUNT=LR
                  DELTA=1.
                  KAT=0
                  PFRD=1.0E+30
                  FALPA=1.
                  IND=1
                  HDPDX=DPDX
                  EPSI=1.E-6
                  RX=X+DX
                  CALL PRESS(YW,DYWDX,RX,CGP,XP,LY)
                  DO 143 I=1,LR
                  RW(I)=.0
                  DO 43 J=1,NGAS
               43 RW(I)=RW(I)+YSPEC(J,I)/WTMOLE(J)
              143 RW(I)=RG*RW(I)
               22 CONTINUE
              122 PNEW=P+DPDX*DX
                  IF(PNEW.GT.0)GO TO 35
                  PNEW=.67*P
                  DPDX=(PNEW-P)/DX
                  HDPDX=DPDX
               35 CONTINUE
                  IF(LZ.NE.0)WRITE(6,20)YW,DPDX,PNEW,P,DX,X
            C LT IS MAXIMUM NUMBER OF ITERATIONS ON DYWPDX
               25 DO 11 M=1,LT
            C COMPUTE DENSITY AT X SUB N+1
                  GO TC 51
               50 IF(LZ.NE.0)WRITE(6,3)IOUT,I,LR,MPSI,KAT
                3 FORMAT(14I5)
                  IF(LZ.NE.0)WRITE(6,20)PNEW,RU(I),RUR(I),DX,DPDX,RUT(I)
                  IF(KAT.EQ.0)PFRD=PNEW
                  IF(PNEW.LT.PFRD)PFRD=PNEW
                  PNEW=.9*PFRD
               52 DPDX=(PNEW-P)/DX
```

```
      UFAC=ABS(DX*DPDX/RUT(I))
      IF(UFAC.LT.U(I))GO TO 53
      PFAC=.1*U(I)*RUT(I)/DX
      IF(DPDX.LT..0)PFAC=-PFAC
      DPDX=PFAC
      PNEW=P+DX*DPDX
      KAT=KAT+1
      IF(KAT-10)52,52,54
   54 WRITE(6,20)PNEW,RU(I),U(I),DX,DPDX,RUT(I),UFAC,PFAC
   53 CONTINUE
      KAT=KAT+1
      IF(KAT.GT.10)GO TO 23
   51 DO 36 I=1,LR
      RU(I)=U(I)-DX*DPDX/RUT(I)
      IF(RU(I).LE..0)GO TO 50
      SMALLH(I)=H(I)-.5*RU(I)*RU(I)/AKA
      DO 236 J=1,NSPC
  236 AP(J)=YSPEC(J,I)
      CALL INVERT(RT(I),SMALLH(I),AP,CPX)
      IF(RT(I).LE.0.)GO TO 50
   36 RHOZ(I)=PNEW/RT(I)/RW(I)
C COMPUTE PHYSICAL Y COORDINATES
      IF(NTYPE.EQ.0)GO TO 12
      YZ(1)=PSI(1)/    (RHOZ(1)*RU(1))
      GO TO 42
   12 YZ(1)=PSI(1)/SQRT(RHOZ(1)*RU(1))
   42 DO 14 I=2,KOUNT
      IF(NTYPE.NE.0)GO TO 28
      YZZ=YZ(I-1)**2+DELPSI*(PSI(I)/RHOZ(I)/RU(I)+PSI(I-1)/RHOZ(I-1)/RU(
     1I-1))
      IF(YZZ.LE..0)GO TO 300
      YZ(I)=SQRT(YZZ)
      GO TO 14
   28 YZ(I)=YZ(I-1)+DELPSI/2.*(1./RHOZ(I)/RU(I)+1./RHOZ(I-1)/RU(I-1))
   14 CONTINUE
      IF(LZ.EQ.0)GO TO 13
      WRITE(6,20)RT(KOUNT),RHOZ(KOUNT),YZ(KOUNT),RU(KOUNT)
C TEST ON REACHING WALL
   13 IF(NEW.EQ.0)GO TO 4
      YWP=YZ(MPSI)
      GO TO 5
    4 IF(NTYPE.NE.0)GO TO 29
      YWP=SQRT(YZ(MPSI)**2+(PSIW**2-PSI(MPSI)**2)/(RHOZ(MPSI)*RU(MPSI)))
      GO TO 5
   29 YWP=YZ(MPSI)+(PSIW-PSI(MPSI))/RHOZ(MPSI)/RU(MPSI)
    5 ALPA=ABS(YWP/YW-1.)
      IF(ALPA.GE.FALPA)  GO TO 115
      FALPA=ALPA
      FPNEW=PNEW
      FDPDX=DPDX
      FYWP=YWP
  115 IF(ALPA.LE.EPSI)GO TO 6
   27 IF(DELTA*ALPA.LT..001)GO TO 26
      DELTA=.1*DELTA
      GO TO 27
   26 IF(M.EQ.1) GO TO 700
C     MONOTONICALLY APPROACHING CONVERGENCE
    9 IND=9
      PN=PNEW
      PAY(1)=POLD/PN-1.0
      PAY(2)=YW/YWP-1.0
      PAY(3)=YL/YWP-1.0
      PAY(4)=1.+PAY(1)*PAY(2)/PAY(3)
   41 IF(PAY(4).GT.1.2)PAY(4)=1.2
      IF(PAY(4).LT..8)PAY(4)=.8
   40 PNEW=PN*PAY(4)
      IF(PNEW.LT.PFRD)GO TO 30
      IND=11
      PNEW=.9*PFRD
   30 CONTINUE
      BETA=ABS(PN/PNEW-1.)
      IF(BETA.GT.1.E-7)GO TO 32
      IF(ALPA.GT..001) GO TO 33
      DPDX=(PNEW-P)/DX
      GO TO 6
   32 POLD=PN
      GO TO 7
C     FIRST TIME THRU
  700 POLD=PNEW
      PNEW=POLD*(1.+DELTA*ALPA)
    7 DPDX=(PNEW-P)/DX
      IF(PNEW.LT.0.0)GO TO 300
      IF(LZ.EQ.0)GO TO 210
      WRITE(6,21)IND
      WRITE(6,20)ALPA,POLD,YWP,DPDX,PAY
  210 YL=YWP
   11 CONTINUE
   23 IF(IFINIS-2)24,160,160
   24 WRITE(6,15)LT
   15 FORMAT(1H1,40X,20HYWP DOESNT CONVERGE-,1I5,11H ITERATIONS)
      IFINIS=2
      IF(LZ.NE.0)  GO TO 31
      DPDX=HDPDX
      LZ=666
      GO TO 99
   33 WRITE(6,34)
   34 FORMAT(/40X,24HGETTING READY TO BLOW UP/)
      LZ=666
  300 DPDX=HDPDX
      DELTA=DELTA/10.
      IND=1
      IF(LZ.NE.0)WRITE(6,301)DELTA,PNEW,RU(1),RT(1)
  301 FORMAT(//29H BAD ITERATION--DELTA CUT TO 1P5E15.7)
      NUMBER=NUMBER+M
      KAT=KAT+1
      IF(KAT-10)22,22,23
    6 CONTINUE
C     YWP HAS CONVERGED-RETURN TO SBR TWO
      NUMBER=NUMBER+M
      IF(LZ.EQ.0) GO TO 150
      WRITE(6,20)DPDX,PNEW,ALPA,YW,YWP
C     TESTING OF DELTA 2 AND DELTA 3
  150 IF(IOUT.LE.LX)GO TO 160
      IF(NUMBER.LE.LU*IOUT)GO TO 160
      IFINIS=2
      WRITE(6,21)LX,LU,LT,NUMBER,IOUT
   21 FORMAT(1H0,10X,6I10)
      WRITE(6,20)DPDX,PNEW,ALPA,YW,YWP
   20 FORMAT(8E15.7)
      WRITE(6,20)YW,DYWDX,PNEW,P,DX,X
  160 IF(LZ.EG.666)LZ=0
      RETURN
   31 PNEW=FPNEW
      DPDX=FDPDX
      YWP=FYWP
      WRITE(6,3)NEW,LR,KOUNT,MPSI,NPSI,IOUT,KAT,MINIT,MHALF
      GO TO 6
      END
```

```
 ● ELT BEGIN,1,720512, 51555     , 1

        SUBROUTINE BEGIN
        COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
       1,GSCALE,TWX(7,4)
        COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
       1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
       2,ISHATY,MJ,MK,ML,MM,MN,MO,NSLOT
       3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
       4,NEW,MQ,MR,LN,NNT,JR,NSLCR
        COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
        DIMENSICN HFA(270),HFD(90),GAMD(9),WTD(3),        DHF(9),WTED(3)
C       SPECIES ARE 1-H 2-O 3-H2O 4-H2 5-O2 6-OH 7-HO2 8-H2O2 9-DILUENT
      1 FORMAT(7E17.8)
     11 FORMAT(7E10.6)
        CNFSMS=,0929
        CNKCJL=4.184E6
        DATA GAN/1.,C.,0., 0.,1.,0., 2.,1.,0., 2.,0.,0., 0.,2.,0.,
       1          1.,1.,0., 1.,2.,0., 2.,2.,0., 0.,0.,2./
        DATA GAMD/0.,0.,2., 0.,0.,1., 0.,0.,1./
C
C       MOLECULAR WEIGHTS
        DATA WTMOLE/1.008,16.,18.016,2.016,32.,17.008
       1,33.006,34.016,28.016/
        DATA WTD/28.016,4.003,39.94/
        WTE(1)=1.008
        WTE(2)=16.
        DATA WTED/14.008,4.003,39.94/
C
        DATA DHF/51.632,58,989,-57.103,0.,0.,9,273
       1,5,697,-31,025,0./
C
        DATA(HFA(I),I=1,90)/
       19000.,2,3066E09,2,2115E05,1,4553E-01,-3,7024E04,              H
       110250.,3.6373E09,-8.5672E04,1,7191E01,2,4497E06,              H
       111250.,7.3904E09,-8,0824E05,5,2438E01,8,4001E06,              H
       112000.,1.5228E10,-2.2017E06,1,1437E02,2,0005E07,              H
       113250.,2.6151E10,-4,0221E06,1,9022E02,3,5283E07,              H
       115000.,4.3323E10,-6.6141E06,2,8803E02,5,7294E07,              H
       1500.,1.6509E06,1.6188E04,-2.0676E00,1,7256E04,                O
       12500.,1.6640E08,1,4154E04,-3.3621E-02,2,7862E04,              O
       114750.,1.6732E08,1,3418E04,1.1355E-01,3,2884E04,              O
       117750.,2.5559E08,1,4407E03,5.1930E-01,1,3581E05,              O
       126000.,4.3567E08,-1.8841E04,1,0908E00,3,1404E05,              O
       130000.,8.7230E07,7,9617E03,5,7541E-01,6,8365E04,              O
       1300.,-1,4770E08,1,9782E04,3.4739E-01,-2,2776E02,              H2O
       11700.,-1.4249E08,1,8027E04,3.2716E00,8,0253E03,               H2O
       12800.,-1.4772E08,2.4170E04,1.4649E00,-3,1525E04,
       14000.,-1.5619E08,3,0223E04,3,8408E-01,-7.3514E04,             H2O
       16400.,-1,5866E08,3,1459E04,2.2950E-01,-8,2535E04,             H2O
       115000.,-1,6748E08,3.4216E04,1,4163E-02,-1.0393E05/
        DATA(HFA(I),I=91,180)/
       1800.,3.8299E04,1.5492E05,1,6710E00,-1,8684E05,                H2
       11800.,5.6323E06,1.4093E05,1,0411E01,-1,0734E05,               H2
       12800.,-2.6248E04,1,4722E05,8.6652E00,-1,4818E05,              H2
       14300.,-3.9467E07,1,7539E05,3,6345E00,-3,4362E05,              H2
       19000.,-6.2866E07,1,8628E05,2,3690E00,-4,2379E05,              H2
       115000.,-9,8419E07,1,9418E05,1,9301E00,-4,8783E05,             H2
       11250.,3.0346E04,9,3717E03,1,1542E00,1,4875E04,                O2
       14500.,-1.1501E06,1,1342E04,3,3306E-01,2,8742E03,              O2
```

```
          18800.,-3.8614E06,1.2575E04,1.9294F-01,-6.2367F03,                        02
          112250.,-1.9227E07,1.6108E04,-1.0088E-02,-3.4750E04,                      02
          116000.,-2.8490E07,1.7653E04,-7.4441E-02,-4.7711E04,                      02
          130000.,-4.8845E07,2.0238E04,-1.5651E-01,-7.0111E04,
          1600.,2.4630E07,1.8391E04,1.0495E-01,1.1638E04,                           OH
          11700.,2.5033E07,1.7045E04,1.2264E00,1.8701E04,                           OH
          12600.,2.4436E07,1.7748E04,1.0196E00,1.4374E04,                           OH
          13800.,1.9712E07,2.1381E04,3.2092E-01,-1.0562E04,                         OH
          110000.,1.8694E07,2.1917E04,2.5043E-01,-1.4442E04,                        OH
          115000.,1.3507E07,2.2955E04,1.9856E-01,-2.2961E04/
           DATA(HFA(I),I=181,240)/
          1    100C.,     7.8369E6, 9890.8,   2.9545,   16210.,                     HO2-1
          2    200C.,     5.9836E6, 13681.,   1.0277,   -6126.,                     HO2-2
          3    300C.,     3.4835E6, 16303.,   .34163,   -23310.,                    HO2-3
          4    400C.,     1.7952E6, 17461.,   .14308,   -31360.,                    HO2-4
          5    500C.,     6.9213E5, 18025.,   .071122,  -35440.,                    HO2-5
          6    600C.,     -19776.,  18315.,   .041428,  -37760.,                    HO2-6
          1    500.,      -4.1019E7,9154.,    7.3542,   16339.,                     H2O2-1
          2    100C.,     -4.2176E7,13584.,   3.2636,   -6030.,                     H2O2-2
          3    150C.,     -4.3447E7,16337.,   1.7967,   -22070.,                    H2O2-3
          4    3000C.,    -4.5962E7,20649.,   0.,       -48220.,                    H2O2-4
          4    3000C.,    -4.5962E7,20649.,   0.,       -48220.,                    H2O2-4
          4    3000C.,    -4.5962E7,20649.,   0.,       -48220./
           DATA HFC/
          11800.,3.6566E04,1.0730E04,8.6656E-01,1.1777E04,                          N2
          13700.,-2.3211E06,1.3349E04,1.3888E-01,-5.2383E03,                        N2
          18500.,-3.2845F06,1.3870E04,6.8507E-02,-8.9963E03,                        N2
          110500.,2.9248E07,6.2157E03,5.1879F-01,5.2605E04,                         N2
          116000.,4.6082E07,2.6283E03,6.8962F-01,8.2233E04,                         N2
          118000.,-3.2656E07,1.2720E04,3.7423E-01,-5.3676E03,                       N2
          1    3000C.,    0.,       55934.,   0.,       20484.,                     HE
          1    3000C.,    0.,       55934.,   0.,       20484.,                     HE
          1    3000C.,    0.,       55934.,   0.,       20484.,                     HE
          1    3000C.,    0.,       55934.,   0.,       20484.,                     HE
          1    3000C.,    0.,       55934.,   0.,       20484.,                     HE
          1    3000C.,    0.,       55934.,   0.,       20484.,                     HE
          1    1100C.,    917.53,   5592.6,   .0013637, 9826.2,                     AR-1
          2    1225C.,    2.1278E7, 1724.,    .1772,    41957.,                     AR-2
          3    1350C.,    3.9705E7, -1284.5,  .3,       67269.,                     AR-3
          4    1450C.,    1.1884E8, -13009.,  .73426,   167050.,                    AR-4
          5    1575C.,    2.0493E8, -24883.,  1.1437,   268950.,                    AR-5
          6    1750C.,    4.1154E8, -51119.,  1.9766,   496280.,   /                AR-6
      C
        12 FORMAT(5E10.6)
         2 FORMAT(5E15.7)
           WTE(LM)=WTED(MG)
           WTMOLE(KNT)=WTD(MG)
           DO 20 M=1,30
           L=240+M
           K=M+(MG-1)*30
        20 HFA(L)=HFD(K)
           DO 21 L=1,3
           K=L+(MG-1)*3
           M=24+L
        21 GAN(M)=GAMD(K)
           DO 49 I=1,NEL
           DO 49 J=1,NSPC
           K=I+NEL*(J-1)
           GAM(I,J)=GAN(K)*WTE(I)/WTMOLE(J)
        49 CONTINUE
```

```
C
      DO 50 J=1,NSPC
C    CONVERT DATA FROM KCAL/MOLE TO M2/SEC2
      DEL(J)=CNKCJL*DHF(J)/WTMOLE(J)
      DO 50 K=1,6
      I=1+5*(K-1)+30*(J-1)
      HF(1,K,J)=HFA(I)
      DO 50 L=2,5
      I=L+5*(K-1)+30*(J-1)
C   CONVERT DATA FROM FT2/SEC2 TO M2/SEC2
      HF(L,K,J)=CNFSMS*HFA(I)
   50 CONTINUE
      IF(MA.GT.1)WRITE(6,2)HF
      RETURN
      END
```

```
• ELT BL,1,720512, 51650 , 1

              SUBROUTINE BL(RE,RT,PT,WW,MW,TG,DF)
      C CALCULATE BOUNDARY LAYER LOSS PER CPIA 178
      C  RE= RADIUS OF EXIT  INCHES
      C  RT  RADIUS OF THROAT  INCHES
      C  PT=  CHAMBER PRESSURE  PSIA
      C  CSTAR  CHARACTERISTIC VELOCITY   FT/SEC
      C   MW   MOLECULAR WEIGHT
      C  TG  TEMPERATURE GAS   TOTOAL   DEG  R
      C  DF   BOUNDARY LAYER LOSS   LB  FORCE
      C  M  MACH NUBER
                REAL M,MW
                AR=ALOG((RE/RT)**2)
                M=1.461+.9122*AR-.06832*AR**2+.0069018*AR**3
                M=.91*M
                PRO=(1./(1.2*M**2))
                ROP=((1.2*M**2)/((1.+(.1*M**2))**6))
                VIS=46.6E-10*(TG**.6)*MW**.5
            R=(.636*WW)/(RT*VIS)
                FIGB3=.0036+.030785*M-.001786*M**2
                FIGB5=-1.18+.09405*M+.11429*M**2-.008333*M**3
                FIGB7=57.7+.6714*M-.5714*M**2
            TREF=FIGB3*RE*(1./(R**.2))*(1./(SIN((6.283/360.)*FIGB7))**.5)
                TREF=1.1*TREF
            DF=6.283*PT*RE*ROP*TREF*COS((6.283/360.)*FIGB7)*(1.*(PRO*FIGB5))
                RETURN
                END
```

• ELT BULK,1,720512, 51558   , 1

```
              SUBROUTINE BULK
              COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
             1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
             2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
             3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
             4,FEE(200)
              COMMON/ZB/OF(200),HSEN(200)
              COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
             1,GSCALE,TWX(7,4)
              COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
             1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
             2,ISUATY,MJ,MK,ML,MM,MN,MO,NSLOT
             3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
             4,NEW,MQ,MR,LN,NNT,JR,NSLCR
              COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,OPDX,XTRA,HST
             1DR,USTOR,RAY,RG,AK,AKA
              COMMON/ZF/SBL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL,UTBL,CPBL,HEBL
             1,RHOBL,RUTBL,WMBL,TBL,GMBL,SSBL,EMBL,PTBL,TTBL
              COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
              COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
C             SPECIES ARE 1-H 2-O 3-H2O 4-H2 5-O2 6-OH 7-HO2 8-H2O2 9-DILUENT
              COMMON/STORNP/RRUSE,NUSE,RPRB,RRUT,RALAW(9),RHEE,RHAW,RUE,RRHO,RCP
             1,RVB,RVS
              COMMON/ZL/QLX(7,4),          TCX(7,4),QL,TC,PRR,GAMW,RUC,RUCX(7,4)
             1,SCB,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
             2,GG,UB(21),UBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
             3,VWS(21),RWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,GCG(21)
             4,CRB,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
             5,NCL,NP,PYE,ANP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),HC,HCB(21)
            1 FORMAT(8E15.7)
              IF(NTYPE.NE.0)GO TO 154
              EMD=PYE*PSIW*PSIW
              DO 153 I=1,LR
          153 EMDT(I)=PYE*PSI(I)*PSI(I)
              GO TO 155
          154 EMD=PSIW
              DO 156 I=1,LR
          156 EMDT(I)=PSI(I)
          155 BARA=EMDT(1)/EMD
              BARB=1.-EMDT(LR)/EMD
              UBL=BARA*U(1)+BARB*U(LR)
              HBL=BARA*H(1)+BARB*H(LR)
              IF(ICHEM.EQ.2)GO TO 151
          150 DO 157 J=1,NSPC
          157 SBL(J)=BARA*YSPEC(J,1)+BARB*YSPEC(J,LR)
              GO TO 159
          151 DO 158 K=1,NEL
          158 EBL(K)=BARA*ALPHA(K,1)+BARB*ALPHA(K,LR)
          159 CONTINUE
              DO 160 I=2,LR
              IM=I-1
              BARC=.5*(EMDT(I)-EMDT(IM))/EMD
              UBL=UBL+BARC*(U(I)+U(IM))
              HBL=HBL+BARC*(H(I)+H(IM))
              IF(ICHEM.EQ.2)GO TO 164
          163 DO 161 J=1,NSPC
          161 SBL(J)=SBL(J)+BARC*(YSPEC(J,I)+YSPEC(J,IM))
              GO TO 169
```

```
  164 DO 162 K=1,NEL
  162 EBL(K)=EBL(K)+BARC*(ALPHA(K,I)+ALPHA(K,IH))
  169 CONTINUE
  160 CONTINUE
      IF(ICHEM.EQ.2)GO TO 175
      DO 171 K=1,NEL
      EBL(K)=.0
      DO 171 J=1,NSPC
  171 EBL(K)=EBL(K)+GAM(K,J)*SBL(J)
      DO 172 J=1,NSPC
  172 YSPEC(J,MF)=SBL(J)
      GO TO 179
  175 CONTINUE
      DO 176 K=1,NEL
  176 ALPHA(K,MF)=EBL(K)
      CALL EQLILC(MF,MF)
      DO 177 J=1,NSPC
  177 SBL(J)=YSPEC(J,MF)
  179 CONTINUE
      KOPT=2
      H(MF)=HEL
      U(MF)=UEL
      CALL HEAT(MF,MF)
      HEBL=SMALLH(MF)
      UTBL=TALT(MF)
      CPBL=CPBAR(MF)
      TBL=T(MF)
      CALL DENSE(MF,MF)
      RHOBL=RHO(MF)
      RUTBL=RUT(MF)
      WMBL=WTMIX(MF)
      RETURN
      END
```

```
• ELT BUMP.1.720512, 51657     , 1

                FUNCTION BUMP(L)
        C          RCUTINE TO SET CONVERGE CRITERIA IN DYNAMICS
                DIMENSICN A(5)
                DATA A/.2,.6,.8,.9,.95/
        C
                IF(L .GT. 5)GO TO 5
                BUMP=A(L)
                RETURN
        C
         5      BUMP=1.C
                RETURN
                END
```

```
• ELT CFLOW,1,720512, 5154R     , 1

            FUNCTION CFLOW(F4LD)
C                                    FUNCTION TO COMPUTE CHOKED DUCT FLOW
C                                    PARAMETER GIVEN  4FL/D
            X=ALOG(F4LD)
            X2=X*X
            X3=X*X2
C
            CFLOW=.522489027 -.97105945E-1*X -.94743323E-2*X2
           1      + .22501798E-2*X3
            RETURN
            END
```

```
• ELT COLD,1,720512, 516C5     , 1
      SUBROUTINE COLD(PC,W,CF,RT,RE,XN,PERBEL,WM,TG,TIME,OF)
      COMMON/FCP/T,PER
      REAL ISP
      CSTAR=PC*(RT**2*3.14159)*32.174/W
      A=(RE**2/RT**2)
      B=9.81
      C=.3349
      PE=-3.0756-.197097*A+.003062*A**2-1.589E-5*A**3
      PE=EXP(PE)
      D=(1.-PE)**(.286)
      CF=(B*C*D)**.5+PE*A
      F=PC*CF*(RT**2*3.14159)
      ISP=CSTAR*CF/32.174
      CALL BL(RE,RT,PC,CSTAR,WM,TG,DF)
      CALL DIV(RE,RT,DVL)
      DFISP=DF/W
      DDIV=(1.-DVL)*ISP
      DISP=ISP-DFISP-DDIV
      FDEL=DISP*W
      WRITE(6,100) TIME,DISP,FDEL,DDIV,DFISP,ISP
  100 FORMAT(1H1,30H THE ISP AND THRUST AT TIME = ,F6.3, 4H IS ,F6.2,
     111H SEC. AND  ,F7.2, 4H LBS/12H DIV LOSS = ,F6.3,5H SEC./23H BOUND
     1ARY LAYER LOSS = ,F6.3,5H SEC./12H THEO ISP = ,F6.2)
      IF(PER.LT..01) GO TO 51
      ARA=(RE/RT)**2
      ARA=ALOG(ARA)
      S=211.79+42.07*A-10.94*A**2+.94066*A**3
      S=S*(T/530.)**.5
      WWS=(PER*W )/((1.-PER)*(1.+OF))
      FF=S*WWS
      WWT=WWS+W
      FF=FF+FDEL
      DS=FF/WWT
      WRITE(6,105)FF,DS
  105 FORMAT(//58H DUE TO MASS ADDITION IN THE SUPERSCNIC REGION THE THR
     1UST /18H HAS INCREASED TO ,F7.2,4H LBS//26H THE DELIVERED ISP IS N
     20W ,F7.2,5H SEC.)
   51 CONTINUE
      RETURN
      END
```

```
• ELT COMB,1,720512, 51593      , 1

      SUBRCUTINE COMB
      COMMCN/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
      COMMCN/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DFL(9),TW
      COMMCN/ZL/QLX(7,4),              TCX(7,4),QL,TC,PRB,GAMW,RUC,RUCX(7,4)
     1,SCB,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
     2,GG,UB(21),LBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
     3,VWS(21),RWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,GGG(21)
     4,CPB,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
     5,NCL,NP,PYE,AMP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),HC,HCS(21)
      COMMCN/STORNP/RRUSE,NUSE,RPRB,RRUT,RALAW(9),RHEE,RHAW,RUE,RRHO,RCP
     1,RVB,RVS
      COMMCN/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      COMMCN/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HFLD(200),FIX(200)
     1,IDA(200),TCHEM(200),WDT(9,200)
      COMMCN/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TWX(7,4)
      COMMCN/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISRATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMCN/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,MST
     1CR,USTOF,RAY,RG,AK,AKA
      DATA LH,LO,LN,NHTO,NHT,NOT,NNT/
     1       1, 2, 3,    3,  4,  5,  9/
      NSPC=9
      NGAS=NSPC
      NEL=3
      MF=200
      NREAC=17
      NCL=NHT
      RG=8314.3
      AKA=1.
      PYE=3.141592
C
      GO TC 5151
    4 RETURN
 5151 CONTINUE
      IFINIS=C
      IOUT=0
      IPAGE=0
      DX=.0
      NEW=0
      DO 33 I=1,NREAC
   33 DAMP(I)=1.
C
      CALL DENSE(1,MPSI)
C
      KOPT=1
      CALL HEAT(1,MPSI)
C
      IF(MPSI.EQ.1)GO TO 400
C
C     INITIALIZATION
```

```
  903 DO 20 I=1,MF
      XI=I-1
      PSI(I)=XI*DELPSI+PSI(1)
      DO 20 J=1,NSPC
      WDT(J,I)=.0
      W(J,I)=.0
   20 CONTINUE
      DO 90 I=MP,MF
      TELAP(I)=TELAP(MPSI)
      FIX(I)=FIX(MPSI)
      H(I)=H(MPSI)
      DO 81 J=1,NSPC
   81 YSPEC(J,I)=YSPEC(J,MPSI)
      DO 80 J=1,NEL
   80 ALPHA(J,I)=ALPHA(J,MPSI)
      T(I)=T(MPSI)
   90 U(I)=U(MPSI)
C
      IF(NTYPE.EQ.0)GO TO 135
      Y(1)=PSI(1)/RUT(1)
      GO TO 133
  135 Y(1)=PSI(1)/SQRT(RUT(1))
  133 DO 124 I=2,MPSI
      IF(NTYPE.EQ.0)GO TO 25
      Y(I)=Y(I-1)+.5*DELPSI*(1./RUT(I)+1./RUT(I-1))
      GO TO 124
   25 Y(I)=SQRT(Y(I-1)**2+DELPSI*(PSI(I)/RUT(I)+PSI(I-1)/RUT(I-1)))
  124 CONTINUE
      IF(NTYPE.EQ.0)GO TO 122
      PSIW=PSI(MPSI)+RUT(MPSI)*(YW-Y(MPSI))
      GO TO 123
  122 PSIW=SQRT(PSI(MPSI)**2+RUT(MPSI)*(YW**2-Y(MPSI)**2))
  123 WRITE(6,1)PSIW,P,DPDX,YW,DYWDX
    1 FORMAT (7E17.8)
      PSIWA=PSIW
      DO 127 J=1,NSPC
  127 ALC(J)=.0
      ALC(NNT)=1.E-6
      ALC(NCL)=1.-ALC(NNT)
      ELC(LN)=ALC(NNT)*GAM(LN,NNT)
      ELC(LH)=1.-ELC(LN)
      ELC(LO)=.0
      LR=MPSI
      IF(PSIW.GE.PSI(MPSI))GO TO 140
      CALL VISC
      CALL WALL
  140 IF(PSIW.LT.PSI(NPSI-1))GO TO 4
C
C     CALL SUBROUTINES AND LOOP
C
    2 CONTINUE
      CALL MARCH
      GO TO 4
C
  400 CONTINUE
      IF(NTYPE.EQ.0)GO TO 235
      PSI(1)=RUT(1)*YW
      GO TO 224
  235 PSI(1)=YW*SQRT(RUT(1))
  224 PSIW=PSI(1)
```

00003080

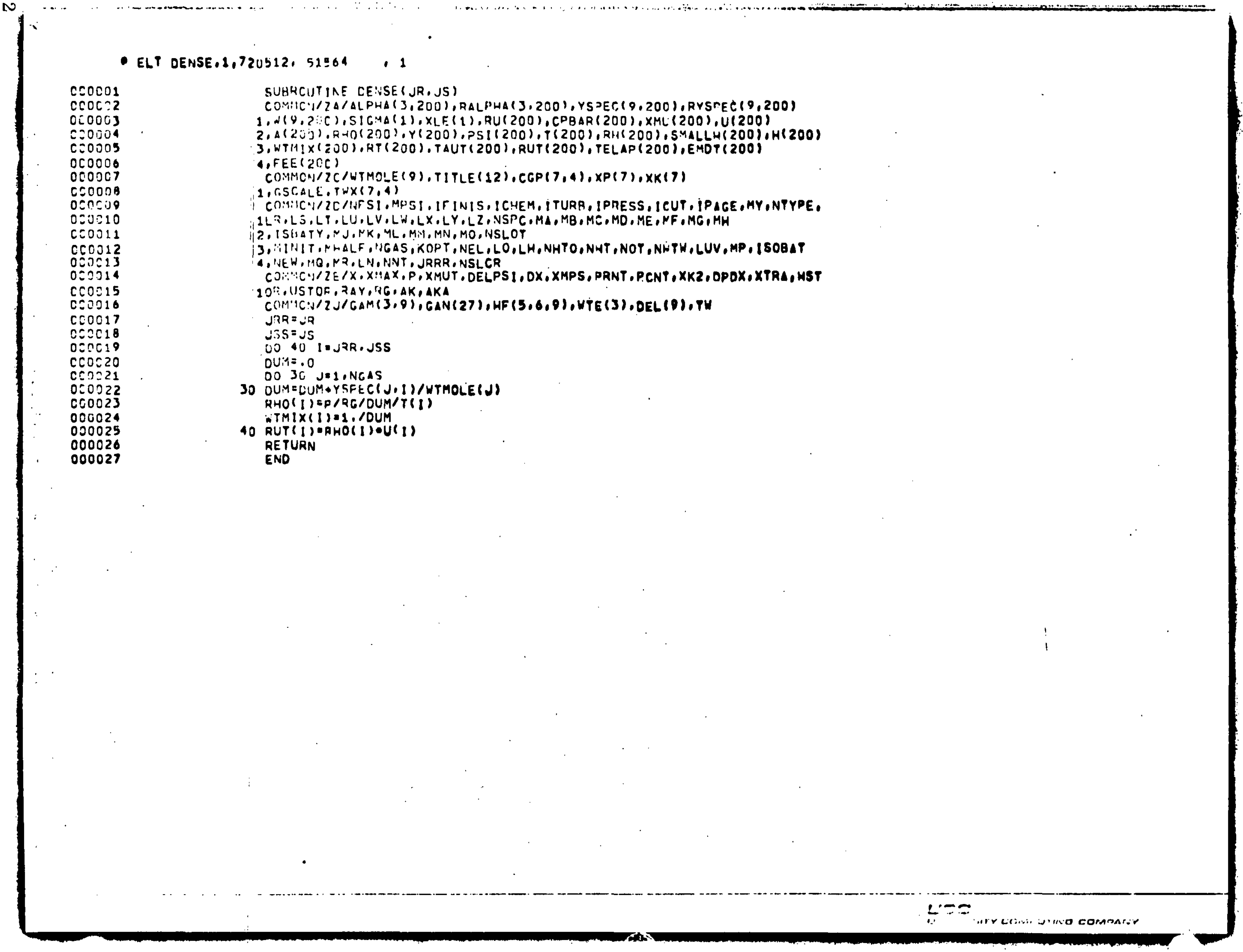
```
      PSIWA=PSIW
      Y(1)=YW
      GO TO 2
      END
```

```
• ELT CONSRV,1,720512, 51587    , 1
              SUBROUTINE CONSRV
              COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
             1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
             2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
             3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
             4,FEE(200)
              COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
             1,IDA(200),TCHEM(200),WDT(9,200)
              COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
             1,GSCALE,TMX(7,4)
              COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
             1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
             2,ISHATY,MJ,MK,ML,MM,MN,MO,NSLOT
             3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
             4,NEW,NQ,NR,LN,NNT,JR,NSLCR
              COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
             1OR,USTOR,RAY,RG,AK,AKA
              COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      C
              IF(NPSI.EQ.1)GO TO 22
      C
              DLS=DELPSI**2
              SCH=SIGMA(1)/XLE(1)
              EX1=DLS/DX
           70 DO 100 I=2,NPSI
              IP=I+1
              IM=I-1
      C
      C       MOMENTUM
              IF(NTYPE.EQ.0)EX1=PSI(I)*DLS/DX
              EX11=.5*(A(I)+A(I+1))
              EX12=.5*(A(I)+A(I-1))
              RU(I)=(EX11*(U(I+1)-U(I))+EX12*(U(I-1)-U(I)))/EX1+U(I)
      C
      C       ENERGY
              EX5=A(I)/SCH
              EX6=.5*(EX5+A(I+1)/SCH)
              EX7=.5*(EX5+A(I-1)/SCH)
              EX13=EX1-EX6-EX7
              EX8=A(I)/SIGMA(1)
              EX9=.5/SIGMA(1)*(A(I)+A(I+1))
              EX10=.5/SIGMA(1)*(A(I)+A(I-1))
              EX14=EX1-EX9-EX10
              IF(ABS(SIGMA(1)-1.).LT..001) GO TO 26
           29 EX20=EX11-EX9
              EX21=EX12-EX10
              EX22=EX20*TAUT(I+1)
              EX23=EX21*TAUT(I-1)
              EX24=TAUT(I)*(EX20+EX21)
              EX25=EX22-EX24+EX23
              GO TO 34
           26 EX25=.0
           34 IF(ABS(XLE(1)-1.).LT..001) GO TO 32
           33 EX30=EX6-EX9
              EX31=EX7-EX10
              EX32=.0
              EX33=.0
              EX34=.0
      EX35=.0
      DO 35 J=1,NSPC
      EX32=EX32+W(J,I)*YSPEC(J,I+1)
      EX33=EX33+W(J,I+1)*YSPEC(J,I)
      EX34=EX34+W(J,I-1)*YSPEC(J,I)
   35 EX35=EX35+W(J,I)*YSPEC(J,I-1)
      EX36=.5*EX30*(SMALLH(I+1)+EX32)
      EX37=.5*(EX30*(SMALLH(I)+EX33)+EX31*(SMALLH(I)+EX34))
      EX38=.5*EX31*(SMALLH(I-1)+EX35)
      EX39=EX36-EX37+EX38
      GO TO 36
   32 EX39=.0
   36 RH(I)=(EX9*H(I+1)+EX14*H(I)+EX10*H(I-1)+EX25+EX39)/EX1
C
      IF(ICHEM.EQ.2)GO TO 300
C
C     DIFFUSION (SPECIES)
      DO 240 J=1,NSPC
  240 RYSPEC(J,I)=WDT(J,I)+(EX6*YSPEC(J,IP)+EX13*YSPEC(J,I)
     1+EX7*YSPEC(J,IM))/EX1
      GO TO 100
C
  300 CONTINUE
C     DIFFUSION (ELEMENTS)
      DO 40 J=1,NEL
   40 RALPHA(J,I)=+(EX6*ALPHA(J,I+1)+EX13*
     1ALPHA(J,I)+EX7*ALPHA(J,I-1))/EX1
C
  100 CONTINUE
C
C     SOLVE EXPLICIT EQUATIONS   ON AXIS
      IF(PSI(1).GE.DELPSI)  GO TO 21
C
C     MOMENTUM
      IF(NTYPE.NE.0)GO TO 133
      EX16=4.*XMU(1)*DX/DLS
      GO TO 76
  133 EX16=2.*XMU(1)*DX*RUT(1)/DLS
   76 CONTINUE
      RU(1)=EX16*(U(2)-U(1))+U(1)
C
C     ENERGY
      IF(ABS(SIGMA(1)-1.0)-.001)38,39,39
   39 EX40=(1.0-1.0/SIGMA(1))*(TAUT(2)-TAUT(1))
      GO TO 41
   38 EX40=.0
   41 IF(ABS(XLE(1)-1.0)-.001)42,43,43
   43 EX50=.0
      DO 37 J=1,NSPC
   37 EX50=EX50+W(J,1)*YSPEC(J,2)
      EX51=(XLE(1)-1.0)/SIGMA(1)*(EX50-SMALLH(1))
      GO TO 44
   42 EX51=.0
   44 RH(1)=H(1)+EX16*((H(2)-H(1))/SIGMA(1)+EX40+EX51)
C
      IF(ICHEM.EQ.2)GO TO 400
C
      DO 250 J=1,NSPC
  250 RYSPEC(J,1)=YSPEC(J,1)+WDT(J,1)+EX16*(YSPEC(J,2)-YSPEC(J,1))/SCH
      GO TO 22
```

```
C
  400 CONTINUE
C     DIFFUSION (ELEMENTS)
      DO 200 J=1,NEL
  200 RALPHA(J,1)=                     *EX16*XLE(1)*(ALPHA(J,2)-ALPHA(J,1))
     1/SIGMA(1)+ALPHA(J,1)
      GO TO 22
C     MOMENTUM
   21 RU(1)=U(1)
C     ENERGY
      RH(1)=H(1)
      IF(ICHEM.EQ.2)GO TO 500
C     DIFFUSION (SPECIES)
      DO 25 J=1,NSPC
      RYSPEC(J,1)=YSPEC(J,1)+WDT(J,1)
   25 CONTINUE
      GO TO 22
  500 CONTINUE
C     DIFFUSION (ELEMENTS)
      DO 23 J=1,NEL
   23 RALPHA(J,1)=ALPHA(J,1)
C     EDGE CONDITIONS
   22 CONTINUE
      RU(MPSI)=U(MPSI)
      RH(MPSI)=H(MPSI)
      IF(ICHEM.EQ.2)GO TO 600
      DO 27 J=1,NSPC
      RYSPEC(J,MPSI)=YSPEC(J,MPSI)+WDT(J,MPSI)
   27 CONTINUE
      GO TO 303
  600 CONTINUE
      DO 24 J=1,NEL
   24 RALPHA(J,MPSI)=ALPHA(J,MPSI)
  303 CONTINUE
      DO 50 I=1,MPSI
      RU(I)=RU(I)-DX*DPDX/RUT(I)
   50 CONTINUE
      IF(LZ.EQ.0)RETURN
      WRITE(6,2)
    2 FORMAT(50X,11HFROM CONSRV)
      DO 20 I=1,MPSI
   20 WRITE(6,1)RU(I),U(I),RH(I),H(I),TAUT(I),T(I),RT(I),SMALLH(I)
    1 FORMAT(8E15.7)
      RETURN
      END
```

```
• ELT CPRF,1,720512, 51548     , 1

                FUNCTION CPRF(F4LD)
          C                          FUNCTION TO COMPUTE DUCT PRESSURE RATIO
          C                          GIVE 4FL/D
          C
                X=ALOG(F4LD)
                X2=X*X
                X3=X*X2
          C
                CPRF=.38804392 -.732167078E-1*X - .69749989E-2*X2 + .16726396E-2
               1      *X3
                RETURN
                END
```

```
• ELT CRSPLT•1•720512, 5:629    , 1

              SUBROUTINE CRSPLT(DMIN,AR,AFRZ,PT)
              DIMENSION DMIN(6,22),AR(22),AFRZ(6),AF(6,22)                      CRSP20
        C  PROGRAM TO FIND AFRZ AT OF=4,6,8  AND HIGH AND LOW PT                CRSP30
              DC 2C I=1,22                                                      CRSP60
          IF ( AR(I) .GT. 2,5) GO TO 30
              AF(1,I)=EXP(130,063-292,214*AR(I)+247,01*AR(I)**2-93,2601*AR(I)
           1**3+12.9647*AR(I)**4)
         30   IF(AR(I).GT.2.5)   AF(1,I)=AF(1,I-1)
              AF(2,I)=EXP(22.9227-23.3745*AR(I)+9,9097*AR(I)**2-2,00204*AR(I)
           1**3+.14P849*AR(I)**4)
         20   AF(3,I)=EXP(16.1877-11.9281*AR(I)+4,0312*AR(I)**2-,68539*AR(I)
           1**3+.044397*AR(I)**4)
              IF(PT.LT.500.) GO TO 24
              DC 22 I=1,22                                                      CRSP50
          IF ( AR(I) .GT. 3,0) GO TO 31
              AF(4,I)=EXP(40,8338-63,8576*AR(I)+38,100*AR(I)**2-11,0758*AR(I)
           1**3+1.21947*AR(I)**4)
         31   IF(AR(I).GT.3.) AF(4,I)=AF(4,I-1)
              AF(5,I)=EXP(17,8105-12,5154*AR(I)+3,9978*AR(I)**2-,693467*AR(I)
           1**3+.046737*AR(I)**4)
              AF(6,I)=EXP(15.1589-4.80814*AR(I)+.52358*AR(I)**2+,037887*AR(I)
           1**3-.008876*AR(I)**4)
         22   CONTINUE
              GC TC 100                                                         CRSP150
         24   DC 21 I=1,22
              AF(4,I)=AF(1,I)                                                   CRSP170
              AF(5,I)=AF(2,I)                                                   CRSP180
              AF(6,I)=AF(3,I)                                                   CRSP190
          IF ( AR(I) .GT. 2.0) GO TO 32
              AF(1,I)=EXP(372.719-1018.39*AR(I)+1035.75*AR(I)**2-466,58*AR(I)
           U**3+77.855R*AR(I)**4)
         32   IF(AR(I).GT.2.) AF(1,I)=AF(1,I-1)
              AF(2,I)=EXP(13.7836-15.7784*AR(I)+6,0358*AR(I)**2-1.1348*AR(I)
           1**3+.070778*AR(I)**4)
         21   AF(3,I)=EXP(10.1731-9.4258*AR(I)+2,87533*AR(I)**2-.460734*AR(I)
           1**3+.028904*AR(I)**4)
        100   DC 110 I=1,6                                                      CRSP240
              IF(DMIN(I,1).GE.AF(I,1)) GO TO 111                                CRSP250
              DC 112 J=2,22                                                     CRSP260
              IF(DMIN(I,J).LT.AF(I,J)) GO TO 112                                CRSP270
              A=AF(I,J-1)-((AF(I,J-1)-AF(I,J))/(AR(  J-1)-AR(  J)))*AR(  J-1)
              C=(AF(I,J-1)-AF(I,J))/(AR(  J-1)-AR(  J))
              D=(DMIN(I,J-1)-DMIN(I,J))/(AR(  J-1)-AR(  J))
              B=DMIN(I,J-1)-D*AR(  J-1)
              AFRZ(I)=(B-A)/(C-D)                                               CRSP320
              GO TC 110                                                         CRSP330
        112   CONTINUE                                                          CRSP340
              AFRZ(I)=5.0                                                       CRSP360
              GO TC 110                                                         CRSP370
        111   AFRZ(I)=1.0                                                       CRSP380
        110   CONTINUE                                                          CRSP390
              RETURN                                                            CRSP420
              END                                                               CRSP430
```



```
 • ELT DENSE,1,720512, 51564     , 1

          SUBROUTINE DENSE(JR,JS)
          COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
         1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XMU(200),U(200)
         2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
         3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
         4,FEE(200)
          COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
         1,GSCALE,TWX(7,4)
          COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
         1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
         2,ISOATY,MJ,MK,ML,MM,MN,MO,NSLOT
         3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
         4,NEW,MQ,MR,LN,NNT,JRRR,NSLCR
          COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,WST
         1OR,USTOR,RAY,RG,AK,AKA
          COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
          JRR=JR
          JSS=JS
          DO 40 I=JRR,JSS
          DUM=.0
          DO 30 J=1,NGAS
       30 DUM=DUM+YSPEC(J,I)/WTMOLE(J)
          RHO(I)=P/RG/DUM/T(I)
          WTMIX(I)=1./DUM
       40 RUT(I)=RHO(I)*U(I)
          RETURN
          END
```

UCC
ITY COMPUTING COMPANY

```
• ELT DIV,1,720512, 51606      , 1

            SUBRCUTINE DIV(RE,RT,DVL)
      C  DIVERGENCE LOSS PER CPIA 178  GAMMA =1.2  TAO CONTCURED NOZZLES
      C  RE  RADIUS EXIT
      C  RT  THRCAT RADIUS
      C  DVL  DIVERGENCE LOSS  PERCENT/100
      C  ASSUMS VACLUM CONDITION  SEE PAGE 6  FIG A-4  CPIA 178
            AR=ALOG((RE/RT)**2)
            DVL=.9309+.021308*AR-.0026279*AR**2+.00011325*AR**3
            RETURN
            END
```

```
00101          SUBROUTINE DYNAM
00101     C ******* DYNAMICS MODEL FLOW CALCLLATIONS
00101     C
00101     C         THIS SECTION PERFORMS THE FLOW CALCULATIONS ASSOCIATED WITH
00101     C         EACH CONNECTOR.  THE CALCULATIONS ARE DEPENDENT ON THE GIVEN
00101     C         INITIAL CONDITIONS FOR THE VOLUMES AND THEIR RESPECTIVE
00101     C         CONNECTORS.
00101     C
00101     C
00101     C         THE GIVEN INITIAL CONDITIONS FOR THE VOLUMES ARE
00101     C
00101     C         1. NODE NUMBER
00101     C         2. VOLUME SIZE
00101     C         3. TEMPERATURE
00101     C         4. PRESSURE
00101     C         5. MIXTURE  (( OXYGEN BY WEIGHT)
00101     C         6. COMBUSTER OR NOT COMBUSTER
00101     C         THE GIVEN INITIAL CONDITIONS FOR THE CONNECTORS ARE
00101     C
00101     C         1. CONNECTED NODE 1
00101     C         2. CONNECTED NODE 2
00101     C         3. ORIFICE DISCHARGE COEFF OR DUCT LENGTH
00101     C         4. CROSS SECTIONAL AREA
00101     C         5. RESTRICTION-TYPE  (ORIFICE OR DUCT)
00101     C
00101     C    DYNAMICS MODEL SUBROUTINE
00101     C
00103          COMMON/MDYN/NNODE,NCONN,CD,WOUTC,NOINJ(24),LPRTF,
00103         1NVOL(60),ICOMB(60),PRES(60),TEMP(60),RMIX(60),VOL(60),
00103         2IADMIT(60,2),IRTYPE(60),CAREA(60),CCOEFF(60),DLEN(60),TIMON(60),
00103         3TIMOFF(60),TIMOPN(60),TIMCLS(60),NNCINJ,TFSTST,
00103     32.         4SPTIME,SPKTL,SPKF,SPGAP,SPARKP,SPARKE,DC,NCHEK,TIME,L,ICVOL,DPRES
00104     33.          COMMON/NJECT/WT,PO,FM,PSTAT                                                            *NEW
00105     34.          COMMON/CONP/NCOMBU,NINJE                                                               ***I
00106     35.          COMMON/JLCDYN/THRTAR,ACIMP,ACW
00107     36.          DIMENSION SWDCT(60),QDOT(60),SADMIT(60),C(60),Q(60),SMDOT(60)
00110     37.          DIMENSION WNEW(20),RMIXN(20),QNEW(20),TNEW(20),PNEW(20)
00111     38.          EQUIVALENCE (CCOEF(1),DLEN(1))
00112     39.          DIMENSION W(60),DCOEF(60)
00113     40.          DIMENSION BLK(3),PRTP(3),PRTC(3),PRT(3)
00114     41.          DATA BLK,PRTP,PRTC/ '    ','    ','    ','(PIL','CT) ','    ',
00114     42.         1 '(COM','BUST','OR) '/
00120   *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
00120     43.          IF (TIME.NE. 0.0)  GO TO 1900
00122     44.          I=0
00123     45.          DATA NSPK/0/
00125     46.          DELT=.0005
00126     47.          DPRES=1000.
00127     48.          ACIMP=0.0
00130     49.          ACW=0.0
00131     50.          IADVTM=0
00131     51.    C     W=0.0
00132     52.          DO 1361 I=1,NCONN
00135     53.          IF (IRTYPE(I).EQ.2) THRTAR=CAREA(I)
00137     54.     1361 CONTINUE
00141   *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
00141     55.          IF (SPTIME.NE.0.0)  GO TO 1978
00143     56.          SPTIME=SPKF
00144     57.     1978 CONTINUE
00144     58.    C                                 SET COMBUSTER/PILOT NODES
00145     59.          DO 1311 I=1,NNODE
00150     60.          IF(ICOMB(I).EQ.1 .OR. ICOMB(I).EQ.2)ICVOL=I
00152     61.          IF(ICOMB(I).EQ.3 .OR. ICOMB(I).EQ.4)IPVOL=I
00154     62.     1311 CONTINUE
00154     63.    C
00154     64.    C          LOOP THRU DYNAMICS MODEL ONCE FOR EACH CONNECTOR AND
00154     65.    C          PERFORM THE FLOW CALCULATIONS BASED UPON RESTRICTION
00154     66.    C          TYPE AND HOT OR COLD GAS CRITEREA.
00154     67.    C
00154     68.    C
00154     69.    C
00154     70.    C          SET INDEXES WHICH WILL DETERMINE IF VOLUME IS A COMBUSTER
00154     71.    C          OR NOT, IF IGNITION HAS OCCURRED OR NOT, AND THE RESTRICTION
00154     72.    C          TYPE
00154     73.    C
00154     74.    C               IF ICOMB(J)=0  THEN VOLUME IS NOT A COMBUSTER
00154     75.    C               IF ICOMB(J)=1  THEN VOLUME IS A COMBUSTER, NOT IGNITED
00154     76.    C               IF ICOMB(J)=2  THEN VOLUME IS AN IGNITED COMBUSTER
00154     77.    C               IF ICOMB(J)=3 THEN VOLUME IS A PILOT, NOT IGNITED
00154     78.    C               IF ICOMB(J)=4 THEN VOLUME IS AN IGNITED PILOT
00154     79.    C
00154     80.    C               IF IRTYPE(J)=0 THEN RESTRICTION TYPE IS ORIFICE
00154     81.    C               IF IRTYPE(J)=1 THEN RESTRICTION TYPE IS DUCT
00154     82.    C               IF IRTYPE(J)=2 THEN RESTRICTION IS THE THROAT
00154     83.    C
00154     84.    C
00154     85.    C
00154     86.    C
00156     87.     1900 CONTINUE
00157     88.          DATA PI/3.1415926/
00161     89.          WT=0.0
```

```
00162     90*          FM=0.0
00162     91*    C
00163     92*          C1=64.5)8
00164     93*          C2=2.016
00165     94*          C3=32.0
00166     95*          C4=1.0
00167     96*          C5=18540.C
00170     97*          C6=1.4
00171     98*          C7=2.0
00172     99*          C8=SQRT(4.0/PI)
00173    100*          C9=46.6E-10
00174    101*          C10=SQRT(32.174*12.0)
00174    102*    C                          STARTING GUESS AT 4F
00175    103*          FFC=0.02
00176    104*          IF (IADVTM.EQ.0) GO TO 1369
00200    105*          TIME=TIME+.00005
00201    106*          IADVTM=0
00202    107*     1369 CONTINUE
00203    108*          L=L+1
00204    109*          OLDP=PRES(NCHEK)
00204    110*    C                    ZERO SUMMING VECTORS + GET START VOL MASSES
00205    111*          DO 10 I=1,NNODE
00210    112*          QDOT(I)=0.0
00211    113*          C(I)=0.0
00212    114*          SWDOT(I)=C.0
00213    115*          SADMIT(I)=0.0
00214    116*          SMDOT(I)=C.0
00215 *DIAGNOSTIC*     THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00215    117*          IF(PRES(I).EQ.0.0)PRES(I)=1.0E-11
00217    118*          WA1=C1/(C2*RMIX(I) + C3*(1.0-RMIX(I)))
00220    119*          W(I)=PRES(I)*VOL(I)*WA1/(C5*TEMP(I))
00221    120*          IF(PRES(I).LT.1.0E-10)W(I)=1.0E-20
00223    121*          Q(I)=TEMP(I)*W(I)*(3.5-3.26*RMIX(I))
00224    122*       10 CONTINUE
00224    123*    C
00224    124*    C         CALCULATE THE PRESSURE RATIO ACROSS CONNECTOR J
00224    125*    C         AND DETERMINE THE DIRECTION OF FLOW
00224    126*    C
00226    127*          DO 500 J=1,NCONN
00231    128*          IVOL1=IADMIT(J,1)
00232    129*          IVOL2=IADMIT(J,2)
00232    130*    C                    CALC CROSS SECTIONAL AREA
00233    131*          ADD=1.0
00234    132*          B=TIMON(J)+TIMOFF(J)+TIMOPN(J)+TIMCLS(J)
00235 *DIAGNOSTIC*     THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00235    133*          IF(B.EQ.0.0)GO TO 50
00237    134*          ADD=0.0
00240    135*          IF(TIME .LE. TIMON(J))GO TO 50
00242    136*          IF(TIME .GT. TIMON(J)+TIMOPN(J))GO TO 45
00242    137*    C                                                 FIT FOR OPENING
00244    138*          ADD=(TIME-TIMON(J))/(TIMOPN(J)+1.0E-10)
00245 *DIAGNOSTIC*     THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00245    139*          IF(TIMOPN(J).EQ.0.0)ADD=1.0
00247    140*          GO TO 50
00247    141*    C                                                 CHECK IF COMPLETELY OPEN
00250    142*       45 ADD=1.0
00251    143*          IF(TIME .LT. TIMOFF(J))GO TO 50
00251    144*    C                    CHECK IF COMPLETELY CLOSED
00253    145*          ADD=0.0
00254    146*          IF(TIME .GT. TIMCLS(J)+TIMOFF(J))GO TO 50
```

```
00254   147*   C                                          FIT FOR CLOSING
00256   148*          ADD=1.0 - ((TIME-TIMOFF(J))/(TIMCLS(J)+1.0E-10))
00257  *DIAGNOSTIC*   THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00257   149*          IF(TIMCLS(J).EQ.0.0)ADD=0.0
00257   150*   C
00261   151*     50   AREA=ADD*CAREA(J)
00262   152*          DIAM=C8*SQRT(AREA)
00262   153*   C                            CALC PRESSURE RATIO
00263   154*          PRESRT=PRES(IVOL1)/PRES(IVOL2)
00264   155*          NDX3=IVOL2
00265   156*          NDX4=IVOL1
00265   157*   C                            TEST FOR HIGHER PRESSURE
00266   158*          IF (1.0-PRESRT) 101,102,102
00271   159*    101   NDX3=IVOL1
00272   160*          NDX4=IVOL2
00273   161*          PRESRT=1.0/PRESRT
00273   162*   C
00274   163*    102   CONTINUE
00275   164*          PDIF=PRES(NDX3)-PRES(NDX4)
00276   165*          NDX1=ICOMP(NDX3) + 1
00277   166*          NDX2=IRTYPE(J)+1
00277   167*   C                            GO TO HOT OR COLD GAS CALCULATION
00300   168*          IF(PDIF .LT. 1.0E-3)GO TO 150
00302   169*          GO TO (200,200,250,200,250),NDX1
00302   170*   C                            GO TO ORIFICE OR DUCT CALCULATIONS
00303   171*    103   GO TO (300,350,300),NDX2
00303   172*   C
00303   173*   C
00303   174*   C                  IF NO PRESSURE DIFFERENCE, SET VALUES TO ZERO
00303   175*   C
00304   176*    150   WDOT=0.0
00305   177*          ADMIT=0.0
00306   178*          GO TO 375
00306   179*   C
00306   180*   C
00306   181*   C            COLD GAS CALCULATIONS
00306   182*   C
00306   183*   C
00306   184*   C                            USE WA TO AVOID DOUBLE USE OF INDEX
00307   185*    200   WA1=RMIX(NDX3)
00307   186*   C                            CALCULATE ORIFICE MOLE WT
00310   187*          RMWT=C1/(C2*WA1+C3*(C4-WA1))
00310   188*   C                            CALCULATE GAS CONSTANT
00311   189*          RCONST=C5/RMWT
00312   190*          GM=1.4
00313   191*          GO TO 103
00313   192*   C
00313   193*   C
00313   194*   C           HOT GAS CALCULATIONS
00313   195*   C
00313   196*   C
00314   197*    250   CONTINUE
00315   198*          OF=RMIX(NDX3)/(1.0-RMIX(NDX3))
00316   199*          GM=SPHEAT(OF)
00317   200*          RMWT=FMOLWT(OF)
00320   201*          RCONST=C5/RMWT
00321   202*          GO TO 103
00321   203*   C
00321   204*   C
00321   205*   C      PROCESS AN ORIFICE
```

```
00321    C
00322     300  CONTINUE
00322    C
00322    C                                    CALCULATE CHOKING PRESSURE RATIO
00323          WA1=GM+1.0
00324          WA2=2.0/WA1
00325          WA3=GM-1.0
00326          CPR=WA2**(GM/WA3)
00327          IF (PRESRT-CPR) 301,302,302
00327    C                                    CHOKED
00332     301  F=SQRT(WA2**(WA1/WA3))
00333          GO TO 303
00333    C                                    UNCHOKED
00334     302  F=SQRT(2.0/WA3)
00335          F=F*SQRT( (PRESRT**(2.0/GM)) - (PRESRT**(WA1/GM)) )
00336      303 WA3=AREA*PRES(NDX3)
00337          WA5=RCONST*TEMP(NDX3)
00340          WA5=SQRT(GM/WA5)
00341          ADMIT   =WA3*WA5*F*C10/PDIF
00342          WDOT=ADMIT*PDIF
00343          GO TO 375
00343    C
00343    C
00343    C     PROCESS A DUCT
00343    C
00344     350  CONTINUE
00344    C
00345          DO 356 K=1,3
00345    C                                    CALCULATE  4FL/D
00350          FFP=FFC*DLEN(J)/DIAM
00350    C                                    CALCULATE CHOKING PRESSURE RATIO
00351          CPR=CPRF(FFP)
00352          IF (PRESRT-CPR) 351,352,352
00352    C                                    CHOKED
00355      351 B=CFLOW(FFP)
00356          GO TO 353
00356    C                                    UNCHOKED
00357      352 B =FLOW(PRESRT,FFP)
00360     353  WA6=AREA*PRES(NDX3)
00361          WA7=SQRT(RCONST*TEMP(NDX3))*PDIF
00362          ADMIT=B*WA6*C10/WA7
00363          WDOT=ADMIT*PDIF
00364          FMU=C9*SQRT(RMWT)*(TEMP(NDX3)**0.6)
00365          REN=W(NDX3)*DIAM/(AREA*FMU)
00366          IF(2100.0-REN)354,355,355
00371      354 FFC=FFCF(REN)
00372          GO TO 356
00373      355 FFC=64.0/REN
00374      356 CONTINUE
00374    C
00374    C
00374    C                                 CALCULATE ADMITTANCE DEPENDENT VARIABLES
00374    C
00376     375  CONTINUE
00377          IF (NINJE.EQ.0)  GO TO 385
00401          IF (NDX4.NE.ICVOL)  GO TO 385
00403          DO 386 I=1,NNCINJ
00406          IF(NOINJ(I) .EQ.NDX3)GO TO 387
00410      386 CONTINUE
00412          GO TO 385
```

```
00412    266*    C WT IS USED AS INPUT BY THE INJECTION MODEL
00413    267*      387 WT=WT+WDOT
00414    268*          FM=((FM*(WT-WDOT))+(WDOT*RMIX(NDX3)))/WT
00415    269*      385 CONTINUE
00416    270*          IF (IRTYPE(J).NE.2)  GO TO 388
00416    271*    C W IS USED AS INPUT BY COLD, THE COLD PERFORMANCE MODEL
00420    272*          WFLW=WDOT
00421    273*      388 CONTINUE
00421    274*    C
00422    275*          ADD =TEMP(NDX3)*WDOT*(3.5-3.26*RMIX(NDX3))
00423    276*          QDOT(NDX3)=QDOT(NDX3) - ADD
00424    277*          QDOT(NDX4)=QDOT(NDX4) + ADD
00425    278*          SWDOT(NDX3)=SWDOT(NDX3) - WDOT
00426    279*          SWDOT(NDX4)=SWDOT(NDX4) + WDOT
00427    280*          SADMIT(NDX3)=SADMIT(NDX3) + ADMIT
00430    281*          SADMIT(NDX4)=SADMIT(NDX4) + ADMIT
00431    282*          ADD=RMIX(NDX3)*WDOT
00432    283*          SMDOT(NDX3)=SMDOT(NDX3) - ADD
00433    284*          SMDOT(NDX4)=SMDOT(NDX4) + ADD
00434    285*     500  CONTINUE
00434    286*    C
00434    287*    C        CALCULATE TEMP CHANGES IN ACCUMULATION VOLUMES
00434    288*    C
00436    289*          KOUNT=100
00437    290*     600  CONTINUE
00440    291*          KOUNT=KOUNT-1
00441    292*          IF(KOUNT .EQ. 0)GO TO 9000
00443    293*          CHECK=1000.0
00444    294*          DO 1100 I=1,NNODE
00447    295*          WNEW(I)=W(I) + DELT*SWDOT(I)
00450    296*          RMIXN(I)=(RMIX(I)*W(I) + DELT*SMDOT(I))/WNEW(I)
00451    297*          QNEW(I)=Q(I) + DELT*QDOT(I)
00452    298*          IF(ICOMB(I).NE.2 .AND. ICOMB(I).NE.4)GO TO 1090
00452    299*    C                                    HOT COMBUSTER EQUATION
00454    300*          OF=RMIXN(I)/(1.0-RMIXN(I))
00455    301*          RMWT=FMOLWT(OF)
00456    302*          TNEW(I)=TGR(PRES(I),OF)
00457    303*          GO TO 1095
00457    304*    C                                        NON-COMBUSTER EQUATION
00460    305*     1090 TNEW(I)=QNEW(I)/(WNEW(I)*(3.5-3.26*RMIXN(I)))
00461    306*          RMWT   =64.508/(2.016*RMIXN(I)+32.0*(1.0-RMIXN(I)))
00462    307*     1095 CONTINUE
00463    308*          C(I)=VOL(I)*RMWT/(C5*TNEW(I))
00464    309*          PNEW(I)=PRES(I) + DELT*SWDOT(I)/C(I)
00465    310*          CHCK=SADMIT(I)
00466    *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
00466    311*          IF(CHCK.EQ.0.0)CHCK=1.0E-15
00470    312*          CHCK=C(I)/CHCK
00471    313*          CHECK=AMIN1(CHECK,CHCK)
00472    314*     1100 CONTINUE
00474    315*          IF(DELT .LE. CHECK*BUMP(L))GO TO 1200
00474    316*    C                                        NOT STABLE, RESET AND TRY AGAIN
00476    317*          DELT=0.9*CHECK*BUMP(L)
00477    318*          GO TO 600
00477    319*    C                                        WAS STABLE, GO ON
00500    320*     1200 CONTINUE
00501    321*          IF (DELT.GE.CHECK*BUMP(L)*.8)  GO TO 1367
00503    322*          IF (NSPK.EQ.1.OR.L.EQ.1)  GO TO 1367
00505    323*          DELT=CHECK*BUMP(L)*.8
00506    324*          GO TO 600
```

```
00507      1367 CONTINUE
C0510           IF (DELT.GT. 1E-07 .OR.NSPK.EQ.1)  GO TO 1368
C0512           IADVTM=1
00513      1368 CONTINUE
00514           ADD=TIME+DELT
C0515           IF (SPTIME.GE.ADD.OR.TIME.GE.TFSTST)  GO TO 1205
C0517           NSPK=1
C0520           OLDELT=DELT
00521           DELT=SPTIME-TIME
C0522           GO TO 630
00522      C
00523      1205 CONTINUE
C0524           TIME=ADD
C0524      C                                      SET NEW STABLE VALUES
C0525           DO 1300 I=1,NNODE
C0530           W(I)=WNEW(I)
00531           RMIX(I)=RMIXN(I)
00532           Q(I)=QNEW(I)
C0533           TEMP(I)=TNEW(I)
C0534           PRES(I)=PNEW(I)
C0535      1300 CONTINUE
00537           THRUST=PRES(ICVOL)*THRTAR*CD                                              *NEW
C0540           ACIMP=ACIMP+(THRUST*(TIME-OLDTIM))                                        ***I
C0541           ACW=ACW+(WFLW*(TIME-OLDTIM))
00542           SPIMP=ACIMP/ACW
00543           IF (NSPK.EQ.1.OR.MOD(L,LPRTF).EQ.0)  GO TO 1310
C0545           GO TO 1312
00545      C  PRINT RESULTS
0C546      1310 WRITE (6,952)  TIME
C0551           WRITE (6,9963)   L
00554      9963 FORMAT (2X,'ITERATION NO'I16)
C0555           WRITE(6,950)
C0557           DO 1312 I=1,NNODE                                                         ***I
C0562           IF(ICVCL.EQ.I)GO TO 1215
0C564           IF(IPVCL.EQ.I)GO TO 1210
00566           DO 1207 J=1,3
0C571      1207 PRT(J)=BLK(J)
00573           GO TO 1220
00573      C
00574      1210 CONTINUE
C0575           DO 1212 J=1,3
0C600      1212 PRT(J)=PRTP(J)
00602           GO TO 1220
0C602      C
00603      1215 CONTINUE
00604           DO 1217 J=1,3
0C607      1217 PRT(J)=PRTC(J)
C0607      C
00611      1220 CONTINUE
00612           PERCT=100.0*RMIX(I)
00613           WRITE(6,951) I,PRT,PRES(I),TEMP(I),PERCT
00625      1312 CONTINUE
00627           WRITE (6,1492) DELT
0C632      1492 FORMAT (6X,'DELT USED',F10.6)
00633           WRITE (6,7775) THRUST,ACIMP,SPIMP
00640      7775 FORMAT ('C',6X,'THRUST=',F12.2,6X,'IMPULSE=',F12.5,6X,'ISP=',F12.2
00640          1)
00640      C  REMOVE LATER
00641           WRITE (6,8005) ACW
00644      8005 FORMAT (6X,'ACCUM WT FLOW=',F12.5)
```

```
00645           WRITE (6,953)  NCHEK,DPRES
00651      1313 CONTINUE
00651      C
00651      C            CONVERGENCE LOOP
00651      C
00651      C
00651      C
00651      C                IGNITION MODEL
00651      C
00651      C            CHECK IF IGNITION HAS ALREADY OCCURRED
00651      C
00652           IF (ICOMB(ICVCL).EQ.1.AND.ICOMB(IPVOL).EQ.4)  GO TO 850
00654           IF (NSPK.EQ.0)  GO TO 898
00656           IF (ICCMB(ICVCL).EQ.2)  GO TO 899
00656      C
00656      C            SPARK IGNITION MODEL
00656      C
00656      C            USE PILOT CONDITIONS OR COMBUSTIOR CONDITIONS?
00656      C
00660           IF(IPVCL.EQ.0)  GO TO 810
00662           PLUGP=PRES(IPVOL)
00663           IF (RMIX(IPVOL).GT.  .999)  GO TO 1825
00665           PLUGOF=RMIX(IPVOL)/(1.-RMIX(IPVOL))
00666           GC TO 825
00667       810 PLUGP=PRES(ICVOL)
00670           IF (RMIX(ICVOL).GT.  .999)  GO TO 1825
00672           PLUGOF=RMIX(ICVOL)/(1.-RMIX(ICVOL))
00673           GC TO 825
00674      1825 PLUGOF=1000.
00675       825 CCNTINUE
00675      C
00675      C            IS SUPPLIED POTENTIAL GREATER THAN BREAKDOWN POTENTIAL?
00675      C
00676           PG=PLUGP*SPGAP
00677           REQV=     5.202+(1.0224*(ALOG(PG)))+(.1029*((ALOG(PG))**2))+
00677          1(.0292*((ALOG(PG))**3))+(.0044*((ALOG(PG))**4))
00700           IF (REQV.GT.174.) REQV=20.
00702           REQV=EXP(REQV)
00703           IF(REQV-SPARKP)830,830,890
00703      C
00703      C            IS SPARK ENERGY SUFFICIENT FOR IGNITION?
00703      C
00706       830 IF(PLUGOF.LT.0.8.OR.PLUGOF.GT.390.) GO TO 891
00711           REQE=EXP(2.4959-(6.7104*(ALOG(PLUGOF)))+(1.9408*((ALOG(PLUGOF))
00711          1**2))-(.0512*((ALOG(PLUGOF))**3)))
00712           IF(REQE-SPARKE)840,840,891
00715       840 IF(IPVCL.EQ.0)  GO TO 893
00720           WRITE (6,905)
00722           ICOMB(IPVCL)=4
00723           TEMP(IPVOL)=TGR(PRES(IPVOL),PLUGOF)
00724           RMWT=FMOLWT(PLUGOF)
00725           WA1=64.5 8/((2.01*RMIX(IPVOL))+(32.*(1.-RMIX(IPVOL))))
00726           PRES(IPVOL)=W(IPVOL)*C5*TEMP(IPVOL)/(VOL(IPVOL)*RMWT)
00727           WRITE (6,907)TEMP(IPVOL),PRES(IPVOL)
00727      C
00727      C            CHECK FOR FLAME QUENCHING IN THE COMBUSTIR
00727      C
00733       850 IF (RMIX(ICVOL).GT..999)   GO TO 894
00736           OFCOMB=RMIX(ICVOL)/(1.-RMIX(ICVOL))
00737           IF(OFCOMB.LT.0.8.OR.OFCOMB.GT.390.)  GO TO 894
```

```
00741           QREG=17.6427-(14.5603*(ALOG(OFCOMB)))+(5.8153*((ALOG(OFCOMB))
00741          1**2))-(1.1561*((ALOG(OFCOMB))**3))+(.0991*((ALOG(OFCOMB))**4))
00742           IF(QREG-(PRES(ICVOL)*DC))893,893,894
00745       890 WRITE (6,901)
00747           GO TO 899
00750       891 IF(IPVCL.EQ.0)  GO TO 892
00753           WRITE (6,902)
00755           GO TO 899
00756       892 WRITE (6,903)
00760           GO TO 899
00761       893 WRITE (6,904)
00763           ICOMB(ICVCL)=2
00764           TEMP(ICVOL)=TGR(PRES(ICVOL),OFCOMB)
00765           RMWT=FMOLWT(OFCOMB)
00765     C REMOVE THIS WRITE STATEMENT LATER
00766           WRITE (6,4333) W(ICVOL),TEMP(ICVOL),VOL(ICVOL),RMWT
00774      4333 FORMAT (4F12.5)
00775           PRES(ICVOL)=W(ICVOL)*C5*TEMP(ICVOL)/(VOL(ICVOL)*RMWT)
00776           WRITE (6,906)  TEMP(ICVOL),PRES(ICVOL)
01002           IF (NSPK.EQ.1)  GO TO 899
01004           GO TO 898
01005       894 WRITE (6,908)
01007           IF (NSPK.EQ.1)  GO TO 899
01011           GO TO 898
01011     C  DETERMINE NEXT TIME A SPARK WILL OCCUR IF COMBUSTOR NOT YET LITE
01012       899 SPTIME=SPTIME+SPKF
01013           IF (SPTIME.LE.SPKTL)  GO TO 898
01015           WRITE  (6,909)
01017       898 CONTINUE
01020           DELP=PRES(NCHEK)-OLDP
01021           DELP=ABS(DELP)
01021     C
01022           PO=PRES(ICVOL)
01023           IF (L.LE.50)  GO TO 2022
01025           DPRES=DELP/(DELT*1000.)
01026           GO TO 2023
01027      2022 DPRES=1000.
01030      2023 CONTINUE
01031           IF (NSPK.EQ.1) DELT=OLDELT
01033           NSPK=0
01034           OLDTIM=TIME
01034     C
01035      2000 CONTINUE
01035     C
01036           RETURN
01036     C
01036     C                       ERROR CONDITION
01036     C
01037      9000 CONTINUE
01040           WRITE(6,9001)
01042      9001 FORMAT('1    CANNOT CONVERGE ON STABLE DEL TIME IN DYNAM')
01043           CALL EXIT
01044           RETURN
01044     C
01045       901 FORMAT ('C',6X,'NO SPARK HAS OCCURRED')
01046       902 FORMAT ('C',6X,'NO IGNITION IN THE PILOT')
01047       903 FORMAT ('C',6X,'NO IGNITION IN THE COMBUSTOR')
01050       904 FORMAT ('C',6X,'THE COMBUSTOR HAS BEEN IGNITED')
01051       905 FORMAT ('C',6X,'THE PILOT HAS BEEN IGNITED')
01052       906 FORMAT ('  NEW COMBUSTOR TEMP IS ',F8.1,' NEW PRESSURE IS ',F8.2)
```

```
01053    505*      907 FORMAT ('  NEW PILOT TEMP IS ',F8.1,' NEW PRESSURE IS ',F8.1)
01054    506*      908 FORMAT ('C',6X,'THE FLAME HAS QUENCHED IN THE COMBUSTOR')
01055    507*      909 FORMAT  ('0',6X,'SPARKING HAS BEEN COMPLETED')
01056    508*     950  FORMAT('      VOL NO',13X,'PRESSURE',6X,'TEMP',6X,'( OX', /
01056    509*         1       25X,'(PSIA)',7X,'(R)' /)
01057    510*     951  FORMAT(4X,I4,1X,3A4,F11.2,F10.1,F10.3)
01060    511*     952  FORMAT ('1',40X,'** DYNAMICS RESULTS **',// 43X,'TIME =',F10.6//)
01061    512*      953 FORMAT ('C',6X,'VOL ',I14,2X,'PRES CHANGE=',F10.3,'PSI PER MS')
01062    513*          END

          END OF UCC 1108 FORTRAN V COMPILATION.     7 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME    00:00.894
PHASE 2 TIME    00:00.054
PHASE 3 TIME    00:01.843
PHASE 4 TIME    00:00.068
PHASE 5 TIME    00:01.261
PHASE 6 TIME    00:01.106

TOTAL COMPILATION TIME =   00:05.231
    DYNAM              SYMBOLIC                                     (FN5)    12 JUN 72  19:02:07   0  00347272    14  514  (DELETED)
    DYNAM    CODE      RELOCATABLE                                  (FN5)    12 JUN 72  19:02:07   1  00365326    48    1  (DELETED)
                                                                                                   0  00365406    14  160
```

● ELT EQUILC,1,720512, 51563 , 1

```
      SUBROUTINE EQUILC(LAA,LBB)
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TWX(7,4)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZF/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
     1OR,USTOR,RAY,RG,AK,AKA
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      DIMENSION AW(9),EW(3)
      IF(MA.GT.0)WRITE(6,9)
    9 FORMAT(53X,14HFROM CHEMISTRY)
      LCC=LAA
      LDD=LBB
      DO 46 I=LCC,LDD
      DO 30 K=1,NEL
   30 EW(K)=ALPHA(K,I)/WTE(K)
      DO 91 J=1,NSPC
   91 AW(J)=0.0
      AW(NNT)=EW(LN)/GAM(LN,NNT)
C     TEST FOR PURE N2
      IF(ALPHA(LN,I).GE.1.)GO TO 53
C
   32 CONTINUE
C     H AND O BURNED TO H2O
      IF(EW(LH).GT.2.*EW(LO))GO TO 61
      AW(NHTO)=.5*EW(LH)
      AW(NOT)=.5*EW(LO)-.25*EW(LH)
      GO TO 53
   61 AW(NHTO)=EW(LO)
      AW(NHT)=.5*EW(LH)-EW(LO)
   53 CONTINUE
      DO 56 J=1,NSPC
      IF(AW(J).LT..0)AW(J)=.0
   56 YSPEC(J,I)=AW(J)*WTMOLE(J)
      IF(MA.GT.0)WRITE(6,4)T(I),SMALLH(I),FEE,(ALPHA(K,I),K=1,NEL)
     1,(YSPEC(J,I),J=1,NSPC)
    4 FORMAT(8E15.7)
   46 CONTINUE
      RETURN
      END
```

UCC
UNIVERSITY COMPUTING COMPANY

```
• ELT EXIT,1,720512,

                      SUBROUTINE EXIT
                      STOP
                      RETURN
                      END
```

```
 • ELT FG,1,720512, 51604 , 1

      SUBROUTINE FG
C     SPECIES ARE 1-H 2-O 3-H2O 4-H2 5-O2 6-OH 7-HO2 8-H2O2 9-DILUENT
      COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
     1,IDA(200),TCHEM(200),WDT(9,200)
      COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TKX(7,4)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZE/XXX,XMAX,PRES,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX
     1,XTRA,HSTOR,USTOR,RAY,RG,AK,AKA
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      COMMON/FCG/T,H,ALPHI(9),KPSI,A(8,8),Y(9),B(8),RHOB,TIMEP,DT       FG
      COMMON/FGY/BBB(8,8),E(8,8),N,NDUM,DEAD
      COMMON/FAR/RHO(6),QK(17,2),PL(17,10),X
      DIMENSION JRD(4,17),JRDY(68)
     1,RATES(3,17),RAZZ(51),QKC(17),JHDRDY(17),FRT(18)
     2,REORD(4,18),DLFRT(17)
      EQUIVALENCE(RATES,RAZZ)
     1,(JRD,JRDY)
      DATA RAZZ/
     16.E+13,C.,0.,
     26.E+13,C.,0.,
     31.E+13,C.,0.,
     41.E+13,C.,0.,
     52.19E+13,0.,-2593.,
     65.75E+12,0.,-392.7,
     71.74E+13,0.,-4758.,
     82.24E+14,0.,-8459.,
     91.E+13,C.,-906.3,
     Z2.34E+13,0.,-4632.,
     A3.18E+14,0.,-4532.,
     B1.E+16,C.,0.,
     C9.30E+14,0.,0.,
     D5.E+15,C.,0.,
     E1.E+17,C.,0.,
     F1.59E+15,0.,503.5,
     G8.4E+14,0.,2669./
      DATA JRDY/
     17,1,5,4, 7,1,6,6, 7,2,5,6, 7,6,5,3, 4,6,1,3,
     26,6,2,3, 4,2,1,6, 5,1,6,2, 8,6,3,7, 8,1,4,7,
     38,1,6,3, 2,1,6,18, 2,2,5,18, 1,1,4,18, 1,6,3,18,
     45,1,7,18, 6,6,8,18/
      DATA JHDRDY/11*0,6*-1/
C     CONVERT FROM KG/M**3 TO GM/CM**3
      RHO(1)=,001*RHOB
      DO 5 I=2,6
    5 RHO(I)=RHO(I-1)*RHO(1)
      X=0.
      DO 6 I=1,NGAS
    6 X=X+Y(I)
      DO 51 I=1,NGAG
      DO 52 J=1,6
      IF(T.LE.HF(1,J,I))GO TO 53
   52 CONTINUE
   53 FRT(I)=WTMOLE(I)/RG*(HF(5,J,I)+HF(4,J,I)*T
     1-HF(2,J,I)/T+HF(3,J,I)*(ALOG(T)-1.))
51    CONTINUE
      FRT(10)=.0
      IF(MH.GT.0)WRITE(6,3)(FRT(J),J=1,NGAS)
    3 FORMAT(56X,7HFROM FG/(8E15.7))
      DO 18 I=1,NREAC
      JPA=JRD(3,I)
      JPB=JRD(4,I)
      JRA=JRD(1,I)
      JRB=JRD(2,I)
      DLFRT(I)=FRT(JPA)+FRT(JPB)-FRT(JRA)-FRT(JRB)
      QKC(I)=EXP(DLFRT(I))/(82.05*T)**JHDBDY(I)
      GK(I,1)=RATES(1,I)*T**RATES(2,I)*EXP(RATES(3,I)/T)
     1*DAMP(I)
      QK(I,2)=GK(I,1)/QKC(I)
      IF(MH.GT.0) WRITE(6,1)QK(I,1),QK(I,2),QKC(I)
    1 FORMAT(8E15.7)
   18 CONTINUE
      DO 25 I=1,N
      B(I)=.0
      DO 25 J=1,N
   25 A(I,J)=C.
      CALL FGX
      RETURN
      END
```

```
• ELT FGX,1,720512, 51622      , 1

                    SUBROUTINE FGX
                    COMMON/FCG/T,H,ALPHI(9),KPSI,A(8,8),Y(9),B(8),RHOB,TIMEP,DT       FGX
                    COMMON/FAR/RHO(6),QK(17,2),PL(17,10),X
                    PL(  1,  1) = QK(  1,  1)*RHO(1)*Y(  1)
                    PL(  1,  2) = QK(  1,  1)*RHO(1)*Y(  7)
                    PL(  1,  6) = QK(  1,  2)*RHO(1)*Y(  5)
                    PL(  1,  7) = QK(  1,  2)*RHO(1)*Y(  4)
                    PL(  2,  1) = QK(  2,  1)*RHO(1)*Y(  1)
                    PL(  2,  2) = QK(  2,  1)*RHO(1)*Y(  7)
                    PL(  2,  6) = 2.*QK(  2,  2)*RHO(1)*Y(  6)
                    PL(  3,  1) = QK(  3,  1)*RHO(1)*Y(  2)
                    PL(  3,  2) = QK(  3,  1)*RHO(1)*Y(  7)
                    PL(  3,  6) = QK(  3,  2)*RHO(1)*Y(  5)
                    PL(  3,  7) = QK(  3,  2)*RHO(1)*Y(  6)
                    PL(  4,  1) = QK(  4,  1)*RHO(1)*Y(  6)
                    PL(  4,  2) = QK(  4,  1)*RHO(1)*Y(  7)
                    PL(  4,  6) = QK(  4,  2)*RHO(1)*Y(  5)
                    PL(  4,  7) = QK(  4,  2)*RHO(1)*Y(  3)
                    PL(  5,  1) = QK(  5,  1)*RHO(1)*Y(  6)
                    PL(  5,  2) = QK(  5,  1)*RHO(1)*Y(  4)
                    PL(  5,  6) = QK(  5,  2)*RHO(1)*Y(  1)
                    PL(  5,  7) = QK(  5,  2)*RHO(1)*Y(  3)
                    PL(  6,  1) = 2.*QK(  6,  1)*RHO(1)*Y(  6)
                    PL(  6,  6) = QK(  6,  2)*RHO(1)*Y(  2)
                    PL(  6,  7) = QK(  6,  2)*RHO(1)*Y(  3)
                    PL(  7,  1) = QK(  7,  1)*RHO(1)*Y(  2)
                    PL(  7,  2) = QK(  7,  1)*RHO(1)*Y(  4)
                    PL(  7,  6) = QK(  7,  2)*RHO(1)*Y(  1)
                    PL(  7,  7) = QK(  7,  2)*RHO(1)*Y(  6)
                    PL(  8,  1) = QK(  8,  1)*RHO(1)*Y(  1)
                    PL(  8,  2) = QK(  8,  1)*RHO(1)*Y(  5)
                    PL(  8,  6) = QK(  8,  2)*RHO(1)*Y(  6)
                    PL(  8,  7) = QK(  8,  2)*RHO(1)*Y(  2)
                    PL(  9,  1) = QK(  9,  1)*RHO(1)*Y(  6)
                    PL(  9,  2) = QK(  9,  1)*RHO(1)*Y(  8)
                    PL(  9,  6) = QK(  9,  2)*RHO(1)*Y(  3)
                    PL(  9,  7) = QK(  9,  2)*RHO(1)*Y(  7)
                    PL( 10,  1) = QK( 10,  1)*RHO(1)*Y(  1)
                    PL( 10,  2) = QK( 10,  1)*RHO(1)*Y(  8)
                    PL( 10,  6) = QK( 10,  2)*RHO(1)*Y(  4)
                    PL( 10,  7) = QK( 10,  2)*RHO(1)*Y(  7)
                    PL( 11,  1) = QK( 11,  1)*RHO(1)*Y(  1)
                    PL( 11,  2) = QK( 11,  1)*RHO(1)*Y(  8)
                    PL( 11,  6) = QK( 11,  2)*RHO(1)*Y(  6)
                    PL( 11,  7) = QK( 11,  2)*RHO(1)*Y(  3)
                    PL( 12,  1) = QK( 12,  1)*RHO(2)*Y(  1)*X
                    PL( 12,  2) = QK( 12,  1)*RHO(2)*Y(  2)*X
                    PL( 12,  6) = QK( 12,  2)*RHO(1)*X
                    PL( 12,  5) = QK( 12,  1)*RHO(2)*Y(  2)*Y(  1)
                    PL( 12, 10) = QK( 12,  2)*RHO(1)*Y(  6)
                    PL( 13,  1) = 2.*QK( 13,  1)*RHO(2)*Y(  2)*X
                    PL( 13,  6) = QK( 13,  2)*RHO(1)*X
                    PL( 13,  5) = QK( 13,  1)*RHO(2)*Y(  2)*Y(  2)
                    PL( 13, 10) = QK( 13,  2)*RHO(1)*Y(  5)
                    PL( 14,  1) = 2.*QK( 14,  1)*RHO(2)*Y(  1)*X
                    PL( 14,  6) = QK( 14,  2)*RHO(1)*X
                    PL( 14,  5) = QK( 14,  1)*RHO(2)*Y(  1)*Y(  1)
                    PL( 14, 10) = QK( 14,  2)*RHO(1)*Y(  4)
```

```
      PL( 15,  1) = QK( 15,  1)*RHO(2)*Y(  6)*X
      PL( 15,  2) = QK( 15,  1)*RHO(2)*Y(  1)*X
      PL( 15,  6) = QK( 15,  2)*RHO(1)*X
      PL( 15,  5) = QK( 15,  1)*RHO(2)*Y(  1)*Y(  6)
      PL( 15, 10) = QK( 15,  2)*RHO(1)*Y(  3)
      PL( 16,  1) = QK( 16,  1)*RHO(2)*Y(  1)*X
      PL( 16,  2) = QK( 16,  1)*RHO(2)*Y(  5)*X
      PL( 16,  6) = QK( 16,  2)*RHO(1)*X
      PL( 16,  5) = QK( 16,  1)*RHO(2)*Y(  5)*Y(  1)
      PL( 16, 10) = QK( 16,  2)*RHO(1)*Y(  7)
      PL( 17,  1) = 2.*QK( 17,  1)*RHO(2)*Y(  6)*X
      PL( 17,  6) = QK( 17,  2)*RHO(1)*X
      PL( 17,  5) = QK( 17,  1)*RHO(2)*Y(  6)*Y(  6)
      PL( 17, 10) = QK( 17,  2)*RHO(1)*Y(  8)
      BB=  PL(  1,  2)+   PL(  2,  2)+   PL(  5,  7)+   PL(  7,  7)
      BB=BB+   PL(  8,  2)+   PL( 10,  2)+   PL( 11,  2)+   PL( 12,  2)
      BB=BB+   PL( 12,  5)+2.*PL( 14,  1)+2.*PL( 14,  5)+   PL( 15,  1)
      AA=  PL( 12, 10)+2.*PL( 14, 10)+   PL( 15, 10)+   PL( 16, 10)
      BB=BB+   PL( 15,  5)+   PL( 16,  2)+   PL( 16,  5)
      A(  1,  1) = AA - BB
      AA=  PL(  7,  2)+   PL(  8,  6)+   PL( 12, 10)+2.*PL( 14, 10)
      BB=  PL( 12,  1)+   PL( 12,  5)+2.*PL( 14,  5)+   PL( 15,  5)
      AA=AA+   PL( 15, 10)+   PL( 16, 10)
      BB=BB+   PL( 16,  5)
      A(  1,  2) = AA - BB
      AA=  PL( 11,  6)+   PL( 12, 10)+2.*PL( 14, 10)+   PL( 15,  6)
      BB=  PL(  5,  6)+   PL( 12,  5)+2.*PL( 14,  5)+   PL( 15,  5)
      AA=AA+   PL( 15, 10)+   PL( 16, 10)
      BB=BB+   PL( 16,  5)
      A(  1,  3) = AA - BB
      AA=  PL(  1,  6)+   PL(  5,  1)+   PL(  7,  1)+   PL( 10,  7)
      AA=AA+   PL( 12, 10)+2.*PL( 14,  6)+2.*PL( 14, 10)+   PL( 15, 10)
      AA=AA+   PL( 16, 10)
      BB=  PL( 12,  5)+2.*PL( 14,  5)+   PL( 15,  5)+   PL( 16,  5)
      A(  1,  4) = AA - BB
      BB=  PL(  8,  1)+   PL( 12,  5)+2.*PL( 14,  5)+   PL( 15,  5)
      AA=  PL(  1,  7)+   PL( 12, 10)+2.*PL( 14, 10)+   PL( 15, 10)
      AA=AA+   PL( 16, 10)
      BB=BB+   PL( 16,  1)+   PL( 16,  5)
      A(  1,  5) = AA - BB
      AA=  PL(  2,  6)+   PL(  5,  2)+   PL(  8,  7)+   PL( 11,  7)
      BB=  PL(  7,  6)+   PL( 12,  5)+2.*PL( 14,  5)+   PL( 15,  2)
      AA=AA+   PL( 12,  6)+   PL( 12, 10)+2.*PL( 14, 10)+   PL( 15, 10)
      AA=AA+   PL( 16, 10)
      BB=BB+   PL( 15,  5)+   PL( 16,  5)
      A(  1,  6) = AA - BB
      BB=  PL(  1,  1)+   PL(  2,  1)+   PL( 12,  5)+2.*PL( 14,  5)
      AA=  PL( 10,  6)+   PL( 12, 10)+2.*PL( 14, 10)+   PL( 15, 10)
      AA=AA+   PL( 16,  6)+   PL( 16, 10)
      BB=BB+   PL( 15,  5)+   PL( 16,  5)
      A(  1,  7) = AA - BB
      BB=  PL( 10,  1)+   PL( 11,  1)+   PL( 12,  5)+2.*PL( 14,  5)
      AA=  PL( 12, 10)+2.*PL( 14, 10)+   PL( 15, 10)+   PL( 16, 10)
      BB=BB+   PL( 15,  5)+   PL( 16,  5)
      A(  1,  8) = AA - BB
      AA=  PL(  7,  7)+   PL(  8,  2)+   PL( 12, 10)+2.*PL( 13, 10)
      BB=  PL( 12,  2)+   PL( 12,  5)+2.*PL( 13,  5)
      A(  2,  1) = AA - BB
      BB=  PL(  3,  2)+   PL(  6,  7)+   PL(  7,  2)+   PL(  8,  6)
      AA=  PL( 12, 10)+2.*PL( 13, 10)
```

```
BB=BB+   PL( 12,  1)+   PL( 12,  5)+2.*PL( 13,  1)+2.*PL( 13,  5)
A(  2,  2) = AA - BB
AA=   PL( 12, 10)+2.*PL( 13, 10)
BB=   PL(  6,  6)+   PL( 12,  5)+2.*PL( 13,  5)
A(  2,  3) = AA - BB
AA=   PL( 12, 10)+2.*PL( 13, 10)
BB=   PL(  7,  1)+   PL( 12,  5)+2.*PL( 13,  5)
A(  2,  4) = AA - BB
AA=   PL(  3,  7)+   PL(  8,  1)+   PL( 12, 10)+2.*PL( 13,  6)
AA=AA+2.*PL( 13, 10)
BB=   PL( 12,  5)+2.*PL( 13,  5)
A(  2,  5) = AA - BB
AA=   PL(  3,  6)+   PL(  6,  1)+   PL(  7,  6)+   PL( 12,  6)
AA=AA+   PL( 12, 10)+2.*PL( 13, 10)
BB=   PL(  8,  7)+   PL( 12,  5)+2.*PL( 13,  5)
A(  2,  6) = AA - BB
AA=   PL( 12, 10)+2.*PL( 13, 10)
BB=   PL(  3,  1)+   PL( 12,  5)+2.*PL( 13,  5)
A(  2,  7) = AA - BB
AA=   PL( 12, 10)+2.*PL( 13, 10)
BB=   PL( 12,  5)+2.*PL( 13,  5)
A(  2,  8) = AA - BB
AA=   PL( 11,  2)+   PL( 15,  1)+   PL( 15,  5)
BB=   PL(  5,  7)+   PL( 15, 10)
A(  3,  1) = AA - BB
AA=   PL( 15,  5)
BB=   PL(  6,  7)+   PL( 15, 10)
A(  3,  2) = AA - BB
BB=   PL(  4,  6)+   PL(  5,  6)+   PL(  6,  6)+   PL(  9,  7)
AA=   PL( 15,  5)
BB=BB+   PL( 11,  6)+   PL( 15,  6)+   PL( 15, 10)
A(  3,  3) = AA - BB
AA=   PL(  5,  1)+   PL( 15,  5)
BB=   PL( 15, 10)
A(  3,  4) = AA - BB
AA=   PL( 15,  5)
BB=   PL(  4,  7)+   PL( 15, 10)
A(  3,  5) = AA - BB
AA=   PL(  4,  2)+   PL(  5,  2)+   PL(  6,  1)+   PL(  9,  2)
AA=AA+   PL( 15,  2)+   PL( 15,  5)
BB=   PL( 11,  7)+   PL( 15, 10)
A(  3,  6) = AA - BB
AA=   PL(  4,  1)+   PL( 15,  5)
BB=   PL(  9,  6)+   PL( 15, 10)
A(  3,  7) = AA - BB
AA=   PL(  9,  1)+   PL( 11,  1)+   PL( 15,  5)
BB=   PL( 15, 10)
A(  3,  8) = AA - BB
AA=   PL(  1,  2)+   PL(  5,  7)+   PL(  7,  7)+   PL( 10,  2)
AA=AA+   PL( 14,  1)+   PL( 14,  5)
BB=   PL( 14, 10)
A(  4,  1) = AA - BB
AA=   PL( 14,  5)
BB=   PL(  7,  2)+   PL( 14, 10)
A(  4,  2) = AA - BB
AA=   PL(  5,  6)+   PL( 14,  5)
BB=   PL( 14, 10)
A(  4,  3) = AA - BB
BB=   PL(  1,  6)+   PL(  5,  1)+   PL(  7,  1)+   PL( 10,  7)
AA=   PL( 14,  5)
```

```
      BB=BB+   PL( 14,  6)+   PL( 14, 10)
      A(  4,  4) = AA - BB
      AA=   PL( 14,  5)
      BB=   PL(  1,  7)+   PL( 14, 10)
      A(  4,  5) = AA - BB
      AA=   PL(  7,  6)+   PL( 14,  5)
      BB=   PL(  5,  2)+   PL( 14, 10)
      A(  4,  6) = AA - BB
      AA=   PL(  1,  1)+   PL( 14,  5)
      BB=   PL( 10,  6)+   PL( 14, 10)
      A(  4,  7) = AA - BB
      AA=   PL( 10,  1)+   PL( 14,  5)
      BB=   PL( 14, 10)
      A(  4,  8) = AA - BB
      AA=   PL(  1,  2)+   PL( 13,  5)+   PL( 16, 10)
      BB=   PL(  8,  2)+   PL( 13, 10)+   PL( 16,  2)+   PL( 16,  5)
      A(  5,  1) = AA - BB
      AA=   PL(  3,  2)+   PL(  8,  6)+   PL( 13,  1)+   PL( 13,  5)
      AA=AA+   PL( 16, 10)
      B3=   PL( 13, 10)+   PL( 16,  5)
      A(  5,  2) = AA - BB
      AA=   PL( 13,  5)+   PL( 16, 10)
      BB=   PL(  4,  6)+   PL( 13, 10)+   PL( 16,  5)
      A(  5,  3) = AA - BB
      AA=   PL( 13,  5)+   PL( 16, 10)
      BB=   PL(  1,  6)+   PL( 13, 10)+   PL( 16,  5)
      A(  5,  4) = AA - BB
      BB=   PL(  1,  7)+   PL(  3,  7)+   PL(  4,  7)+   PL(  8,  1)
      AA=   PL( 13,  5)+   PL( 16, 10)
      BB=BB+   PL( 13,  6)+   PL( 13, 10)+   PL( 16,  1)+   PL( 16,  5)
      A(  5,  5) = AA - BB
      AA=   PL(  4,  2)+   PL(  8,  7)+   PL( 13,  5)+   PL( 16, 10)
      BB=   PL(  3,  6)+   PL( 13, 10)+   PL( 16,  5)
      A(  5,  6) = AA - BB
      AA=   PL(  1,  1)+   PL(  3,  1)+   PL(  4,  1)+   PL( 13,  5)
      AA=AA+   PL( 16,  6)+   PL( 16, 10)
      BB=   PL( 13, 10)+   PL( 16,  5)
      A(  5,  7) = AA - BB
      AA=   PL( 13,  5)+   PL( 16, 10)
      BB=   PL( 13, 10)+   PL( 16,  5)
      A(  5,  8) = AA - BB
      AA=2.*PL(  2,  2)+   PL(  5,  7)+   PL(  8,  2)+   PL( 11,  2)
      BB=   PL(  7,  7)+   PL( 12, 10)+   PL( 15,  1)+   PL( 15,  5)
      AA=AA+   PL( 12,  2)+   PL( 12,  5)+   PL( 15, 10)+2.*PL( 17, 10)
      BB=BB+2.*PL( 17,  5)
      A(  6,  1) = AA - BB
      AA=   PL(  3,  2)+2.*PL(  6,  7)+   PL(  7,  2)+   PL( 12,  1)
      AA=AA+   PL( 12,  5)+   PL( 15, 10)+2.*PL( 17, 10)
      BB=   PL(  8,  6)+   PL( 12, 10)+   PL( 15,  5)+2.*PL( 17,  5)
      A(  6,  2) = AA - BB
      AA=   PL(  4,  6)+   PL(  5,  6)+2.*PL(  6,  6)+   PL(  9,  7)
      AA=AA+   PL( 12,  5)+   PL( 15,  6)+   PL( 15, 10)+2.*PL( 17, 10)
      BB=   PL( 11,  6)+   PL( 12, 10)+   PL( 15,  5)+2.*PL( 17,  5)
      A(  6,  3) = AA - BB
      AA=   PL(  7,  1)+   PL( 12,  5)+   PL( 15, 10)+2.*PL( 17, 10)
      BB=   PL(  5,  1)+   PL( 12, 10)+   PL( 15,  5)+2.*PL( 17,  5)
      A(  6,  4) = AA - BB
      AA= . PL(  4,  7)+   PL(  8,  1)+   PL( 12,  5)+   PL( 15, 10)
      AA=AA+2.*PL( 17, 10)
      BB=   PL(  3,  7)+   PL( 12, 10)+   PL( 15,  5)+2.*PL( 17,  5)
      A( 6, 5) = AA - BB
      BB=2.*PL( 2, 6)+  PL( 3, 6)+  PL( 4, 2)+  PL( 5, 2)
      BB=BB+2.*PL( 6, 1)+  PL( 7, 6)+  PL( 8, 7)+  PL( 9, 2)
      BB=BB+  PL( 11, 7)+  PL( 12, 6)+  PL( 12, 10)+  PL( 15, 2)
      AA=  PL( 12, 5)+  PL( 15, 10)+2.*PL( 17, 10)
      BB=BB+  PL( 15, 5)+2.*PL( 17, 1)+2.*PL( 17, 5)
      A( 6, 6) = AA - BB
      AA=2.*PL( 2, 1)+  PL( 3, 1)+  PL( 9, 6)+  PL( 12, 5)
      AA=AA+  PL( 15, 10)+2.*PL( 17, 10)
      BB=  PL( 4, 1)+  PL( 12, 10)+  PL( 15, 5)+2.*PL( 17, 5)
      A( 6, 7) = AA - BB
      AA=  PL( 11, 1)+  PL( 12, 5)+  PL( 15, 10)+2.*PL( 17, 6)
      AA=AA+2.*PL( 17, 10)
      BB=  PL( 9, 1)+  PL( 12, 10)+  PL( 15, 5)+2.*PL( 17, 5)
      A( 6, 8) = AA - BB
      AA=  PL( 10, 2)+  PL( 16, 2)+  PL( 16, 5)
      BB=  PL( 1, 2)+  PL( 2, 2)+  PL( 16, 10)
      A( 7, 1) = AA - BB
      AA=  PL( 16, 5)
      BB=  PL( 3, 2)+  PL( 16, 10)
      A( 7, 2) = AA - BB
      AA=  PL( 4, 6)+  PL( 16, 5)
      BB=  PL( 9, 7)+  PL( 16, 10)
      A( 7, 3) = AA - BB
      AA=  PL( 1, 6)+  PL( 16, 5)
      BB=  PL( 10, 7)+  PL( 16, 10)
      A( 7, 4) = AA - BB
      AA=  PL( 1, 7)+  PL( 3, 7)+  PL( 4, 7)+  PL( 16, 1)
      AA=AA+  PL( 16, 5)
      BB=  PL( 16, 10)
      A( 7, 5) = AA - BB
      AA=  PL( 2, 6)+  PL( 3, 6)+  PL( 9, 2)+  PL( 16, 5)
      BB=  PL( 4, 2)+  PL( 16, 10)
      A( 7, 6) = AA - BB
      BB=  PL( 1, 1)+  PL( 2, 1)+  PL( 3, 1)+  PL( 4, 1)
      AA=  PL( 16, 5)
      BB=BB+  PL( 9, 6)+  PL( 10, 6)+  PL( 16, 6)+  PL( 16, 10)
      A( 7, 7) = AA - BB
      AA=  PL( 9, 1)+  PL( 10, 1)+  PL( 16, 5)
      BB=  PL( 16, 10)
      A( 7, 8) = AA - BB
      AA=  PL( 17, 5)
      BB=  PL( 10, 2)+  PL( 11, 2)+  PL( 17, 10)
      A( 8, 1) = AA - BB
      AA=  PL( 17, 5)
      BB=  PL( 17, 10)
      A( 8, 2) = AA - BB
      AA=  PL( 9, 7)+  PL( 11, 6)+  PL( 17, 5)
      BB=  PL( 17, 10)
      A( 8, 3) = AA - BB
      AA=  PL( 10, 7)+  PL( 17, 5)
      BB=  PL( 17, 10)
      A( 8, 4) = AA - BB
      AA=  PL( 17, 5)
      BB=  PL( 17, 10)
      A( 8, 5) = AA - BB
      AA=  PL( 11, 7)+  PL( 17, 1)+  PL( 17, 5)
      BB=.  PL( 9, 2)+  PL( 17, 10)
      A( 8, 6) = AA - BB
      AA=  PL( 9, 6)+  PL( 10, 6)+  PL( 17, 5)
      BB=   PL( 17, 10)
      A(  8,  7) = AA - BB
      BB=   PL(  9,  1)+   PL( 10,  1)+   PL( 11,  1)+   PL( 17,  6)
      AA=   PL( 17,  5)
      BB=BB+   PL( 17, 10)
      A(  8,  8) = AA - BB
      PL(  1,  1)=Y(  7)*PL(  1,  1)
      PL(  1,  6)=Y(  4)*PL(  1,  6)
      PL(  2,  1)=Y(  7)*PL(  2,  1)
      PL(  2,  6)=Y(  6)*PL(  2,  6)/2.
      PL(  3,  1)=Y(  7)*PL(  3,  1)
      PL(  3,  6)=Y(  6)*PL(  3,  6)
      PL(  4,  1)=Y(  7)*PL(  4,  1)
      PL(  4,  6)=Y(  3)*PL(  4,  6)
      PL(  5,  1)=Y(  4)*PL(  5,  1)
      PL(  5,  6)=Y(  3)*PL(  5,  6)
      PL(  6,  1)=Y(  6)*PL(  6,  1)/2.
      PL(  6,  6)=Y(  3)*PL(  6,  6)
      PL(  7,  1)=Y(  4)*PL(  7,  1)
      PL(  7,  6)=Y(  6)*PL(  7,  6)
      PL(  8,  1)=Y(  5)*PL(  8,  1)
      PL(  8,  6)=Y(  2)*PL(  8,  6)
      PL(  9,  1)=Y(  8)*PL(  9,  1)
      PL(  9,  6)=Y(  7)*PL(  9,  6)
      PL( 10,  1)=Y(  8)*PL( 10,  1)
      PL( 10,  6)=Y(  7)*PL( 10,  6)
      PL( 11,  1)=Y(  8)*PL( 11,  1)
      PL( 11,  6)=Y(  3)*PL( 11,  6)
      PL( 12,  1)=Y(  2)*PL( 12,  1)
      PL( 12,  6)=Y(  6)*PL( 12,  6)
      PL( 13,  1)=Y(  2)*PL( 13,  1)/2.
      PL( 13,  6)=Y(  5)*PL( 13,  6)
      PL( 14,  1)=Y(  1)*PL( 14,  1)/2.
      PL( 14,  6)=Y(  4)*PL( 14,  6)
      PL( 15,  1)=Y(  1)*PL( 15,  1)
      PL( 15,  6)=Y(  3)*PL( 15,  6)
      PL( 16,  1)=Y(  5)*PL( 16,  1)
      PL( 16,  6)=Y(  7)*PL( 16,  6)
      PL( 17,  1)=Y(  6)*PL( 17,  1)/2.
      PL( 17,  6)=Y(  8)*PL( 17,  6)
      BB=   PL(  1,  1)+   PL(  2,  1)+   PL(  5,  6)+   PL(  7,  6)
      AA=   PL(  1,  6)+   PL(  2,  6)+   PL(  5,  1)+   PL(  7,  1)
      BB=BB+   PL(  8,  1)+   PL( 10,  1)+   PL( 11,  1)+   PL( 12,  1)
      AA=AA+   PL(  8,  6)+   PL( 10,  6)+   PL( 11,  6)+   PL( 12,  6)
      AA=AA+2.*PL( 14,  6)+   PL( 15,  6)+   PL( 16,  6)
      BB=BB+2.*PL( 14,  1)+   PL( 15,  1)+   PL( 16,  1)
      B(  1) = AA - BB
      BB=   PL(  3,  1)+   PL(  6,  6)+   PL(  7,  1)+   PL(  8,  6)
      AA=   PL(  3,  6)+   PL(  6,  1)+   PL(  7,  6)+   PL(  8,  1)
      AA=AA+   PL( 12,  6)+2.*PL( 13,  6)
      BB=BB+   PL( 12,  1)+2.*PL( 13,  1)
      B(  2) = AA - BB
      AA=   PL(  4,  1)+   PL(  5,  1)+   PL(  6,  1)+   PL(  9,  1)
      BB=   PL(  4,  6)+   PL(  5,  6)+   PL(  6,  6)+   PL(  9,  6)
      AA=AA+   PL( 11,  1)+   PL( 15,  1)
      BB=BB+   PL( 11,  6)+   PL( 15,  6)
      B(  3) = AA - BB
      AA=   PL(  1,  1)+   PL(  5,  6)+   PL(  7,  6)+   PL( 10,  1)
      BB=   PL(  1,  6)+   PL(  5,  1)+   PL(  7,  1)+   PL( 10,  6)
      AA=AA+   PL( 14,  1)
      BB=BB+    PL( 14,  6)
      B(  4) = AA - BB
      AA=   PL(  1,  1)+   PL(  3,  1)+   PL(  4,  1)+   PL(  8,  6)
      BB=   PL(  1,  6)+   PL(  3,  6)+   PL(  4,  6)+   PL(  8,  1)
      AA=AA+    PL( 13,  1)+   PL( 16,  6)
      BB=BB+    PL( 13,  6)+   PL( 16,  1)
      B(  5) = AA - BB
      BB=2.*PL(  2,  6)+   PL(  3,  6)+   PL(  4,  1)+   PL(  5,  1)
      AA=2.*PL(  2,  1)+   PL(  3,  1)+   PL(  4,  6)+   PL(  5,  6)
      AA=AA+2.*PL(  6,  6)+   PL(  7,  1)+   PL(  8,  1)+   PL(  9,  6)
      BB=BB+2.*PL(  6,  1)+   PL(  7,  6)+   PL(  8,  6)+   PL(  9,  1)
      AA=AA+    PL( 11,  1)+   PL( 12,  1)+   PL( 15,  6)+2.*PL( 17,  6)
      BB=BB+    PL( 11,  6)+   PL( 12,  6)+   PL( 15,  1)+2.*PL( 17,  1)
      B(  6) = AA - BB
      AA=   PL(  1,  6)+   PL(  2,  6)+   PL(  3,  6)+   PL(  4,  6)
      BB=   PL(  1,  1)+   PL(  2,  1)+   PL(  3,  1)+   PL(  4,  1)
      AA=AA+    PL(  9,  1)+   PL( 10,  1)+   PL( 16,  1)
      BB=BB+    PL(  9,  6)+   PL( 10,  6)+   PL( 16,  6)
      B(  7) = AA - BB
      AA=   PL(  9,  6)+   PL( 10,  6)+   PL( 11,  6)+   PL( 17,  1)
      BB=   PL(  9,  1)+   PL( 10,  1)+   PL( 11,  1)+   PL( 17,  6)
      B(  8) = AA - BB
   50 RETURN
      END
```

```
* ELT FLUX,1,720512, 51593      , 1
      SUBROUTINE FLUX
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLW(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
     1,IDA(200),TCHEM(200),WDT(9,200)
      COMMON/ZC/WTMCLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TWX(7,4)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
     1OR,USTCF,RAY,RG,AK,AKA
      COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      COMMON/ZL/GLX(7,4),          TCX(7,4),QL,TC,PRB,GAMW,RUC,RUCX(7,4)
     1,SCB,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
     2,GG,UB(21),UBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
     3,VWS(21),RWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,GGG(21)
     4,CPB,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
     5,NCL,NP,FYE,ANP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),HC,HCB(21)
    1 FORMAT(PE15.7)
      IF(ISOBAT.GT.3)GO TO 100
C     TRANSPIRED WALL
      IF(NTYPE.EQ.0)GO TO 30
      PSIWA=PSIWA+RUC*DX
      GO TO 40
   30 PSIWA=SQRT(PSIWA**2+2.*RUC*YW*DX)
   40 CONTINUE
      IF(NEW.EQ.0)GO TO 150
      DEER=PSIWA-PSI(LR)
      IF(DEER.LT..5*DELPSI)GO TO 200
C     ADD A GRID POINT TO THE FLOW FIELD
      DO  70 I=NPSI,MP
      TELAP(I)=TELAP(NPSI)
      U(I)=U(NPSI)
      H(I)=H(NPSI)
      T(I)=T(NPSI)
      IF(ICHEM.NE.3)GO TO  71
      FIX(I)=FIX(NPSI)
      IDA(I)=IDA(NPSI)
      THLD(I)=THLD(NPSI)
      HHLD(I)=HHLD(NPSI)
   71 DO  72 J=1,NSPC
   72 YSPEC(J,I)=YSPEC(J,NPSI)
      IF(ICHEM.NE.2)GO TO  70
      DO  73 J=1,NEL
   73 ALPHA(J,I)=ALPHA(J,NPSI)
   70 CONTINUE
      NPSI=NPSI+1
      GO TO 180
  100 CONTINUE
C     SLOT COOLED WALL
      NNS=NSLCR+1
      IF(NNS.GT.NSLOT)GO TO 200
      CALL HOFT(TCS(NNS),HCS(NNS),ALC,CPC)
      IF(X.LT.XS(NNS))GO TO 200
      IF(NTYPE.EQ.0)GO TO 110
      DPS=RUCF(NNS)*SH(NNS)
      GO TO 120
  110 CONTINUE
      DPS=SQRT(PSIWA**2+RUCF(NNS)*SH(NNS)*(SH(NNS)+2.*YW))-PSIWA
  120 CONTINUE
      PSIWA=PSIWA+DPS
      IF(NEW.EQ.0)GO TO 150
      HAD=PSIW+.5*DELPSI
      IF(PSIWA.LT.HAD)GO TO 200
      ADD=1.+(PSIWA-PSI(1))/DELPSI
      NAD=ADD
      GAD=NAD
      IF((ADD-GAD).GT..5)NAD=NAD+1
       NADD=NAD-NPSI
      NAMP=MP+NADD
      DO 170 I=MPSI,NAMP
      TELAP(I)=.0
      U(I)=UC(NNS)
      H(I)=HCS(NNS)
      T(I)=TCS(NNS)
      IF(ICHEM.NE.3)GO TO 171
      FIX(I)=XP(5)
      IDA(I)=C
  171 DO 172 J=1,NSPC
  172 YSPEC(J,I)=ALC(J)
      IF(ICHEM.NE.2)GO TO 170
      DO 173 J=1,NEL
  173 ALPHA(J,I)=ELC(J)
  170 CONTINUE
      NPSI=MPSI+NADD
  180 CONTINUE
      LR=NPSI
      MPSI=LR+1
      PSIW=PSI(LR)
      MP=MPSI+1
      MQ=MP+1
      MR=2*MPSI
      IF(MR.GT.MF)MR=MF
      GO TO 200
  150 PSIW=PSIWA
  200 CONTINUE
      IF(MY.GT.,0)WRITE(6,2)IOUT,NADD,NAD,ADD,HAD,PSIWA,PSIW,DEER,DPS
    2 FORMAT(6X,  9HFROM FLUX,3I5,6E15.7)
      RETURN
      END
```

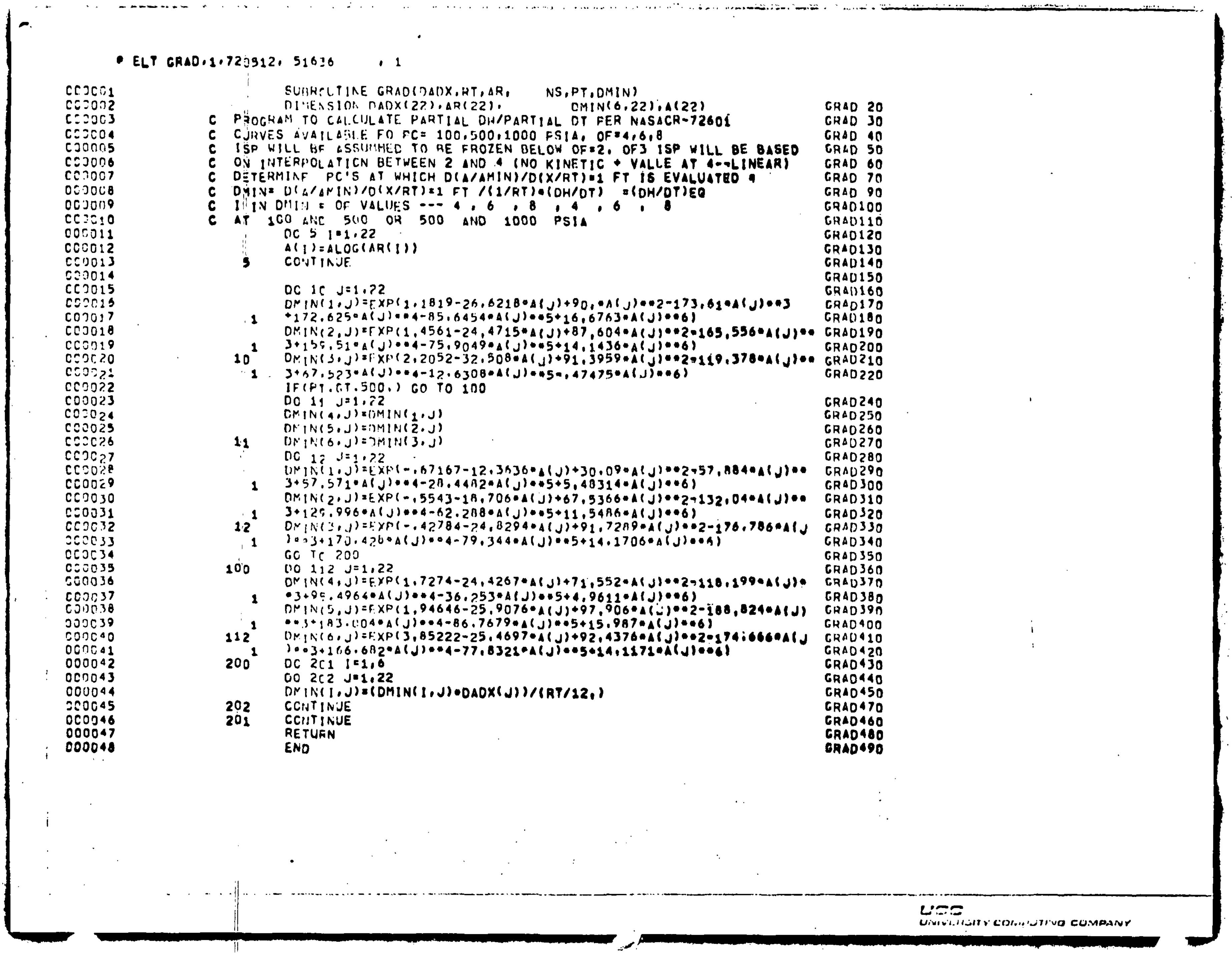
```
• ELT FFCF,1,720512, 51549      , 1

                    FUNCTION FFCF(R)
          C                             FUNCTION TO COMPUTE DUCT FRICTION COEFFICIENT
          C                             GIVEN REYNOLD'S NUMBER
          C
                    X=ALOG(R)
                    X2=X*X
          C
                    FFCF= -.521207056 - .39399288*X + X2*.795810126E-2
                    FFCF=EXP(FFCF)
                    RETURN
                    END
```

```
 ELT FLOW,1,720512, 51550     , 1
              FUNCTION FLOW(PR,F4LD)
C                                   FUNCTION TO COMPUTE UNCHOKED DUCT FLOW
C                                   PARAMETER GIVEN  PRESSURE RATIO + 4FL/D
C
              X2=PR*PR
              X3=PR*X2
              X4=PH*X3
C
              Y=ALCG(F4LD)
              Y2=Y*Y
              Y3=Y*Y2
              Y4=Y*Y3
C
              FLOW=.596304084 - PR*.13283345 - X2*1.2179872 + X3*3.64026368 -
             1     X4*2.650439756 - Y*.16298555 + Y2*.841392545E-4 +
             2 Y3*.141089272E-2 + Y4*.5151692509E-4 + PR*Y*.3602577874 -
             3 PR*Y2*.1814897935E-1 - X2*Y*.72519953 - PR*Y3*.148729387E-3 +
             4 X3*Y*.5108571395 + X2*Y2*.1293878649E-1
              IF(FLOW.LT.0.0)FLOW=0.0
              RETURN
              END
```

```
• ELT FMOLWT,1,720512, 51546    , 1
      FUNCTION FMCLWT(X)
C                           FUNCTION TO COMPUTE MOLECULAR WEIGHT,
C                           GIVEN MIXTURE RATIO   X
C
      X2=X*X
      X3=X*X2
      FMOLWT=2.8672526 + 1.7481442*X -.36021210E-1*X2 +.65282906E-4*X3
      RETURN
      END
```

```
 ELT FVEL,1,720512, 51610      , 1
            SUBROUTINE FVEL(X,F,DF)
      C                                 SUBROUTINE TO COMPUTE TCTAL MASS
      C                                 FLOW VALUES FOR NEWTON ITERATION.
            COMMON /NJEC2/ B
            DATA P/-2,5/
      C
            A=1.0 + 0.2*X*X
      C                                 CALC  DERIVATIVE VALUE
            DF=(1.0-X*X)/(A**4)
      C                                 CALC  FUNCTION VALUE
            F=SQRT(X*X/A) * (A**P) - B
            RETURN
            END
      C                                                                      RTNI 001
      C     ..................................................................RTNI 002
      C                                                                      RTNI 003
      C     SUBRCUTINE RTNI                                                  RTNI 004
      C                                                                      RTNI 005
      C     PURPCSE                                                          RTNI 006
      C        TO SCLVE GENERAL NONLINEAR EQUATIONS OF THE FORM F(X)=0       RTNI 007
      C        BY MEANS OF NEWTON-S ITERATION METHOD.                        RTNI 008
      C                                                                      RTNI 009
      C     USAGE                                                            RTNI 010
      C        CALL RTNI (X,F,DERF,FCT,XST,EPS,IEND,IER)                     RTNI 011
      C        PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.                 RTNI 012
      C                                                                      RTNI 013
      C     DESCRIPTION OF PARAMETERS                                        RTNI 014
      C        X      - RESULTANT ROOT OF EQUATION F(X)=0.                   RTNI 015
      C        F      - RESULTANT FUNCTION VALUE AT ROOT X.                  RTNI 016
      C        DERF   - RESULTANT VALUE OF DERIVATIVE AT ROCT X.             RTNI 017
      C        FCT    - NAME OF THE EXTERNAL SUBROUTINE USED. IT COMPUTES    RTNI 018
      C                 TO GIVEN ARGUMENT X FUNCTION VALUE F AND DERIVATIVE  RTNI 019
      C                 DERF. ITS PARAMETER LIST MUST BE X,F,DERF.           RTNI 020
      C        XST    - INPUT VALUE WHICH SPECIFIES THE INITIAL GUESS OF     RTNI 021
      C                 THE ROOT X.                                          RTNI 022
      C        FPS    - INPUT VALUE WHICH SPECIFIES THE UPPER ROUND OF THE   RTNI 023
      C                 ERROR OF RESULT X.                                   RTNI 024
      C        IEND   - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.         RTNI 025
      C        IER    - RESULTANT ERROR PARAMETER CODED AS FOLLOWS           RTNI 026
      C                  IER=0 - NO ERROR,                                   RTNI 027
      C                  IER=1 - NO CONVERGENCE AFTER IEND ITERATION STEPS,  RTNI 028
      C                  IER=2 - AT ANY ITERATION STEP DERIVATIVE DERF WAS   RTNI 029
      C                          EQUAL TO ZERO.                              RTNI 030
      C                                                                      RTNI 031
      C     REMARKS                                                          RTNI 032
      C        THE PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER=2   RTNI 033
      C        IF AT ANY ITERATION STEP DERIVATIVE OF F(X) IS EQUAL TO 0.    RTNI 034
      C        POSSIBLY THE PROCEDURE WOULD BY SUCCESSFUL IF IT IS STARTED   RTNI 035
      C        ONCE MORE WITH ANOTHER INITIAL GUESS XST.                     RTNI 036
      C                                                                      RTNI 037
      C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    RTNI 038
      C        THE EXTERNAL SUBROUTINE FCT(X,F,DERF) MUST BB FURNISHED       RTNI 039
      C        BY THE USER.                                                  RTNI 040
      C                                                                      RTNI 041
      C     METHOD                                                           RTNI 042
      C        SOLUTION OF EQUATION F(X)=0 IS DONE BY MEANS OF NEWTON-S      RTNI 043
      C        ITERATION METHOD. WHICH STARTS AT THE INITIAL GUESS XST OF    RTNI 044
      C        A ROCT X. CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF       RTNI 045
```

```
C        F(X) AT ROOT X IS NOT EQUAL TO ZERO, ONE ITERATION STEP        RTNI 046
C        REQUIRES ONE EVALUATION OF F(X) AND ONE EVALUATION OF THE      RTNI 047
C        DERIVATIVE OF F(X), FOR TEST ON SATISFACTORY ACCURACY SEE      RTNI 048
C        FORMULAE (2) OF MATHEMATICAL DESCRIPTION.                      RTNI 049
C        FOR REFERENCE, SEE R. ZURMUFHL, PRAKTISCHE MATHEMATIK FUER     RTNI 050
C        INGENIEURE UND PHYSIKER, SPRINGER, BERLIN/GOETTINGEN/          RTNI 051
C        HEIDELBERG, 1963, PP.12-17.                                    RTNI 052
C                                                                       RTNI 053
C     ..................................................................RTNI 054
C                                                                       RTNI 055
```



```
● ELT GRAD,1,720512, 51636      , 1

             SUBROUTINE GRAD(DADX,RT,AR,     NS,PT,DMIN)
             DIMENSION DADX(22),AR(22),         DMIN(6,22),A(22)                 GRAD 20
C  PROGRAM TO CALCULATE PARTIAL DH/PARTIAL DT PER NASACR-72601                   GRAD 30
C  CURVES AVAILABLE FO PC= 100,500,1000 PSIA, OF=4,6,8                           GRAD 40
C  ISP WILL BE ASSUMMED TO BE FROZEN BELOW OF=2, OF3 ISP WILL BE BASED           GRAD 50
C  ON INTERPOLATION BETWEEN 2 AND 4 (NO KINETIC + VALLE AT 4--LINEAR)            GRAD 60
C  DETERMINE  PC'S AT WHICH D(A/AMIN)/D(X/RT)=1 FT IS EVALUATED =                GRAD 70
C  DMIN= D(A/AMIN)/D(X/RT)=1 FT /(1/RT)*(DH/DT)  =(DH/DT)EQ                      GRAD 90
C  I IN DMIN = OF VALUES --- 4 , 6  , 8  , 4  , 6  ,  8                          GRAD100
C  AT  100 AND  500  OR  500  AND  1000  PSIA                                    GRAD110
             DO 5 I=1,22                                                         GRAD120
             A(I)=ALOG(AR(I))                                                    GRAD130
       5     CONTINUE                                                            GRAD140
                                                                                 GRAD150
             DO 10 J=1,22                                                        GRAD160
             DMIN(1,J)=EXP(1,1819-26,6218*A(J)+90,*A(J)**2-173,61*A(J)**3        GRAD170
       1     +172,625*A(J)**4-85,6454*A(J)**5+16,6763*A(J)**6)                   GRAD180
             DMIN(2,J)=EXP(1,4561-24,4715*A(J)+87,604*A(J)**2-165,556*A(J)**     GRAD190
       1     3+159,51*A(J)**4-75,9049*A(J)**5+14,1436*A(J)**6)                   GRAD200
      10     DMIN(3,J)=EXP(2,2052-32,508*A(J)+91,3959*A(J)**2-119,378*A(J)**     GRAD210
       1     3+67,523*A(J)**4-12,6308*A(J)**5-,47475*A(J)**6)                    GRAD220
             IF(PT.GT.500,) GO TO 100
             DO 11 J=1,22                                                        GRAD240
             DMIN(4,J)=DMIN(1,J)                                                 GRAD250
             DMIN(5,J)=DMIN(2,J)                                                 GRAD260
      11     DMIN(6,J)=DMIN(3,J)                                                 GRAD270
             DO 12 J=1,22                                                        GRAD280
             DMIN(1,J)=EXP(-,67167-12,3636*A(J)+30,09*A(J)**2-57,884*A(J)**      GRAD290
       1     3+57,571*A(J)**4-28,4482*A(J)**5+5,48314*A(J)**6)                   GRAD300
             DMIN(2,J)=EXP(-,5543-18,706*A(J)+67,5366*A(J)**2-132,04*A(J)**      GRAD310
       1     3+129,996*A(J)**4-62,288*A(J)**5+11,5486*A(J)**6)                   GRAD320
      12     DMIN(3,J)=EXP(-,42784-24,8294*A(J)+91,7289*A(J)**2-176,786*A(J      GRAD330
       1     )**3+170,426*A(J)**4-79,344*A(J)**5+14,1706*A(J)**6)                GRAD340
             GO TO 200                                                           GRAD350
     100     DO 112 J=1,22                                                       GRAD360
             DMIN(4,J)=EXP(1,7274-24,4267*A(J)+71,552*A(J)**2-118,199*A(J)*      GRAD370
       1     *3+95,4964*A(J)**4-36,253*A(J)**5+4,9611*A(J)**6)                   GRAD380
             DMIN(5,J)=EXP(1,94646-25,9076*A(J)+97,906*A(J)**2-188,824*A(J)      GRAD390
       1     **3+183,004*A(J)**4-86,7679*A(J)**5+15,987*A(J)**6)                 GRAD400
     112     DMIN(6,J)=EXP(3,85222-25,4697*A(J)+92,4376*A(J)**2-174,666*A(J      GRAD410
       1     )**3+166,682*A(J)**4-77,8321*A(J)**5+14,1171*A(J)**6)               GRAD420
     200     DO 201 I=1,6                                                        GRAD430
             DO 202 J=1,22                                                       GRAD440
             DMIN(I,J)=(DMIN(I,J)*DADX(J))/(RT/12,)                              GRAD450
     202     CONTINUE                                                            GRAD470
     201     CONTINUE                                                            GRAD460
             RETURN                                                              GRAD480
             END                                                                 GRAD490
```

UCC
UNIVERSITY COMPUTING COMPANY

```
 • ELT GRID,1,720512, 51597    , 1

          SUBROUTINE GRID
C         DUCTED COMPLETE COMBUSTION DECK FOR CNHM-AIR
          COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
         1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
         2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
         3,WTMIX(200),HT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
         4,FEE(200)
          COMMON/EXTRA/JAN(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
         1,IDA(200),TCHEM(200),WDT(9,200)
          COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
         1,GSCALE,TWX(7,4)
          COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
         1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
         2,ISHATY,MJ,MK,ML,MM,MN,MO,NSLOT
         3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
         4,NEW,MQ,MR,LN,NNT,JR,NSLCR
          COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HGT
         1UR,USTCR,RAY,RG,AK,AKA
          COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
          COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
       59 IF(PSI(LR)-PSI(MHALF))121,1500,47
       47 MINIT=MINIT+1
          MHALF=2*MINIT-1
          GO TC 59
C
C         TEST FOR ZERO SLOPE AT LOWER EDGE
      121 IF(PSI(1).LT.DELPSI)GO TO 998
       49 IF(ABS(U(2)-U(1))/U(1)-.001) 51,51,120
       51 IF(ABS(H(2)-H(1))/H(1)-.001) 52,52,120
       52 IF(ICHEM.EQ.2)GO TO 200
          DO 202 J=1,NSPC
          IF(YSPEC(J,1).LT..001)GO TO 202
          IF(ABS(YSPEC(J,2)/YSPEC(J,1)-1.).GT..001)GO TO 120
      202 CONTINUE
          GO TC 201
      200 DO 53 J=1,NEL
          IF(ALPHA(J,1).LT..001)GO TO 53
          IF(ABS(ALPHA(J,2)/ALPHA(J,1)-1.).GT..001)GO TO 120
       53 CONTINUE
      201 CONTINUE
          GO TC 998
      120 DO 122 I=1,MPSI
          K=MPSI+2-I
          KM=K-1
          TELAP(K)=TELAP(KM)
          H(K)=H(K-1)
          U(K)=U(K-1)
          T(K)=T(K-1)
          IF(ICHEM.NE.2)GO TO 205
          DO 206 J=1,NEL
          ALPHA(J,K)=ALPHA(J,KM)
      206 CONTINUE
          GO TC 122
      205 CONTINUE
          IF(ICHEM.NE.3)GO TO 203
          HHLD(K)=HHLD(KM)
          IDA(K)=IDA(KM)
          THLD(K)=THLD(KM)
      FIX(K)=FIX(KM)
  203 CONTINUE
      DO 204 J=1,NSPC
  204 YSPEC(J,K)=YSPEC(J,KM)
  122 CONTINUE
      MPSI=MPSI+1
      NPSI=NPSI-1
      LR=LR+1
      DO 126 I=1,MF
  126 PSI(I)=PSI(I)-DELPSI
  998 IF(NEW.NE.0)GO TO 2000
C
C     TEST FOR ZERO SLOPE AT EDGE
C
  999 IF(ABS(U(NPSI)-U(MPSI))/U(MPSI)-.001) 1011,1011,1004
 1011 IF(ABS(H(NPSI)-H(MPSI))/H(MPSI)-.001) 1002,1002,1004
 1002 IF(ICHEM.EG.2)GO TO 300
      DO 302 J=1,NSPC
      IF(YSPEC(J,MPSI).LT..001)GO TO 302
      IF(ABS(YSPEC(J,NPSI)/YSPEC(J,MPSI)-1.).GT..001)GO TO 1004
  302 CONTINUE
      GO TO 301
  300 DO 54 J=1,NEL
      IF(ALPHA(J,MPSI).LT..001)GO TO 54
      IF(ABS(ALPHA(J,NPSI)/ALPHA(J,MPSI)-1.).GT..001)GO TO 1004
   54 CONTINUE
  301 CONTINUE
      GO TO 2000
C
C     EXPAND MESH
C
 1004 MPSI=MPSI+1
      NPSI=MPSI-1
      LR=LR+1
      MP=MPSI+1
      DO 30 I=MPSI,MP
      TELAP(I)=TELAP(NPSI)
      U(I)=U(NPSI)
      H(I)=H(NPSI)
      T(I)=T(NPSI)
      IF(ICHEM.NE.3)GO TO 303
      FIX(I)=FIX(NPSI)
      IDA(I)=IDA(NPSI)
      THLD(I)=THLD(NPSI)
      HHLD(I)=HHLD(NPSI)
  303 CONTINUE
      DO 304 J=1,NSPC
  304 YSPEC(J,I)=YSPEC(J,NPSI)
      IF(ICHEM.NE.2)GO TO 30
      DO 306 J=1,NEL
      ALPHA(J,I)=ALPHA(J,NPSI)
  306 CONTINUE
   30 CONTINUE
      GO TO 2000
C
C     HALVE MESH
C
 1500 IFINIS=0
      DELPSI=DELPSI+DELPSI
      DO 1600 I=1,MINIT
      KM=2*I-1
      TELAP(I)=TELAP(KM)
      U(I)=U(2*I-1)
      H(I)=H(2*I-1)
      T(I)=T(2*I-1)
      IF(ICHEM.EQ.2)GO TO 400
      IF(ICHEM.NE.3)GO TO 403
      FIX(I)=FIX(KM)
      THLD(I)=THLD(KM)
      HHLD(I)=HHLD(KM)
      IDA(I)=IDA(KM)
  403 CONTINUE
      DO  404 J=1,NSPC
      YSPEC(J,I)=YSPEC(J,KM)
  404 CONTINUE
      GO TO 405
  400 CONTINUE
      DO 406 J=1,NEL
      ALPHA(J,I)=ALPHA(J,2*I-1)
  406 CONTINUE
  405 CONTINUE
      XI=I-1
 1600 PSI(I)=XI*DELPSI+PSI(1)
      LR=MINIT
      IF(NEW.NE.0)MINIT=MINIT+1
   28 MPSI=MINIT
      NPSI=MPSI-1
      MR=MPSI+2
      DO 1700 I=MINIT,MR
      TELAP(I)=TELAP(MPSI)
      U(I)=U(MPSI)
      H(I)=H(MPSI)
      T(I)=T(MPSI)
      IF(ICHEM.EQ.2)GO TO 500
      IF(ICHEM.NE.3)GO TO 503
      FIX(I)=FIX(MPSI)
      THLD(I)=THLD(MPSI)
      HHLD(I)=HHLD(MPSI)
      IDA(I)=IDA(MPSI)
  503 CONTINUE
      DO  504 J=1,NSPC
      YSPEC(J,I)=YSPEC(J,MPSI)
  504 CONTINUE
      GO TO 505
  500 CONTINUE
      DO 506 J=1,NEL
      ALPHA(J,I)=ALPHA(J,MPSI)
  506 CONTINUE
  505 CONTINUE
      XI=I-1
 1700 PSI(I)=XI*DELPSI+PSI(1)
 2000 MP=MPSI+1
      IF(MP.GT.MF)MP=MF
      MQ=MP+1
      IF(MQ.GT.MF)MQ=MF
      MR=2*MPSI
      IF(MR.GT.MF)MR=MF
      RETURN
      END
```

```
• ELT HEAT,1,720512, 51565      , 1

      SUBROUTINE HEAT(LLA,LLB)
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XMU(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TWX(7,4)
      COMMON/ZD/MPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,MST
     1OR,USTOR,RAY,RG,AK,AKA
      COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      DIMENSION CPGAS(200),HE(200)
      EQUIVALENCE (CPGAS,CPBAR),(HE,SMALLH)
      LA=LLA
      LB=LLB
      DO 805 I=LA,LB
      TAUT(I)=.5*U(I)*U(I)
      CPGAS(I)=.0
      IF(KCPT,EQ,1)GO TO 50
C     OBTAIN T BY INVERSION FROM H
      HE(I)=H(I)-TAUT(I)
      KNT=0
   76 TT=T(I)
      GA=.0
      GB=.0
      GC=-HE(I)
      DO 75 J=1,NSPC
      IF(YSPEC(J,I),LE,.0)GO TO 75
      DO 71 L=1,6
      IF(TT.LE.HF(1,L,J))GO TO 73
   71 CONTINUE
   73 GA=GA+YSPEC(J,I)*HF(4,L,J)
      GB=GB+YSPEC(J,I)*HF(3,L,J)
   75 GC=GC+YSPEC(J,I)*HF(2,L,J)
      IF (ABS (GA-0.).LT.  .000001)  GO TO 72
      GD=-.5*GB/GA
      GE=GD*GD-GC/GA
      IF(GE.LT..0)GO TO 72
      T(I)=GD+SQRT(GE)
      GO TO 74
   72 T(I)=-GC/GB
   74 KNT=KNT+1
      IF(KNT.GT,2)GO TO 79
      IF(ABS(T(I)-TT).GT,100,)GO TO 76
   79 IF(KCPT,GT,1)GO TO 90
   50 HE(I)=.C
   90 TM=T(I)
      TS=TM*TM
      TWT=2.*TM
      DO 30 J=1,NGAS
      DO 20 K=1,6
      IF(TM.LE.HF(1,K,J))GO TO 21
```

UCC
UNIVERSITY COMPUTING COMPANY

```
   20 CONTINUE
   21 CPJ=HF(3,K,J)+TWT*HF(4,K,J)
      CPGAS(I)=CPGAS(I)+CPJ*YSPEC(J,I)
      W(J,I)=HF(2,K,J)+TM*HF(3,K,J)+TS*HF(4,K,J)
      IF(KCPT.EQ.1)HE(I)=HE(I)+W(J,I)*YSPEC(J,I)
   30 CONTINUE
   35 IF(KOPT.EQ.1)H(I)=HE(I)+TAUT(I)
  805 CONTINUE
    1 FORMAT(8E15.7)
      RETURN
      END
```

```
* ELT HEATT,1,720512, 51544     , 1

      SUBROUTINE HEATT
C HEAT TRANSFER PROGRAM FOR H2-O2 ROCKET, YIELDS TRANSIENT WALL
C TEMPERATURES WITH OR WITHOUT MULTI-SLOT FILM COOLING, STEADY STATE
C WITH OR WITHOUT MULTI-SLOT FILM COOLING, REGENERATIVE COOLING, LINER,
C OR INJECTOR.
C
      DIMENSION X( 50),DI     ( 50),DO     ( 50),RI     ( 50),XX     (100),
     1     RRI    (100),RRO    (100),ASI    ( 50),ASO    ( 50),V      ( 50),
     2     AXAVE  ( 50),ADM    ( 50),CAP    ( 50),RADADM( 50),ASTOA  ( 50),
     3     HGCVSI( 50),TW     ( 50),SIGMA  ( 50),SUMADM( 50),TOLD   ( 50),
     4     TF     ( 50),ACUM   ( 50),SLOT   ( 21),HSLOT  ( 21),WCOOL  ( 21),
     5     TB     ( 50),DLI    ( 50),DLO    ( 50),RLI    ( 50),RLO    ( 50),
     6     RRLI   (100),ASLO   (100),ASLI   ( 50),ETAEF  ( 50),TL     ( 50),
     7     THL    ( 50),ACUML  ( 50),HPASS  ( 50),WPASS  ( 50),EL     ( 50),
     8     RO     ( 50),RRLO   (100),HG     ( 50),HED    ( 18),HSLT   ( 50),
     9     WCCL   ( 50),TFF    (50)
      WRITE (6,928)
  928 FORMAT ('1',50X,'* * * HEAT TRANSFER INPUT * * *')
   10 READ (5,1100)  HED
C
C INITIALIZE SOME PARAMETERS, MOSTLY FOR OUTPUT PURPOSES.
C
      HRWD = C.
      EPSL = C.
      NPASS = 0
      NINJ = C
      TAU1 = C.
C
C NO. OF NODES, FLAGS FOR STEADY STATE OR TRANSIENT, FILM COOLING,
C REGEN, LINER, INJECTOR,
C
      READ (5,101) NNODE,NTYPFL,NFLMFL,NRGNFL,NLFL,INJFL
      DO 5 I = 1,NNODE
      TB   (I) = 460.
      THL  (I) = 460.
      TL   (I) = 460.
      HSLT (I) = 0.
      WCOL (I) = 0.
      HPASS(I) = 0.
      WPASS(I) = 0.
      ETAEF(I) = 0.
      DLI  (I) = 0.
    5 DLO  (I) = 0.
      READ (5,102) TSTOP,TPRINT
      READ (5,102) TWALLF,TSINKF,CAPPA,EPS,PO,OF
C THIS CONVERTS TSTOP AND TPRINT (FLOATING) TO NCOUNT AND NPRINT (INTEGER)
C FOR STEADY STATE.  INPUT MUST HAVE A DECIMAL POINT.
C
      IF ( NTYPFL .EQ. 1 ) GO TO 20
      NCOUNT = TSTOP
      NPRINT = TPRINT
      IF ( NPRINT .EQ. 0 ) NPRINT = NCOUNT
   20 TWR = TWALLF + 460.
      TSR = TSINKF + 460.
      TSSQ = TSR **2
      TSCU = TSR * TSSQ
      READ (5,102) RC,RHO,CP,TINJ,TINJO2
      TINF = TINJ
```

UCC UNIVERSITY COMPUTING COMPANY

```
      TINJ = TINJ + 460.                                                FILM
      WTEMPR = TINJ
      TINOF = TINJ02
      TINJ02 = TINJ02 +460.
      DO 200 I = 1,NNODE
  200 READ (5,102) X(I),DI(I),DO(I)
      IF ( NFLMFL .EQ. 0 ) GO TO 202
      READ (5,101) NINJ
      DO 201 I = 1,NINJ
      READ (5,102) SLOT(I), HSLOT(I), WCOOL(I)
  201 CONTINUE
C
C READ REGEN INPUT AND MAKE SURE REGEN STATIONS MATCH NODES
C
  202 IF ( NRGNFL .EQ. 0 ) GO TO 206
      READ (5,109) HRWD,NPASS,XREGEN
      DO 203 I = 1,NNODE
      READ (5,102) XR,HPASS(I),WPASS(I),ETAEF(I)
      IF ( ABS( XR - X(I)) .GT. .0001) GO TO 204
      IF ( ABS ( XR - XREGEN) .LT. .0001) GO TO 205
  203 CONTINUE
  204 WRITE (6,150) I
      CALL EXIT
  205 NRGEN = I
      ILES1 = I -1
C
C READ LINER INPUT AND MAKE SURE LINER STATIONS MATCH NODES
C
  206 IF ( NLFL .EQ. 0 ) GO TO 216
      READ (5,102) EMLDOT,EPSL,XLMAX
      DO 207 I = 1,NNODE
      READ (5,102) XL   ,DLI(I),DLO(I)
      IF ( ABS ( X(I) - XL   ) .GT. .0001 ) GO TO 208
      IF ( ABS ( XL    - XLMAX ) .LT. .0001 ) GO TO 209
  207 CONTINUE
  208 WRITE (6,137) I
      CALL EXIT                                                         LIN
C
C THIS GUARANTEES A FILM INJECTION STATION AT THE END OF THE LINER
C
  209 NLNODE = I
      IF ( NLNODE .NE. 1 ) GO TO 210
      WRITE (6,108)
      CALL EXIT
  210 IF ( NFLMFL .EQ. 1 ) GO TO 217
      NINJ = 1
      SLOT(1) = XLMAX                                                   LIN*
      HSLOT(1) = .5*( DI(NLNODE) - DLO(NLNODE))                         LIN*
      WCOOL(1) = EMLDOT                                                 LIN*
      NFLMFL = 1                                                        LIN*
      GO TC 216                                                         LIN*
  217 DO 218 I=1, NINJ                                                  LIN*
      IF (ABS( SLCT(I) - XLMAX) .LT. .0001) GO TO 219                   LIN*
      IF ( SLCT(I) .GT. XLMAX) GO TO 221                                LIN*
  218 CONTINUE                                                          LIN*
  219 SLOT(I) = XLMAX                                                   LIN*
      HSLOT(I) = .5* ( DI(NLNODE) - DLO(NLNODE))                        LIN*
      WCOOL(I) = EMLDOT                                                 LIN*
      GO TC 216                                                         LIN*
  221 NINJ = NINJ + 1                                                   LIN*
```

```
      NPLS1 = NINJ + 1                                                  LIN*
      NN = NINJ - 1
      DO 222 J = 1, NN                                                  LIN*
      K = NPLS1 - J                                                     LIN*
      SLOT(K) = SLOT(K-1)                                               LIN*
      HSLOT(K) = HSLOT(K-1)                                             LIN*
      WCOOL(K) = WCOOL(K-1)                                             LIN*
  222 CONTINUE                                                          LIN*
      SLOT(I) = XLMAX                                                   LIN*
      HSLOT(I) = .5* ( DI(NLNODE) - DLO(NLNODE))                        LIN*
      WCOOL(I) = EMLDOT                                                 LIN*
  216 NLES1 = NNODE -1                                                  GEOM 100
C
C MATCH FILM COOLING STATIONS WITH NODES FOR OUTPUT PURPOSES
C
      IF ( NINJ .EQ. 0 ) GO TO 18
      DO 15 J = 1 , NINJ
      DO 16 I = 1 , NNODE
      IF ( ABS ( X(I) - SLOT(J)) .LT. .0001) GO TO 17
   16 CONTINUE
   17 HSLT(I) = HSLOT(J)
      WCOL(I) = WCOOL(J)
   15 CONTINUE
C
C READ INJECTOR DATA IF ANY
C
      IF ( INJFL .EQ. 0 ) GO TO 18
      READ (5,102) AINJS,EMINJ,ARINJH,HGINJH,ARINJO,HGINJO
      READ (5,102) RESINJ,QFINJ,WTINJ,CPINJ
C
C WRITE THE INPUT
C
   18 IF (NTYPFL .EQ. 0) WRITE (6,1000) HED,NNODE,NTYPFL,NFLMFL,NRGNFL,
     1NLFL,INJFL,NCOUNT,NPRINT
      IF (NTYPFL .EQ. 1) WRITE (6,1001) HED,NNODE,NTYPFL,NFLMFL,NRGNFL,
     1NLFL,INJFL,TSTOP,TPRINT
      WRITE (6,1002) TWALLF,TSINKF,CAPPA,EPS,PO,OF,RC,RHO,CP,TINF,TINOF
      IF (INJFL .EQ. 1) WRITE (6,1012) AINJS,EMINJ,ARINJH,HGINJH,ARINJO,
     1HGINJO,RESINJ,QFINJ,WTINJ,CPINJ
      WRITE (6,1009) NPASS,HRWD,EPSL
      WRITE (6,1008)
      DO 60 I = 1,NNODE,5
      WRITE (6,1005)
      LINE = MINO(I+4,NNODE)
      DO 70 J = 1,LINE
   70 WRITE (6,1007) J,X(J),DI(J),DO(J),HSLT (J),WCOL (J),HPASS(J),
     1WPASS(J),ETAEF(J),DLI(J),DLO(J)
   60 CONTINUE
      IF ( NTYPFL .EQ. 0 .OR.  TPRINT .GT. 0.) GO TO 61
      WRITE (6,110)
      GO TC 10
   61 CONTINUE
C
C COMBUSTOR GEOMETRIC CALCULATIONS                                      GEOM  00
C                                                                       GEOM  80
C CALCULATE DISTANCE BETEEN NODES                                       GEOM  90
      DO 220 I=1,NLES1                                                  GEOM 110
  220 EL(I)= X(I+1) - X(I)                                              GEOM 120
C                                                                       GEOM 130
C SET UP X,Y COORDINATES FOR INNER AND OUTER WALLS                      GEOM 140
```

```
C CHANGE DIAMETER TO RADIUS                                              GEOM 150
C                                                                        GEOM 160
      DO 230 I=1,NNODE                                                   GEOM 170
      RI(I) = .5* DI(I)                                                  GEOM 180
  230 RO(I) = .5 * DO(I)                                                 GEOM 190
      XX(1) = X(1)                                                       GEOM 200
      RRI(1) = RI(1)                                                     GEOM 210
      RRO(1) = RO(1)                                                     GEOM 220
      DO 240 I=2,NLES1                                                   GEOM 230
      J = 2*I - 2                                                        GEOM 240
      XX(J) = X(I)                                                       GEOM 250
      RRI(J) =RI(I)                                                      GEOM 260
  240 RRO(J) =RO(I)                                                      GEOM 270
      NLES2 = NNODE - 2                                                  GEOM 280
      DO 250 I=2,NLES2                                                   GEOM 290
      J = 2*I - 1                                                        GEOM 300
      XX(J) = .5 * ( X(I)+X(I+1))                                        GEOM 310
      RRI(J) = .5 * ( RI(I)+RI(I+1))                                     GEOM 320
  250 RRO(J) = .5 * (RO(I) +RO(I+1))                                     GEOM 330
      N2LES3 = 2*NNODE - 3                                               GEOM 340
      J = N2LES3                                                         GEOM 350
      XX(J) = X(NNODE)                                                   GEOM 360
      RRI(J) = RI(NNODE)                                                 GEOM 370
      RRO(J) = RO(NNODE)                                                 GEOM 380
C                                                                        GEOM 390
      DO 260 K=1,NLES2                                                   GEOM 400
      J = 2*K                                                            GEOM 410
C                                                                        GEOM 420
C CALCULATE ELEMENT INNER SURFACE AREAS AS SURFACE OF CONE FRUSTRUMS     GEOM 430
C                                                                        GEOM 440
      ASI(K) =3.141593*((RRI(J-1)+RRI(J))* SQRT((RRI(J-1)-RRI(J))**2+    GEOM 450
     1(XX(J)-XX(J-1))**2)+(RRI(J)+RRI(J+1))*SQRT((RRI(J)-RRI(J+1))**2+   GEOM 460
     2(XX(J+1)-XX(J))**2))                                               GEOM 470
C                                                                        GEOM 480
C CALCULATE ELEMENT OUTER SURFACE AREAS AS SURFACE OF CONE FRUSTRUMS     GEOM 490
C                                                                        GEOM 500
      ASO(K) =3.141593*((RRO(J-1)+RRO(J))* SQRT((RRO(J-1)-RRO(J))**2+    GEOM 510
     1(XX(J)-XX(J-1))**2)+(RRO(J)+RRO(J+1))*SQRT((RRO(J)-RRO(J+1))**2+   GEOM 520
     2(XX(J+1)-XX(J))**2))                                               GEOM 530
C                                                                        GEOM 540
C CALCULATE ELEMENT VOLUMES AS DIFFERENCE BETWEEN TWO CONE FRUSTRUMS     GEOM 550
C                                                                        GEOM 560
  260 V(K) = 1.0472 * ((XX(J  ) - XX(J-1)) *
     1((  RRO(J-1) **2 + RRO(J-1) * RRO(J  ) + RRO(J  ) **2 ) -
     2 (  RRI(J-1) **2 + RRI(J-1) * RRI(J  ) + RRI(J  ) **2 ))+
     3                 (XX(J+1) - XX(J  )) *
     4((  RRO(J  ) **2 + RRO(J  ) * RRO(J+1) + RRO(J+1) **2 ) -
     5 (  RRI(J  ) **2 + RRI(J  ) * RRI(J+1) + RRI(J+1) **2 )))
C                                                                        GEOM 610
C CALCULATE AVERAGE CROSS SECTION AREAS BETWEEN NODES AS VOLUMES         GEOM 620
C DIVIDED BY LENGTH                                                      GEOM 630
      DO 270 I=1,NLES1                                                   GEOM 640
  270 AXAVE(I) = 1.0472 * (RO(I)**2+RO(I)*RO(I+1)+RO(I+1)**2-RI(I)**2    GEOM 650
     1-RI(I)*RI(I+1)-RI(I+1)**2)                                         GEOM 660
C
C CALCULATE ACCUMULATED INNER SURFACE AREAS BETWEEN NODES
C
      ACUM(1) =0.
      DO 275 I = 2,NNODE
  275 ACUM(I) = ACUM(I-1) + 3.141593 * ( RI(I-1) + RI(I)) *
```

```
     1SQRT (EL(I-1) **2 + ( RI(I-1) - RI(I)) **2)
C                                                                       LIN
C SET UP X,Y COORDINATES FOR LINER INNER AND OUTER SURFACES             LIN
C CHANGE DIAMETER TO RADIUS                                             LIN
C                                                                       LIN
      IF ( NLFL .EQ. 0 ) GO TO 274                                      LIN
      NLLES1 = NLNODE - 1                                               LIN
      DO 710 I = 1,NLNODE                                               LIN
      RLI(I) = .5 * DLI(I)                                              LIN
  710 RLO(I) = .5 * DLO(I)                                              LIN
      RRLI(1) = RLI(1)                                                  LIN
      RRLO(1) = RLO(1)                                                  LIN
      DO 720 I = 2,NLLES1                                               LIN
      J = 2 * I - 2                                                     LIN
      RRLI(J) = RLI(I)                                                  LIN
  720 RRLO(J) = RLO(I)                                                  LIN
      NLLES2 = NLNODE - 2                                               LIN
      DO 730 I = 2,NLLES1                                               LIN
      J = 2 * I - 1                                                     LIN
      RRLI(J) = .5 * ( RLI(I) + RLI(I+1))                               LIN
  730 RRLO(J) = .5 * ( RLO(I) + RLO(I+1))                               LIN
      NL2LS2 = 2 * NLNODE - 2                                           LIN
      RRLI(NL2LS2) = RLI(NLNODE)                                        LIN
      RRLO(NL2LS2) = RLO(NLNODE)                                        LIN
C                                                                       LIN
      DO 740 K = 1,NLLES2                                               LIN
      J = 2 * K                                                         LIN
C                                                                       LIN
C LINER ELEMENT INNER SURFACE AREA                                      LIN
C                                                                       LIN
      ASLI(K) = 3.141593 * (( RRLI(J-1) + RRLI(J)) * SQRT ((RRLI(J-1) - LIN
     1RRLI(J))**2 + (XX(J) - XX(J-1))**2) + (RRLI(J) + RRLI(J+1)) *     LIN
     2SQRT ((RRLI(J) - RRLI(J+1))**2 + (XX(J+1) - XX(J))**2))           LIN
C                                                                       LIN
C LINER ELEMENT OUTER SURFACE AREAS                                     LIN
C                                                                       LIN
      ASLO(K) = 3.141593 * (( RRLO(J-1) + RRLO(J)) * SQRT ((RRLO(J-1) - LIN
     1RRLO(J))**2 + (XX(J) - XX(J-1))**2) + (RRLO(J) + RRLO(J+1)) *     LIN
     2SQRT ((RRLO(J) - RRLO(J+1))**2 + (XX(J+1) - XX(J))**2))           LIN
  740 CONTINUE                                                          LIN
      NL2LS3 = 2 * NLNODE -3                                            LIN
      ASLI(NLLES1) = 3.141593 * ( RRLI(NL2LS3) + RRLI(NL2LS2)) * SQRT   LIN
     1((RRLI(NL2LS3) - RRLI(NL2LS2)) **2 + (XX(NL2LS2) - XX(NL2LS3))**2)LIN
      ASLO(NLLES1) = 3.141593 * ( RRLO(NL2LS3) + RRLO(NL2LS2)) * SQRT   LIN
     1((RRLO(NL2LS3) - RRLO(NL2LS2)) **2 + (XX(NL2LS2) - XX(NL2LS3))**2)LIN
C                                                                       LIN
C ACCUMULATED LINER INNER SURFACE AREAS BETWEEN NODES                   LIN
C                                                                       LIN
      ACUML(1) = 0.                                                     LIN
      DO 750 I = 2, NLNODE                                              LIN
  750 ACUML(I) = ACUML(I-1) + 3.141593 * ( RLI(I-1) + RLI(I)) *         LIN
     1SQRT (EL(I-1) **2 + ( RLI(I-1) - RLI(I))**2)                      LIN
      DO 755 I = NLNODE, NLES1                                          LIN*
  755 ACUML(I+1) = ACUML(I) + 3.141593 * (RI(I) + RI(I+1)) *            LIN*
     1SQRT (EL(I) **2 + ( RI(I) - RI(I+1))**2)                          LIN*
C FIND THE THROAT
C
  274 NPLUS1 = NNODE + 1
      DO 320 I=1,NLES1
      J = NPLUS1 - I
      IF ( DI(J) - DI(J-1)) 325,320,320
  320 CONTINUE
  325 DSTAR = CI (J)
      ASTAR = .785398 * DSTAR **2                                       HG    70
      NTHRT = J
      IF ( NLFL .EQ. 0 ) GO TO 276
C                                                                       LIN
C INITIALIZE LINER TEMPERATURES                                         LIN
C                                                                       LIN
      DO 760 I = 1, NLNODE                                              LIN
      TL(I) = TWR                                                       LIN
  760 THL(I) = TINJ                                                     LIN
C COMBUSTOR HEAT TRANSFER CHARACTERISTICS
C
C
C CONDUCTION BETWEEN NODES , ADMITTANCE
C
  276 DO 280 I=1,NLES1
  280 ADM(I) = CAPPA * AXAVE(I) /EL(I)
C
C CAPACITIES
C
      CPRHO = CP * RHO
      DO 290 I=1,NLES2
  290 CAP(I) = CPRHO * V(I)
      EPS1 = EPS * 3.304E-15
      DO 300 I=1,NNODE
  300 TW(I) = TWR
      NFILM =1                                                          FILM
      NITR = C
      IF (INJFL .EQ. 0) GO TO 306
C
C INJECTOR CALCULATION
C
  305 IF ( INJFL .EQ. 0) GO TO 311
      IF ( NITR .EQ. 0) TI = TWR
      RADI = 3.304E-15 * AINJS * EMINJ * (TSCU + TI * (TSSQ + TI * (TSR
     1+ TI)))
      SUMI = RADI + ARINJH * HGINJH + ARINJO * HGINJO + 1. /RESINJ
      CI = WTINJ * CPINJ
      DTAUI = CI / SUMI
      TW(1) = (TI / RESINJ + ADM(1) * TW(2)) / ( ADM(1) + 1. / RESINJ)
      IF ( NTYPFL .EQ. 0)
C
C INJECTOR TEMPERATURE, IF STEADY STATE
C
     1TI = (RADI * TSR + ARINJH * HGINJH * HTEMPR + ARINJO * HGINJO *
     2TINJC2 + QFINJ * .785398 * DI(1)**2 + TW(1) / RESINJ) /SUMI
      TIF = TI - 460.
      IF (NITR  .EQ. 0) GO TO 306
C
C WRITE THE OUTPUT FOR TRANSIENT AT INTERVALS DETERMINED BY TPRINT
C
  311 IF ( NTYPFL .EQ. 0 ) GO TO 308
      IF (( TAU - TAU1 ) .LT. TPRINT ) GO TO 306
      WRITE (6,1010) HED,TAU,OF,TZEROF,WDOT
      IF (INJFL .EQ. 1) WRITE (6,1013) TIF
      WRITE (6,1011)
      DO 35 I = 1,NNODE,5
      WRITE (6,1005)
      LINE = MIN0 (I+4,NNODE)
      DO 36 J = 1,LINE
      TWF = TW(J) - 460.
   36 WRITE (6,1006) J,X(J),TWF,TFF(J)
      TAU1 = TAU - AMOD( TAU,TPRINT)
   35 CONTINUE
      IF ( TAU .GT. TSTOP ) GO TO 10
      GO TO 306
C
C WRITE THE OUTPUT FOR STEADY STATE AT INTERVALS DETERMINED BY NPRINT,
C BUT ALWAYS PRINT LAST 4 ITERATIONS
C
  308 NPR = NCOUNT - NITR
      IF ( NPR .LE. 3) GO TO 307
      IF ( MOD ( NITR, NPRINT) .NE. 0) GO TO 306
  307 WRITE (6,1003) HED,NITR,OF,TZEROF,WDOT
      IF (INJFL .EQ. 1) WRITE (6,1013) TIF
      WRITE (6,1004)
      DO 40 I = 1,NNODE,5
      WRITE (6,1005)
      LINE = MIN0(I+4,NNODE)
      DO 50 J = 1,LINE
      TWF = TW(J) - 460.
      TBF = TB(J) - 460.
      TLF = TL(J) -460.
      THLF = THL(J) -460.
   50 WRITE (6,1006) J,X(J),TWF,TFF(J),TBF,TLF,THLF
   40 CONTINUE
      IF ( NTYPFL .EQ. 1 ) GO TO 46
      IF ( NITR .GE. NCOUNT ) GO TO 10
      GO TO 306
   46 IF ( TAU .GE. TSTOP ) GO TO 10
C
C RADIANT ADMITTANCES
C
  306 DO 310 I=1,NLES2
  310 RADADM(I) = EPS1 * (TSCU + TW(I+1) * (TSSQ + TW(I+1) *
     1(TSR + TW(I+1))))* ASO(I)
      IF ( NLES2 .LT. NTHRT ) GO TO 6360
      ELL = X(NNODE) - X(NTHRT)
      LOD = ELL / DSTAR
      FLOD = 444.389 + LOD * ((-814.6) + LOD*( 444.29777 - 74.047 *LOD))
      DO 360 I = NTHRT , NLES2
      XOL =(X(I+1) - X(NTHRT)) / ELL
C
C VIEW FACTOR FOR THE TAILPIPE ( 15 DEGREE CONICAL ANGLE)
C
      VIEW = FLOD + XOL * ((-.04212) * LOD + .28176 + XOL*((-.5135) +
     1 .86475 * XOL))
      RADTP = EPS1 * (TSCU + TW(I+1) * (TSSQ + TW(I+1) * (TSR +
     1TW(I+1)))) * ASI(I) * VIEW
      RADADM(I) = RADADM(I) + RADTP
  360 CONTINUE
 6360 CONTINUE
      IF (NITR .GT. 0) GO TO 326
C                                                                       HG    00
C EVALUATE SOME PARAMETERS FROM DATA WHICH IS STORED AS FUNCTIONS
C                                                                       HG    20
C CP OF THE COMBUSTION GAS
C
```

```
      IF ( OF .GT. 8. ) GO TO 390
      CPG = 3.322 + OF*((- 1.9447 ) + OF*( .6478 + OF*((-.09804) + OF*
     1.005608)))
      GO TO 391
  390 CPG = 39.5584 + OF*((-11.5075) + OF*(1.3548 + OF*((-.07824)+ OF*(
     1.0021966 + CF*(-2.3872E-5)))))
  391 CONTINUE
C
C MOLECULAR WT OF THE COMBUSTION GAS
C
      WTMOL = FMOLWT(OF)
C
C GAMMA
C
      GAM    = SPHEAT(OF)
C
C COMBUSTION CHAMBER TEMPERATURE
C
      TZERO = TGR(PO,OF)
C
C CORRECT TZERO FOR H2 AND O2 INJECTION TEMPERATURES
C
      C1OF =        ((-.00506)     + OF * (  .1022       + OF*((-.0382)
     1      + CF * (  .00614       + OF * ( -.000507     + OF*(  2.28E-5
     2      + CF * ((-5.272E-7)    + OF * ( 4.913E-9     ))))))))
      C2OF =        (1.002         + OF * (-.0627        + OF*((-.0799)
     1      + CF * ( .0199         + OF * ((-.00199)     + OF*( 9.953E-5
     2      + CF * ((-2.463E-6)    + OF * ( 2.398E-8     ))))))))
      TZERO = TZERO    +C1OF * ( TINJO2 - 535.) + C2OF * (TINJ - 535.)
      TZEROF = TZERO -460.
C
C PRANDTL NUMBER
C
      PRAND = (4. * GAM ) / (9. * GAM - 5.)
C
C VISCOSITY
C
      VISC = 46.6E-10 * SQRT (WTMOL) * (TZERO) ** .6
C FIND WEIGHT FLOW
C
      WDOT = PC * ASTAR * SQRT ((( WTMOL * GAM )/( 48.0634 * TZERO ))*  HG   140
     1( 2. / ( GAM + 1. )) ** (( GAM + 1. )/( GAM - 1.)))              HG   150
C                                                                      HG   160
C CALCULATE HG DIVIDED BY  (ASTAR/A)*SIGMA                             HG   170
C                                                                      HG   180
      HOVAS = (.026 * (VISC/DSTAR)**.2 * CPG * (WDOT/ASTAR)**.8 *      HG   190
     1(DSTAR/RC)**.1) / PRAND**.6                                      HG   200
C                                                                      HG   220
C CALCULATE HG(I) OVER SIGMA                                           HG   230
C                                                                      HG   240
      DO 321 I=1,NNODE
      ASTOA(I) = ( DSTAR / DI(I  )) **2
  321 HGOVSI(I) = HOVAS * (ASTOA(I)) **.9
C
      DO 322 I = 1 , NNODE
  322 TF(I) = TZERO
C
C IS THERE FILM COOLING ?
C
  323 IF(NFLMFL .EQ. 1) GO TO 490                                      FILM  50
  326 DO 330 I=1,NLES2
      SIGMA(I) = ( 2.* TF(I+1) / ( TF(I+1) + TW(I+1))) **.8
      HG(I) = HGOVSI(I+1) * SIGMA(I) * ASI(I)
  330 SUMADM(I) = HG(I) + RADADM(I) + ADM(I) + ADM(I+1)
C                                                                       RGEN
C REGEN ?                                                               RGEN
C                                                                       RGEN
      IF (NRGNFL .EQ. 1) GO TO 600                                      RGEN
C
C LINER?
C
      IF ( NLFL .EQ. 1) GO TO 700                                       LIN
C
C STEADY STATE OR TRANSIENT ?
C
      IF ( NTYPFL ) 327,327,335
C
C WALL TEMP FOR STEADY STATE, MAY HAVE MULTI-SLOT FILM COOLING AND
C INJECTOR, BUT NO REGEN OR LINER
C
  327 DO 331 I=1,NLES2
      TW(I+1) = (HG(I) * TF(I+1)+RADADM(I) * TSR + ADM(I) * TW(I) +
     1ADM(I+1) * TW(I+2)) / SUMADM(I)
  331 CONTINUE
      TW(1) = TW(2)
      TW(NNODE) = TW(NLES1)
      NITR = NITR +1
      GO TO 305
C
C TRANSIENT BRANCH, MAY HAVE MULTI- SLOT FILM COOLING AND INJECTOR, BUT
C NO REGEN OR LINER
C
  335 IF (NITR .GT. 0) GO TO 339
C
C MAXIMUM DELTA TIME FOR STABILITY
C
      TAU = 0.
  339 DTAU = CAP(1) / SUMADM(1)
      DO 340 I=2,NLES2
  340 DTAU = AMIN1 ( DTAU, ( CAP(I) / SUMADM(I)))
      IF (INJFL .EQ. 0) GO TO 341
      DTAU = AMIN1 (DTAU,DTAUI)
      TIOLD = TI
C
C INJECTOR TEMPERATURE IF TRANSIENT
C
      TI = (DTAU / CI) * ( RADI * TSR + ARINJH * HGINJH * HTEMPR +ARINJO
     1* HGINJC * TINJO2 + QFINJ * .785398 * DI(1)**2 + TW(1)/RESINJ) +
     2( 1. - (SUMI * DTAU)/ CI ) * TIOLD
  341 TAU = TAU + DTAU
C
C CALCULATE WALL TEMP AT NEXT TIME STEP
C
      DO 345 I=1,NNODE
  345 TOLD(I) = TW(I)
      DO 350 I=1,NLES2
  350 TW(I+1) =  ( HG(I) *TF(I+1)+ RADADM(I) * TSR + ADM(I) * TOLD(I)+
     1ADM(I+1) * TW(I+2) - SUMADM(I) * TOLD(I+1)) * (DTAU / CAP(I)) +
     2TOLD(I+1)
      TW(1) = TW(2)
      TW(NNODE) = TW(NLES1)
      NITR = NITR +1
      GO TC 3C5
C
C FILM COOLING CALCULATIONS
C
  490 IF (NRGNFL .EQ. 1 .AND. NITR .EQ. 0) GO TO 625
      IF (NRGNFL .NE. 1.AND. NLFL .NE. 1 ) GO TO 495                    RG****
      DO 491 I = 1, NNODE                                                RG***
  491 TF(I) = TZERO                                                      RG***
  495 DO 500 J = 1, NINJ                                                 FILM  70
C
C MATCH FILM STATIONS TO NODES. IT'S A RULE OF THIS GAME THAT INJECTION
C STATIONS MUST COINCIDE WITH A NODE.
C
      DO 510 I = 1,NNODE                                                 FILM  80
      IF ( ABS  (X(I) - SLOT(J)) .LT. .0001) GO TO 520                   FILM  90
  510 CONTINUE
      WRITE (6,104) (J)                                                  FILM
      CALL EXIT                                                          FILM
C
C COOLANT VELCCITY
C
  520 IF ( NRGNFL .EQ. 1 ) TINJ = TB(I)                                  RG**
      IF (NLFL.EQ.1.AND.SLOT(J).EQ.XLMAX) TINJ = THL(NLNODE)
      IF ( I .NE. 1) TFBAR1 = .5 * (TF(I-1) + TF(I))
      VG =(1969. * WDOT * TZERO)/(PO * DI(I) ** 2)
      VC    = (244. * TINJ * WCOOL(J)) / ( PO * HSLOT(J) * DI(I))       RG**
      VCSON = 185.88 * SQRT ( TINJ )
      IF (VC .GT. VCSON) WRITE (6,105) J
      IF ( (VG/VC    ) .GT. 1. ) GO TO 525                               FILM 110
      VGVCF     = ( VC    / VG)** ( 1.5 * ( VC/VG -1.))                  FILM 120
      GO TC 526                                                          FILM 130
  525 VGVCF     = 1.+ .4 * ATAN ( VG/VC - 1.)                            FILM 140
  526 SIGMA(I) = ( 2. * TF(I) / (TF(I) + TINJ )) **.8                    FILM 150
      HG(I) = HGOVSI(I) * SIGMA(I)                                       FILM 160
      TFF(I) = TF(I) - 460.
      TF(I) = TINJ                                                       FILM 270
      HGABAR = 0.                                                        FILM 170
      IF ( I .EQ. 1 ) TFF(I) = TINJ - 460.
      DO 530 K = I , NLES1                                               FILM 180
      TFOLD = TF(K)
      SIGMA(K+1) = ((2. * TF(K+1)) / ( TF(K+1) + TFOLD )) ** .8
      HG(K+1) = HGOVSI(K+1) * SIGMA(K+1)                                 FILM 200
      IF ( NLFL .EQ. 0 ) GO TO 527                                       LIN*
      HGABAR = HGABAR + (ACUML(K+1) - ACUML(K)) * (.5 * (HG(K+1) +       LIN*
     1HG(K)))                                                            LIN*
      GO TC 528                                                          LIN*
  527 HGABAR = HGABAR + ( ACUM(K+1) - ACUM(K)) * (.5 * (HG(K+1) +        FILM 210
     1HG(K)))                                                            FILM 220
  528 ETA      = EXP ( -( HGABAR/(3.4 * WCOOL(J)))* ((HSLOT(J) * VG *    FILM 230
     15. ) **.125) * VGVCF     )                                         FILM 240
C
C FILM TEMPERATURES
C
      TF(K+1) = .5 * (TF(K+1) + TFOLD ) - ETA * ( .5 * (TF(K+1) + TFOLD)
     1 - TINJ)
      TFF(K+1) = TF(K+1) - 460.
  530 CONTINUE                                                           FILM
      TFBAR2 = .5 * ( TF(I) + TF(I+1))
      IF ( I .NE. 1 ) TF(1) = .5 * ( TFBAR1 + TFBAR2 )
  500 CONTINUE                                                          FILM 300
      NFILM = NFILM +1                                                  FILM
      GO TO 326                                                         FILM 290
C                                                                       RGEN
C REGEN CALCULATIONS, STEADY STATE ONLY, MAY HAVE MULTI-SLOT FILM
C COOLING AND INJECTOR, BUT NO LINER
C                                                                       RGEN
  600 CONTINUE
  620 IF (NITR .GT. 0) GO TO 6636
      ELBE =  X(NRGEN) - X(1)
C                                                                       RGEN
C INITIALIZE REGEN TEMPERATURES                                         RGEN
C                                                                       RGEN
      IF ( NFILM .GT. 1 ) GO TO 635                                     RG**
  625 DO 630 K=1,NRGEN                                                  RG**
  630 TB(K) = HTEMPR                                                    RGEN
      IF ( NFLMFL .EQ. 1 ) GO TO 495                                    RG**
C                                                                       RGEN
C FIND TEMPERATURES FOR REGEN SECTION                                   RGEN
C                                                                       RGEN
  635 CONTINUE
      DO 636 IAD = 1,ILES1
      APASS = NPASS * HPASS(IAD+1) * WPASS(IAD+1)
      ADM(IAD) = ADM(IAD) * ( 1. - APASS/AXAVE(IAD))                    RGEN
  636 CONTINUE                                                          RGEN
 6636 CONTINUE
      DO 640 L=1,ILES1                                                  RGEN
      ELBOVD =  (.5 * (( ELBE / WPASS(L +1)) + ( ELBE / HPASS(L +1))))
     1** (-.19)
      REMUB = ((2. * HRWD ) / ( HPASS(L +1)+ WPASS(L +1))) ** .8
      HCONST = .041618 * REMUB *ELBOVD                                  RGEN
      SURFCL = 2. * NPASS * ( HPASS(L +1) + WPASS(L +1))
C                                                                       RGEN
C                                                                       RGEN
C EVALUATE CONDUCTIVITY, VISCOSITY, AND CP OF HYDROGEN
C                                                                       RGEN
      TBF = TB(L  ) - 460.                                              RGEN
      HCAPPA = 2.1296E-6 + TBF     * ( 3.8194E-9  + TBF     *           RGEN
     1(( -1.868E-12) + TBF     * ( .9118E-15 - 1.5602E-19 * TBF    )))  RGEN
      HVISC = 4.57E-7 + TBF      * ( 6.74E-10 + TBF      * ((-2.83E-13) + RGEN
     1TBF      * ( 1.5E-16 - 3.14E-20 * TBF    )))                      RGEN
      HGMCW = ( HCONST * HCAPPA * ( TW(L+1) / TB(L+1)) **(-.55)) /      RGEN****
     1(HVISC) ** .8                                                     RGEN****
      ASURF = SURFOL * EL(L+1)                                          RGEN****
      SUMAD  = HG(L  ) + RADADM(L  ) + ADM(L  ) + ADM(L+1) +            RGEN
     1HGMCW * ETAEF(L+1) * ASURF
C
C FIND TWALL UP TO REGEN INJECTION POINT
C
      TW(L+1) = (HG(L  ) * TF(L+1) + RADADM(L  ) * TSR + ADM(L  ) *     RGEN
     1TW(L  ) + ADM(L+1) * TW(L+2) + HGMCW * ETAEF(L+1) *ASURF * TB(L+1)
     1) / SUMAD
  640 CONTINUE                                                          RGEN
      DO 645 IB = 1, ILES1                                              RGEN
      JB = NRGEN - IB                                                   RGEN
      ELBOVD =  (.5 * (( ELBE / WPASS(JB+1)) + ( ELBE / HPASS(JB+1))))
     1** (-.15)
      REMUB = ((2. * HRWD ) / ( HPASS(JB+1)+ WPASS(JB+1))) ** .8
      SURFOL = 2. * NPASS * ( HPASS(JB+1) + WPASS(JB+1))
      TBF = .5 * (TB(JB+1)  + TB(JB)) -460.                             RGEN
      HCAPPA = 2.1276E-6 + TBF      * ( 3.8194E-9  + TBF      *         RGEN
     1(( -1.868E-12) + TBF      * ( .9116E-15 - 1.5602E-19 * TBF    ))). RGEN
      HVISC = 4.57E-7 + TBF      * ( 6.74E-10 + TBF      * ((-2.83E-13) + RGEN
     1TBF      * ( 1.5E-16 - 3.14E-20 * TBF     )))                     RGEN
      CPH = 3.38 + TBF * (.00104 + TBF * ((-3.165E-6) + TBF * (3.62E-9  RGEN
     1+ TBF * ((-1.6E-12) + 2.39E-16 * TBF ))))                         RGEN
      ADCLNT = CPH * HRWD                                               RGEN
      HGMCW = ( HCONST * HCAPPA * ( TW(JB ) / TB(JB )) **(-.55)) /      RGEN
     1(HVISC) ** .8                                                     RGEN
      ASURF = SURFOL * EL(JB )                                          RGEN
C
C FIND HYDROGEN TEMPERATURE IN REGEN PASSAGE
C
      TB(JB ) =(HGMCW * ETAEF(JB)*ASURF*TW(JB ) + ADCLNT * TB(JB+1))/   RGEN
     1( HGMCW * ETAEF(JB) * ASURF + ADCLNT )
  645 CONTINUE                                                          RGEN
C
C FIND TWALL AFT OF REGEN INJECTION POINT
C
      DO 650 M=NRGEN,NLES2
      TW(M+1) = (HG(M  ) * TF(M+1) + RADADM(M  ) * TSR + ADM(M  ) *     RGEN
     1TW(M  ) + ADM(M+1) * TW(M+2)) / SUMADM(M  )                       RGEN
  650 CONTINUE                                                          RGEN
      TW(1) = TW(2)                                                     RGEN
      TW(NNODE) = TW(NLES1)                                             RGEN
      NITR = NITR + 1                                                   RGEN
      GO TO 305
C                                                                       LIN
C LINER CALCULATIONS, STEADY STATE ONLY, MAY HAVE MULTI-SLOT FILM
C COOLING, WILL HAVE A FILM AT END OF LINER WHETHER INPUT OR NOT, MAY
C HAVE INJECTOR, BUT NO REGEN
C
C RADIANT ADMITTANCES BETWEEN LINER AND WALL                            LIN
C                                                                       LIN
  700 ELBE = X(NLNODE) - X(1)                                           LIN
      DO 770 I = 2, NLNODE                                              LIN
      EPSBAR = 1. / ((1./EPSL) + (( DLO(I) / DI(I))**2) * ((1./EPS) - 1.LIN
     1))                                                                LIN
      RADL = 3.304E-15 * EPSBAR * ( TW(I)**3 + TL(I) * (TW(I)**2) +     LIN
     1(TL(I)**2) * TW(I) + (TL(I)**3)) * ASLO(I-1)                      LIN
C                                                                       LIN
C HEAT TRANSFER COEFFICIENTS FOR THE LINER                              LIN
C                                                                       LIN
      SIGMAL =  ((2.* TF(I)) / ( TF (I) + TL(I)))**.8                   LIN
      HGLA  =  ((DSTAR / DLI(I))**1.8) * SIGMAL * HOVAS * ARLI(I-1)     LIN
C                                                                       LIN
C MCCARTHY-WOLF H2 CONVECTION                                           LIN
C                                                                       LIN
      TBF = THL(I) - 460.                                               LIN
      HCAPPA = 2.1296E-6 + TBF      * ( 3.8194E-9  + TBF      *         LIN
     1(( -1.868E-12) + TBF      * ( .9118E-15 - 1.5602E-19 * TBF    )))  LIN
      HVISC = 4.57E-7 + TBF      * ( 6.74E-10 + TBF      * ((-2.83E-13) + LIN
     1TBF      * ( 1.5E-16 - 3.14E-20 * TBF     )))                     LIN
      CPH = 3.38 + TBF * (.00104 + TBF * ((-3.165E-6) + TBF * (3.62E-9  LIN
     1+ TBF * ((-1.6E-12) + 2.39E-16 * TBF ))))                         LIN
      HGMCW = .058233*((HCAPPA) / (DI(I) - DLO(I))) * ((EMLDOT / ( HVISCLIN
     1* (DI(I) + DLO(I)))) **.8) * (( TW(I) / THL(I)) **(-.55))* ( ELBE LIN
     2 / ( DI(I) - DLO(I))) **(-.15)                                    LIN
      SUML = RADL + HGLA + HGMCW                                        LIN
      TL(I) = ( RADL * TW(I) + HGLA * TF(I) + HGMCW * THL(I)) / SUML     LIN
      HGMC = HGMCW * ( TL(I) / TW(I)) **(-.55)                           LIN
      SUMH = HGMCW + HGMC +CPH * EMLDOT                                  LIN
      THL(I) = ( HGMCW * TL(I) + CPH * EMLDOT * THL(I-1) + HGMC * TW(I))LIN
     1/ SUMH                                                             LIN
      SUMW = ADM(I) + ADM(I+1) + HGMCW + RADL + RADADM(I)                LIN
      TW(I) = ( ADM(I) * TW(I-1) + ADM(I+1) * TW(I+1) + HGMCW * THL(I)  LIN
     1+ RADL * TL(I) + RADADM(I) * TSR) / SUMW                           LIN
  770 CONTINUE                                                           LIN
C
C FIND TWALL AFT OF LINER
C
      DO 780 M=1,NLES2
      TW(M+1) = (HG(M  ) * TF(M+1) + RADADM(M  ) * TSR + ADM(M  ) *      LIN
     1TW(M  ) + ADM(M+1) * TW(M+2)) / SUMADM(M  )                        LIN
  780 CONTINUE
      TW(1) = TW(2)
      TW(NNODE) = TW(NLES1)
      NITR = NITR +1                                                     LIN
      GO TO 305
 1100 FORMAT (18A4)
  101 FORMAT (8I3)                                                       LIN
  102 FORMAT (6F12.0)                                                    GEOM 730
  104 FORMAT('0',6X,'A COINCIDENT NODE WAS NOT FOUND FOR FILM INJECTION FILM
     1POINT',I4)                                                         FILM
  105 FORMAT('0',6X,'COOLANT VELOCITY IS PROBABLY SUPERSONIC AT FILM INJ
     1ECTION POINT',I4)
  106 FORMAT('0',6X,'THE REGEN INJECTION STATION DOES NOT CORRESPOND TO RGEN
     1A NODE')                                                           RGEN
  108 FORMAT ('0',6X,'THE LINER IS OF ZERO LENGTH')
  109 FORMAT (F12.0,I12,F12.0)
  110 FORMAT ('0',6X,'TPRINT WAS INPUT AS ZERO.  THE PRINT INTERVAL IS U
     1NDEFINED.')
  137 FORMAT ('0',6X,'THE LINER AND WALL STATIONS ARE NOT COINCIDENT AT LIN
     1NODE',I4)                                                          LIN
  150 FORMAT ('0',6X,'THE REGEN AND WALL STATIONS DO NOT COINCIDE AT NOD
     1E',I4)
 1000 FORMAT ('1  --ROCKET HEAT TRANSFER MODEL--   ',18A4///6X,'NNODE
     1NTYPFL  NFLMFL  NRGNFL  NLFL    INJFL   NCOUNT  NPRINT'//2X8I8)
 1001 FORMAT ('0  --ROCKET HEAT TRANSFER MODEL--   ',18A4///6X,'NNODE
     1NTYPFL  NFLMFL  NRGNFL  NLFL    INJFL   TSTOP        TPRINT'//2X6I8
     2,2G12.5)
 1002 FORMAT ('0',7X,'T WALL    T SINK    CAPPA     EMISS     PO        O
     1/F        RC        RHO       CP       TINJH2   TINJC2'/8X,'DEG F     D
     2EG F                          PSIA                INCHES'22X'DEG F
     3DEG F'/1X,F12.0,F10.0,2F10.4,F8.1,F10.3,F9.2,F10.4,F8.4,F10.1,F9.1
     4)
 1003 FORMAT ('1  --ROCKET HEAT TRANSFER MODEL--   ',18A4///6X,'ITERATIO
     2N NO.   CORE O/F  CORE TEMP   WT. FLOW'/32X,'DEG F         LB/SEC'/
     37X,I12,F10.3,F10.1,F11.3)
 1004 FORMAT ('0',6X,'NODE    STATION   T WALL   T FILM   T BULK  T LINE
     1R     T H2'/43X,'REGEN'13X,'LINER')
 1005 FORMAT ('0')
 1006 FORMAT (' ',6X,I4,F11.2,5F9.1)
 1007 FORMAT (' ',6X,I4,F10.2,2F8.2,F14.3,F8.3,F13.3,F7.3,F8.3,F13.3,
     1F8.3)
 1008 FORMAT ('0',6X,'NODE    STATION     ID        OD',10X,'HEIGHT  WDOT',9
     1X,'HEIGHT  WIDTH   ETA F',10X,'ID        OD')
 1009 FORMAT ('0',28X,'WALL',13X,'FILM COOLING',11X,'REGEN 'I4,' PASSAGE
     1S',10X,'LINER'/69X,'WDOT = ',F8.3,12X,'EMISS =',F7.4)
```

```
1010 FORMAT ('1  --ROCKET HEAT TRANSFER MODEL--   ',18A4///6X,'     TIM
    2E        CORE O/F  CORE TEMP   WT. FLOW'/32X,'DEG F         LB/SEC'/
    37X,F12.3,F10.3,F10.1,F11.3)
1011 FORMAT ('0',6X,'NODE    STATION   T WALL   T FILM')
1012 FORMAT ('0',7X,'INJECTOR INPUT'/9X,'RADIATION      INJECTOR   CONVE
    1CTION  CONVECTION  CONVECTION  CONVECTION     SEAL     FACE HEAT
    2INJECTOR  INJECTOR'/8X,'SURFACE AREA  EMISSIVITY   AREA  H2   COEF
    3F  H2    AREA  O2   COEFF  O2   RESISTANCE    FLUX       WEIGHT
    4  CP'//8X,F11.3,F12.5,F12.3,F12.5,F12.3,F12.5,F12.0,F10.3,F11.4,F1
    50.3)
1013 FORMAT ('0',6X,'INJECTOR TEMPERATURE =',F8.1)
     END                                                               CEOM 750
```

```
 • ELT HCNC,1,720512, 51602     , 1

            SUBROUTINE HONC
      C     HYDROGEN/OXYGEN FINITE RATE CHEMISTRY WITH
      C     HO2,H2O2 QUENCHING REACTIONS-ORDER OF SPECIES
      C     SPECIES ARE 1-H 2-O 3-H2O 4-H2 5-O2 6-OH 7-HO2 8-H2O2 9-DILUENT
      C                                                                    00000120
      C     TIME IN SECONDS    T IN DEG K
      C                                                                    00000040
            COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
           1,IDA(200),TCHEM(200),WDT(9,200)
            COMMON/ZC/W(9),TITLE(12),CGP(7,4),XP(7),XK(7)
           1,GSCALE,TVX(7,4)
            COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
           1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
           2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
           3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
           4,NEW,MQ,MR,LN,NNT,JR,NSLCR
            COMMON/ZE/X,XMAX,PRES,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA
           1,HSTOR,LSTOR,RAY,RG,AK,AKA
            COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TWDUMB
            COMMON/FGG/TEMP,H,ALPHI(9),KPSI,A(8,8),AW(9),C(8),RHO,TIMEF,DT     HONC
            COMMON/FGY/B(8,8),E(8,8),N,NDUM,DEAD
            DIMENSION ALPHA(9),ALP(9),HXY(8),AWW(9),CAW(8)
      C                                                                    00000110
      C     SET INITIAL CONDITIONS                                         00000180
            KTEST=1                                                        00000350
            IDA(KPSI)=IDA(KPSI)+1
            IF(IDA(KPSI).GT.1) GO TO 222
            N=8
            NDUM=N
            NP=9
            NALL=9
            PLH=XP(6)
            PLT=PLH
            PUH=10.*PLH
            PUT=PUH
            TMAX=XP(5)
            FUDGE=.0
            TESTY=1.E-20
      C                                                                    00002460
      C     SAVE INITIAL CONDITIONS I.E. SPECIES AND TEMPERATURE.          00000380
        222 NUT=0
          2 CONTINUE
            T=TEMP                                                         00000230
            PRESS=PRES/RG
      C                                                                    00000300
         94 DUM=.0
            DO 30 J=1,NALL
         14 ALPHA(J)=ALPHI(J)                                              00000220
            AW(J)=ALPHA(J)/W(J)
         30 CONTINUE
            RHG=RHO
      C                                                                    00000320
      C     NOW GO AND COMPUTE.                                            00000330
      C     SET STEP SIZE AND NUMBER OF TIMES THE LOOP MUST BE DONE.       00000370
      C                                                                    00000390
            RW=RHO
            TW=T
         76 DUM=TIMEF
      IF(FIX(KPSI).LE.TMAX)GO TO 176
      FIX(KPSI)=TMAX
  176 IF(DUM-FIX(KPSI))90,90,31
   90 DT=DUM                                                    00000460
      ANN=1.
      GO TC 32                                                  00000480
   31 ANN=DUM/FIX(KPSI)
      DT=DUM/ANN
   32 IF(IDA(KPSI).LT.3)GO TO 55
      AH=ABS(RHO/HHLD(KPSI)-1.)
      AH=AH/ANN
      AT=ABS(T/THLD(KPSI)-1.)
      AT=AT/ANN
      IF(LR.EG.1) GO TO 61
      IF((KTEST+NUT).GT.1) GO TO 61
      IF(AH-PLH)60,61,61
   60 IF(AT-PLT)62,61,61
   62 NUT=1
      FIX(KPSI)=2.*FIX(KPSI)
      GO TC 76
   61 IF(AH-PLH)63,63,64
   63 IF(AT-PLT)55,55,64
   64 FIX(KPSI)=.5*FIX(KPSI)
      GO TC 76
C                                                               00000820
   55 NN=0
      TIME=.0
      IF(MY.EG.0)GO TO 4
      WRITE(6,3)ICUT,KPSI
      WRITE(6,103)RHO,RHG,T,H,FIX(KPSI),DT,ANN,TIMEF,DUM,TMAX
      WRITE(6,100)ALPHA
      WRITE(6,100)AW
    3 FORMAT(14I5)
  100 FORMAT(8E15.7)
  103 FORMAT(8E15.7)
C                                                               00000830
    4 IF(DT.GT.(TIMEF-TIME))  DT=TIMEF-TIME
      DO 85 J=1,N
   85 HXY(J)=AW(J)
C                                                               00000870
   40 CALL FG
C                                                               00001940
      IF(MY.GT.1)WRITE(6,100)A,C
C                                                               00001990
    6 CALL PACE
C                                                               00002010
      IF(MY.GT.0)WRITE(6,100)C
      DO 65 J=1,N
      IF((C(J)+AW(J)+FUDGE).LT..0)GO TO 19
      IF((C(J)+C(J)+AW(J)).LT..0)NUT=NUT+2
      IF((C(J)+AW(J)).LT..0)C(J)=-AW(J)
   65 CONTINUE
C                                                               00000640
   66 DUM=.0
      DO 73 J=1,N
      AW(J)=AW(J)+C(J)
   73 DUM=DUM+AW(J)
      GUM=DUM+AW(NP)
C                                                               00000620
C     USE ENTHALPY TO FIND TEMPERATURE
```

```
  134 TT=T
      HA=-H
      HB=.0
      HC=.0
      DO 38 J=1,NALL
      DO 33 K=1,6
      IF(TT.LE.HF(1,K,J))GO TO 36
   33 CONTINUE
   36 AWW(J)=AW(J)*W(J)
      IF(AW(J).LE..0)GO TO 38
      HA=HA+HF(2,K,J)*AWW(J)
      HB=HB+HF(3,K,J)*AWW(J)
      HC=HC+HF(4,K,J)*AWW(J)
   38 CONTINUE
      IF (ABS (HC-0.) .LT.  .00001)  GO TO 37
      HD=-.5*HB/HC
      HE=HD*HC-HA/HC
      IF(HE.LT..0)GO TO 37
      T=HD+SQRT(HE)
      GO TO 39
   37 T=-HA/HB
   39 CONTINUE
C                                                                       ONE01710
  135 RHO=PRESS/T/GUM
      IF(MY.GT.0)WRITE(6,100)AWW,RHO,T,TT,FIX(KPSI),DT,TIME,TIMEF
C                                                                       G0002600
   20 TIME=TIME+DT
   43 AH=ABS(RHO/RW-1.)
      AT=ABS(T/TW-1.)
      DO 86 J=1,N
      IF((HXY(J).LE.TESTY).OR.(AW(J).LE.TESTY))GO TO 86
      TESTW=1.+.00464/HXY(J)**.333
      TESTX=1./TESTW
      TEST=AW(J)/HXY(J)
      IF((TEST.GT.TESTW).OR.(TEST.LT.TESTX))GO TO 84
   86 CONTINUE
      IF((KTEST+NUT).GT.1) GO TO 81
      IF(AH-PLH)80,81,81
   80 IF(AT-PLT)82,81,81
   82 DT=2.*DT
      FIX(KPSI)=2.*FIX(KPSI)
      NUT=NUT+1
      IF(FIX(KPSI).LE.TMAX) GO TO 44
      FIX(KPSI)=TMAX
      IF(DT.LE.TMAX)GO TO 44
      DT=TMAX
      GO TO 44
   81 IF(AH-PLH)83,83,84
   83 IF(AT-PLT)44,44,84
   84 DT=DT/2.
      FIX(KPSI)=DT
   44 NN=NN+1
      TW=T
      RW=RHO
      DO 23 J=1,N
   23 HXY(J)=AW(J)
      IF(TIME.LT.TIMEF)GO TO 4
      TEMR=T
  500 DO 501 J=1,NALL
  501 ALPHI(J)=AW(J)*W(J)
```

```
      THLD(KPSI)=T
      HHLD(KPSI)=RHO
      IF(MY.GT.0)WRITE(6,100)ALPHI,TEMP
      RETURN
C                                                                 00002680
   19 GO TC(21,21,52),KTEST
   21 KTEST=KTEST+1                                                00002700
      DT=DT/1C.
      FIX(KPSI)=DT
      WRITE(6,101)
  101 FORMAT(36H NEGATIVE MASS FRACTICN IN CHEMISTRY)
      WRITE(6,3)ICUT,KPSI
      DO 123 J=1,N
      ALPHA(J)=AW(J)*W(J)
  123 ALP(J)=(AW(J)+C(J))*W(J)
      DO 124 J=NP,NALL
  124 ALP(J)=AW(J)*W(J)
      WRITE(6,100)ALP,DT,T,X
      WRITE(6,100)ALPHA,DT,T,X
      GO TC 2
C                                                                 00002180
   52 IF(MY.EC.0)GO TO 92
   91 TEMP=-5C00.
C     TEMP IS USED AS A FLAG WHEN TROUBLE OCCURS,                  00002510
      GO TC 5C0                                                    00002520
   92 MY=1
      IDA(KPSI)=IDA(KPSI)-1
      GO TC 222
      END
```

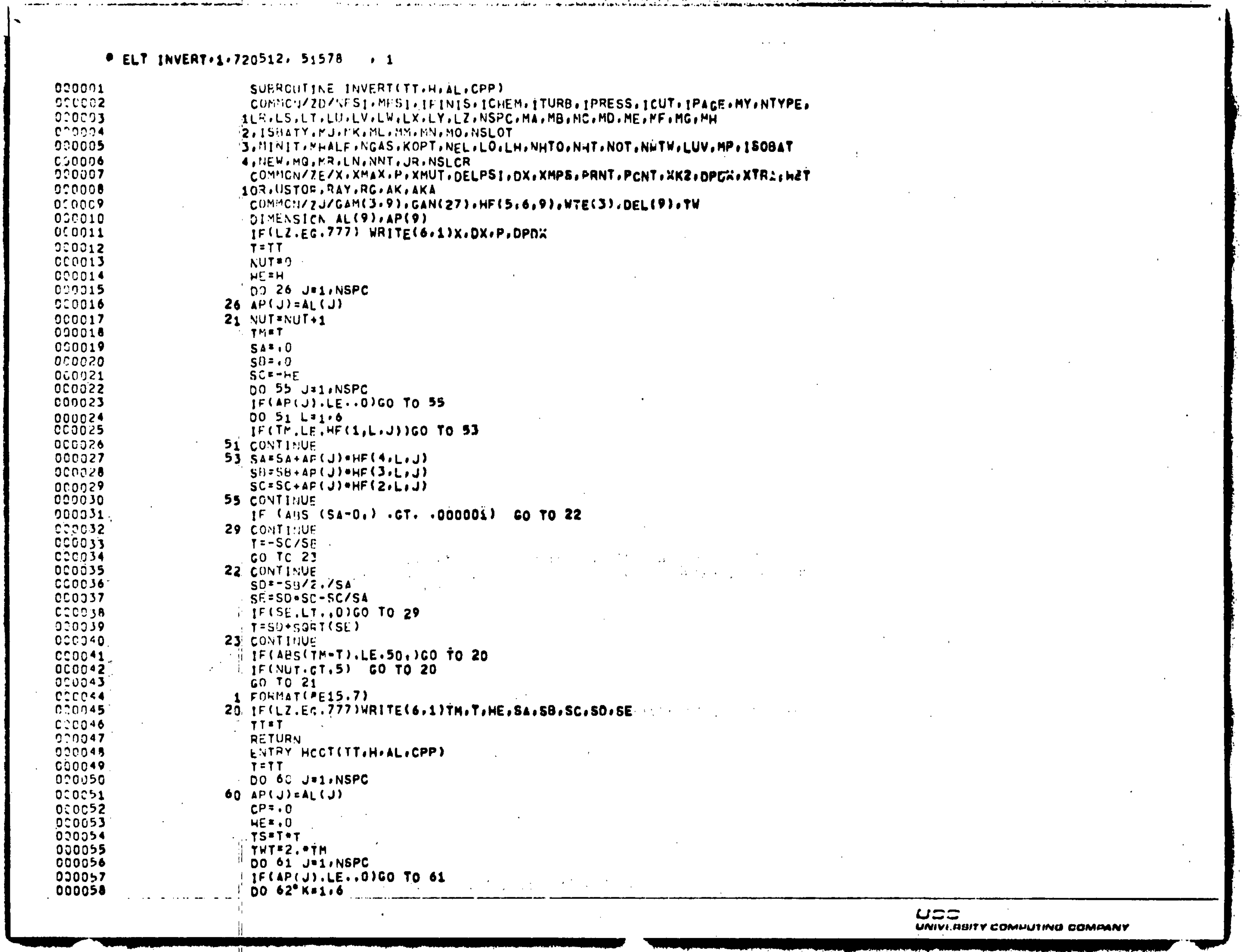
```
 • ELT INJECT,1,720512, 51609   , 1
      SUBROUTINE INJECT (NELEM,DCHAMB,TO,TH,EMR,MPSI,DELPSI)
      EXTERNAL FVEL
      COMMON/NJECT/WT,PO,FM
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XMU(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      DIMENSION R(60),RFD(60),PSIINJ(60),PSIS(99),RF(99)
      DIMENSION WCUM(60)
      COMMON /NJEC2/ B
C
      DATA PI/3.14159/
      DATA ALFOX,ALFUEL/'  OX','FUEL'/
C
C         FM        PER CENT OX OVERALL, INPUT          •
C         EMR       DESIRED MIXING EFFICIENCY, INPUT    •
C         WOF       OX FLOW IN FUEL RICH REGION
C         WFF       FUEL FLOW IN FUEL RICH REGION
C         WOO       OX   FLOW IN OX   RICH REGION
C         WFO       FUEL FLOW IN OX   RICH REGION
C         PMO       PER CENT OX IN OX   RICH REGION
C         PMF       PER CENT OX IN FUEL RICH REGION
C         EM        CALCULATED MIXING EFFICIENCY
C         WT        TOTAL MASS FLOW, INPUT              •
C         TO        INLET OX TEMP, INPUT •
C         TH        INLET FUEL TEMP, INPUT •
C         PO        INLET PRESSURE, INPUT •
C         NELEM     NUMBER ELEMENTS, INPUT •
C
      RCHAMB=DCHAMB/2.
C                          LOOP TO COMPUTE EM
      KOUNT=0
      OF=FM/(1.0-FM)
      DEL=0.05*FM
 1    WOPP=0.5*FM*WT + DEL
      WOP=WOPP-2.0*DEL
      OF=FM/(1.0-FM)
      WFP =0.5*WT*(1.0-FM) + DEL/OF
      WFPP=WFP-2.0*DEL/OF
C
      DP=WOP + WFP
      DPP=WOPP + WFPP
      FMP=WOP/DP
      FMPP=WOPP/DPP
C                          NOW CALCULATE EFFICIENCY + CHECK AGAINST
C                          DESIRED MIXTURE EFFICIENCY.
C
      EM=1.0 - ((DP/WT)*(FM-FMP)/FM + (DPP/WT)*(FMPP-FM)/(1.0-FM) )
      ADD=(EM-EMR)/EMR
      IF(ABS(ADD) .LE. 0.001)GO TO 2
      DEL=DEL + 0.09*FM*ADD
      KOUNT=KOUNT+1
      IF(KOUNT .GT. 1000)GO TO 500
      GO TO 1
C
C                          CALCULATE TO FOR GAS MIXTURE
 2    CONTINUE
```

UCC
UNIVERSITY COMPUTING COMPANY

```
      TP=(.24*FMP*TO + 3.5*TH*(1.0-FMP))/(3.5-3.26*FMP)
      TPP=(.24*FMPP*TO + 3.5*TH*(1.0-FMPP))/(3.5-3.26*FMPP)
C
C                         CALCULATE MOLECULAR WEIGHT
      FMWP=64.508/(2.016*FMP+(32.0*(1.0-FMP)))
      FMWPP=64.508/(2.016*FMPP+(32.0*(1.0-FMPP)))
C
C                         COMPUTE CROSS SECTIONAL AREAS
      AREA=PI*DCHAMB*DCHAMB/4.0
      ARATIO=SQRT(TP*FMWPP/(TPP*FMWP)) * DP/DPP
      APP=AREA/(1.0+ARATIO)
      AP=AREA-APP
C
C                    CALCULATE NUMBER ANNULI
      NAN=0
      K=0
   15 NAN=NAN + 1
      K=K + 1+6*(NAN-1)
      K=K - NAN + 1
      IF(K .LT. NELEM)GO TO 15
C
C                         CALCULATE RADII
      ADD=DCHAMB/(2*NAN)
      DO 17 I=1,NAN
   17 R(I)=I*ADD
      ADD=AP +APP
      ADD= 0.5*APP/ADD
      RF(1)=SQRT(AP*R(1)*R(1)/(AP+APP))
      RF(2)=R(1)
      K=2
      DO 20 I=2,NAN
      R1=R(I-1)**2
      R2=R(I)**2
      RADD=ADD*(R2-R1)
      K=K+1
      RF(K)=SQRT(R1 + RADD)
      K=K+1
      RF(K)=SQRT(R2 - RADD)
      K=K+1
   20 RF(K)=R(I)
C
C                         CALCULATE VEL IN FUEL RICH REGION
      RHOOC=PC*FMWP/TP/18540.
      ACCEL=SQRT(1.4*18540.0*12.0*32.2*TP/FMWP)
      B=RHCOO*ACCEL
      B=DP/AP/B
      TOL=0.0001
      CALL RTNI(FMACH,F,DF,FVEL,.05,TOL,100,NER)
      IF(NER.NE.0)GO TO 510
      VP=FMACH*ACCEL
      TP=TP/(1.+(.2*(FMACH**2.)))
      RHOVP=VP*RHCOO/((1.+(.2*(FMACH**2.)))**2.5)
C                         CALCULATE VEL IN OX RICH REGION
      RHOOC=PC*FMWPP/TPP/18540.
      ACCEL=SQRT(1.4*18540.0*12.0*32.2*TPP/FMWPP)
      B=RHCOO*ACCEL
      B=DPP/APP/B
      CALL RTNI(FMACH,F,DF,FVEL,.05,TOL,100,NER)
      IF(NER.NE.0)GO TO 520
      VPP=FMACH*ACCEL
```

```
         RHOVPP=VPP*RHOOO/((1.+(.2*(FMACH**2.)))**2.5)
         TPP=TPP/(1.0+(.2*(FMACH**2.)))
C
C  FIND RADII AT EACH FUEL RICH-OX RICH DISCONTINUITY
         J=2
         RFD(1)=RF(1)
         LIMIT=((3*NAN)-1)
         DO 21 I=3,LIMIT,3
         RFD(J)=RF(I)
     600 FORMAT (6F12.5)
         RFD(J+1)=RF(I+1)
         J=J+2
      21 CONTINUE
         LIMIT=2*NAN
         RFD(LIMIT)=RCHAMB
         WCUM(1)=0.0
         WCUM(2)=RHOVP*PI*(RFD(1)**2.)
         WCUM(3)=WCUM(2)+RHOVPP*PI*((RFD(2)**2.)-(RFD(1)**2.))
         IF (NAN.EQ.1) GO TO 7735
         DO 7745 I=3,LIMIT,2
         WCUM(I+1)=WCUM(I)+RHOVP*PI*((RFD(I)**2.)-(RFD(I-1)**2.))
         WCUM(I+2)=WCUM(I+1)+RHOVPP*PI*((RFD(I+1)**2.)-(RFD(I)**2.))
    7745 CONTINUE
    7735 CONTINUE
C  FIND VON MISES COORD AT EACH FUEL RICH-OX RICH DISCONTINUITY
         PSIINJ(1)=0.0
         DO 22 I=1,LIMIT
         PSIINJ(I+1)=RFD(I)*((WCUM(I+1)/(PI*(RFD(I)**2.)))**.5)
      22 CONTINUE
C FIND FLOW PROPERTIES AT PSI COORDINATES
         DEPSI=((WT/PI)**.5)/(MPSI-1)
         DELPSI=.673492*DEPSI
         PSISI(1)=0.0
         T(1)=TP
         U(1)=VP
         YSPEC(4,1)=(1.-FMP)
         YSPEC(5,1)=FMP
         DO 725 I=2,MPSI
         PSISI(I)=PSISI(I-1)+DEPSI
         LIMIT=2*NAN+1
         DO 724 J=1,LIMIT
         IF (PSISI(I).LE.PSIINJ(J))  GO TO 723
     724 CONTINUE
     723 CONTINUE
         DO 722 K=1,LIMIT,2
         IF (K.EQ.J)  GO TO 721
     722 CONTINUE
         GO TO 720
C  SET PROPERTIES FOR PSISI(I) FOR OXIDIZER RICH ZONE
     721 T(I)=TPP
         U(I)=VPP
         YSPEC(4,I)=(1.-FMPP)
         YSPEC(5,I)=FMPP
         GO TO 725
C  SET PROPERTIES FOR PSISI(I) FOR FUEL RICH ZONE
     720 T(I)=TP
         U(I)=VP
         YSPEC(4,I)=(1.-FMP)
         YSPEC(5,I)=FMP
     725 CONTINUE
```

```
C ZERO SPECIES MASS FRACTIONS OTHER THAN H2 AND O2
      DO 726 I=1,MPSI
      DO 727 J=1,3
  727 VSPEC(J,I)=0.0
      DO 728 J=6,9
  728 VSPEC(J,I)=0.0
  726 CONTINUE
      RETURN
C
C                              ERROR CONDITIONS
C
 500  WRITE(6,501) EM
 501  FORMAT('1',5X,'COULD NOT CONVERGE ON MIXING EFFECIENCY IN SUBROUTI
     1NE INJECT'// 5X,'LAST EFFEC= ',1PE15.6)
      CALL EXIT
C
 510  TOL=ALFCX
      GO TO 530
C
 520  TOL=ALFLEL
C
 530  WRITE(6,531) TOL,FMACH
 531  FORMAT('1',5X,'COULD NOT CONVERGE ON MACH NO IN CONTINUITY EQ FOR
     1',A4,' RICH AREA SUBROUTINE INJECT'// 5X,'LAST MACH= ',1PE15.6)
      CALL EXIT
      RETURN
      END
```



```
● ELT INVERT,1,720512, 51578    , 1

                      SUBROUTINE INVERT(TT,H,AL,CPP)
                      COMMON/ZD/NFSI,MFSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
                     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
                     2,ISHATY,MJ,MK,ML,MM,MN,MO,NSLOT
                     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NWTW,LUV,MP,ISOBAT
                     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
                      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,W2T
                     1OR,USTOR,RAY,RG,AK,AKA
                      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
                      DIMENSION AL(9),AP(9)
                      IF(LZ.EQ.777) WRITE(6,1)X,DX,P,DPDX
                      T=TT
                      NUT=0
                      HE=H
                      DO 26 J=1,NSPC
                   26 AP(J)=AL(J)
                   21 NUT=NUT+1
                      TM=T
                      SA=.0
                      SB=.0
                      SC=-HE
                      DO 55 J=1,NSPC
                      IF(AP(J).LE..0)GO TO 55
                      DO 51 L=1,6
                      IF(TM.LE.HF(1,L,J))GO TO 53
                   51 CONTINUE
                   53 SA=SA+AP(J)*HF(4,L,J)
                      SB=SB+AP(J)*HF(3,L,J)
                      SC=SC+AP(J)*HF(2,L,J)
                   55 CONTINUE
                      IF (ABS (SA-0.) .GT. .000001)  GO TO 22
                   29 CONTINUE
                      T=-SC/SB
                      GO TO 23
                   22 CONTINUE
                      SD=-SB/2./SA
                      SE=SD*SC-SC/SA
                      IF(SE.LT..0)GO TO 29
                      T=SD+SQRT(SE)
                   23 CONTINUE
                      IF(ABS(TM-T).LE.50.)GO TO 20
                      IF(NUT.GT.5)  GO TO 20
                      GO TO 21
                    1 FORMAT(8E15.7)
                   20 IF(LZ.EQ.777)WRITE(6,1)TM,T,HE,SA,SB,SC,SD,SE
                      TT=T
                      RETURN
                      ENTRY HCCT(TT,H,AL,CPP)
                      T=TT
                      DO 60 J=1,NSPC
                   60 AP(J)=AL(J)
                      CP=.0
                      HE=.0
                      TS=T*T
                      TWT=2.*TM
                      DO 61 J=1,NSPC
                      IF(AP(J).LE..0)GO TO 61
                      DO 62 K=1,6
```

UCC
UNIVERSITY COMPUTING COMPANY

```
      IF(T.LE.HF(1,K,J))GO TO 63
   62 CONTINUE
   63 CPJ=HF(3,K,J)+TWT*HF(4,K,J)
      HJ=HF(2,K,J)+T*HF(3,K,J)+TS*HF(4,K,J)
      CP=CP+CPJ*AP(J)
      HE=HE+HJ*AP(J)
   61 CONTINUE
      TT=T
      H=HE
      CPP=CP
      RETURN
      END
```

```
00101         SUBROUTINE MARCH
00103         COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
00103        1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XMU(200),U(200)
00103        2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
00103        3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
00103        4,FEE(200)
00104         COMMON/EXTRA/LAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
00104        1,IDA(200),TCHEM(200),WDT(9,200)
00105         COMMON/ZC/WTMCLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
00105        1,GSCALE,TWX(7,4)
00106         COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,IOUT,IPAGR,MY,NTYPE,
00106        1IR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
00106        2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
00106        3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
00106        4,NEW,MQ,MR,LN,NNT,JR,NSLCR
00107         COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
00107        1OR,USTCR,RAY,RG,AK,AKA
00110         COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
00111         COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
00112         COMMON/ZB/CF(200),HSEN(200)
00113         COMMON/ZF/SBL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL,UTBL,CPBL,HEBL
00113        1,RHOBL,RUTBL,WMBL,TBL,GMBL,SSBL,EMBL,PTBL,TTBL
00113      C
00114     100 CONTINUE
00115         IF(ICHEM.EQ.2)GO TO 120
00117         DO 139 I=1,MPSI
```

```
00122            DO 139 K=1,NEL
00125            ALPHA(K,I)=.0
00126            DO 139 J=1,NSPC
00131        139 ALPHA(K,I)=ALPHA(K,I)+GAM(K,J)*YSPEC(J,I)
00131     C
00131     C      COMPUTE FUEL-OXYGEN STOICH RATIO
00135        120 DO 110 I=1,MPSI
00140            IF(ALPHA(LH,I).GT..0)GO TO 112
00142            OF(I)=1.E30
00143            GO TO 113
00144        112 OF(I)=ALPHA(LO,I)/ALPHA(LH,I)
00145        113 CONTINUE
00146            IF(ALPHA(LO,I).GT..0)GO TO 111
00150            FFE(I)=1.E30
00151            GO TO 11
00152        111 FFE(I)=7.9365079*ALPHA(LH,I)/ALPHA(LO,I)
00153        110 CONTINUE
00153     C
00153     C      EMPLOY EQUILIBRIUM(COMPLETE COMBUSTION) CHEMISTRY
00155            IF(ICHEM.EQ.2)CALL EQUILC(1,MPSI)
00155     C
00157         31 KOPT=2
00160            CALL HEAT(1,MPSI)
00160     C
00161            CALL DENSE(1,MPSI)
00161     C
00162            IF(MPSI.EQ.1)GO TO 200
00162     C
00164            CALL BULK
00164     C
00164     C      COMPUTE PHYSICAL COORDINATES
00165            IF(NTYPE.NE.0)GO TO 129
00167        782 Y(1)=PSI(1)/SQRT(RUT(1))
00170            DO 25 I=2,MPSI
00173            IM=I-1
00174         25 Y(I)=SQRT(Y(IM)**2+DELPSI*(PSI(I)/RUT(I)+PSI(IM)/RUT(IM)))
00176            GO TO 32
00177        129 Y(1)=PSI(1)/RUT(1)
00200            DO 135 I=2,MPSI
00203        135 Y(I)=Y(I-1)+.5*DELPSI*(1.0/RUT(I)+1./RUT(I-1))
00203     C
00203     C      COMPUTE VISCOSITY
00205         32 CALL VISC
00206            GO TO 43
00206     C                                                                   05051135
00207        200 IF(NB.EQ.2)GO TO 233
00212            IF(NTYPE.NE.0)GO TO 234
00214            Y(1)=PSI(1)/SQRT(RUT(1))
00215            GO TO 43
00216        234 Y(1)=PSI(1)/RUT(1)
00217            GO TO 43
00220        233 Y(1)=YW
00220     C                                                                   05051104
00221         43 IOUT=IOUT+1
00221     C
00221     C      COMPUTE A  ARRAY
00222            IF(NTYPE.NE.0)GO TO 144
00224            I=1
00225            IF(PSI(1).GT..0)GO TO 2001
00227            A(1)=.0
```

```
00230              I=2
00231         2001 DO 41 I=L,MPSI
00234           41 A(I)=XMU(I)*RHO(I)*U(I)*Y(I)*Y(I)/PSI(I)
00236              GO TO 20
00237          144 DO 142 I=1,MPSI
00242          142 A(I)=XMU(I)*RLT(I)
00242     C
00244              IF(MPSI.EQ.1)GO TO 300
00244     C
00246           20 CALL WALL
00246     C
00246     C        DETERMINE STEP SIZE
00247          300 CONTINUE
00250              CALL STEP
00250     C
00250     C        DETERMINE WHETHER TO OUTPUT
00251          350 CONTINUE
00252          555 IF ((ICUT.LE.1).OR.(IFINIS.GE.2))  GO TO 600                     **-5
00255              KKK=KKK+1
00256              PCNT=PCNT+DX
00256     C                                  REMOVE THIS WRITE STATEMENT LATER
00257              WRITE (6,6001) X,XMAX
00263         6001 FORMAT (6X,'X=',F10.4,'XMAX=',F10.4)
00264              IF(X.LT.XMAX)GO TO 401
00266              IFINIS=2
00267              GO TO 600
00270          401 IF(PRNT.LE.PCNT)GO TO 600
00273              IF(KKK.GT.MC)GO TO 600
00275              GO TO 700
00276          600 CALL PRINT
00277              KKK=0
00300              PCNT=0.0
00301              IF(IFINIS.EQ.2)RETURN
00301     C
00303          700 IF(ICHEM.EQ.2)GO TO 50
00306              DO 710 I=1,MPSI
00311              DO 710 J=1,NSPC
00314          710 WDT(J,I)=.0
00314     C        PERFORM FINITE RATE CHEMISTRY COMPUTATION
00317              IF(ICHEM.EQ.3)CALL PREPAR
00321           50 RX=X+DX
00322              IF(MB.GT.1)GO TO 38
00322     C        OBTAIN DPDX FROM INPUTTED POLYNOMIALS FOR P(X)
00324              CALL PRESS(RP,DPDX,RX,CGP,XP,LY)
00325              GO TO 37
00325     C        PERFORM ITERATIVE PROCESS TO OBTAIN DPDX
00326           38 CALL AREA
00327              RP=P+DPDX*DX
00327     C
00327     C        SOLVE EXPLICIT CONSERVATION EQUATIONS
00330           37 CALL CONSEV
00330     C
00331              DO 400 I=1,MPSI
00334              IF(RU(I).LE..0)  GO TO 500
00336              IF(ICHEM.NE.2) GO TO 420
00340              DO 410 K=1,NEL
00343              IF((RALPHA(K,I)+1.E-8).LT..0)GO TO 500
00345          410 CONTINUE
00347              GO TO 53
00350          420 CONTINUE
```

```
00351   147.          DO 400 J=1,NSPC
00354   148.          IF((YSPEC(J,I)+1.E-8).LT.0.0)  GO TO 500
00356   149.      400 CONTINUE
00361   150.          GO TO 53
00361   151.   C                                     PROGRAM TERMINATED-NEGATIVE SPECIES
00362   152.      500 CONTINUE
00363   153.          WRITE(6,501)RX,DX,IOUT
00370   154.      501 FORMAT(40X,30HNEGATIVE CONSERVATION VARIABLE/40X,3HRX=E15.7,
00370   155.        1 5X,3HDX=E15.7,5X,5HIOUT=I5)
00371   156.          DO 505 K=1,MPSI
00374   157.      504 FORMAT(8E15.7)
00375   158.          WRITE(6,3)K
00400   159.        3 FORMAT(14I5)
00401   160.          WRITE(6,504)(YSPEC(L,K),L=1,NSPC)
00407   161.      505 WRITE(6,504)RL(K),RT(K),(RALPHA(L,K),L=1,NEL)
00420   162.          IFINIS=2
00421   163.          GO TO 63
00421   164.   C
00422   165.       53 IF(IFINIS)2,1,2
00422   166.   C
00422   167.   C      SET HALVING MESH CODE FIRST TIME THRU
00426   168.        1 IFINIS=1
00427   169.          MINIT=IPRESS
00430   170.          MHALF=2*MINIT-1
00430   171.   C
00430   172.   C      N+1  INTO  N   AND  INCREMENT  X
00431   173.        2 X=RX
00432   174.          P=RP
00433   175.          DO 11 I=1,MPSI
00436   176.          IF(ICHEM.NE.2)GO TO 7
00440   177.          DO 5 J=1,NEL
00443   178.        5 ALPHA(J,I)=RALPHA(J,I)
00445   179.          GO TO 8
00446   180.        7 CONTINUE
00447   181.          DO 6 J=1,NSPC
00452   182.        6 YSPEC(J,I)=RYSPEC(J,I)
00454   183.        8 CONTINUE
00455   184.          H(I)=RH(I)
00456   185.          TELAP(I)=TELAP(I)+.5*DX*(1./U(I)+1./RU(I))
00457   186.       11 U(I)=RU(I)
00457   187.   C
00461   188.          IF(MPSI.EQ.1)GO TO 100
00463   189.          IF(ISOBAT.GT.2)CALL FLUX
00465   190.          CALL GRID
00466   191.          GO TO 100
00467   192.          END

           END OF UCC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME   00:00.469
PHASE 2 TIME   00:00.045
PHASE 3 TIME   00:00.975
PHASE 4 TIME   00:00.067
PHASE 5 TIME   00:00.632
PHASE 6 TIME   00:00.574

TOTAL COMPILATION TIME =   00:02.707
    MARCH            SYMBOLIC                                     (FN5)   12 MAY 72  14:19:21  0  00116754    14  195  (DELETED)
    MARCH    CODE    RELOCATABLE                                  (FN5)   12 MAY 72  14:19:21  1  00123526    72    1  (DELETED)
```

```
@ ELT NOZ,1,720512, 51649      , 1

      SUBROUTINE NOZ(RT,RE,XN,DADX,AR,PERBEL)                           NOZ 10
      DIMENSION DADX(22),AR(22)                                         NOZ 20
C PROGRAM TO COMPUTE  NOZZLE AREA RATIO GRADIENT AS FUNCTION OF AREA RATIONOZ 30
C RT = THROAT RADIUS - INCHES                                           NOZ 40
C RE = EXIT RADIUS  -  INCHES                                           NOZ 50
C XN = LENGTH OF NOZZLE- INCHES                                         NOZ 60
C DADX = NOZZLE GRADIENT                                                NOZ 70
C AR   = AREA RATIOS CORRESPONDING TO NOZZLE GRADIENT                   NOZ 80
C  ADES= DESIGN AREA RATIO
C PERBEL=PERCENT BELL IN PERCENT
      XRT=XN/RT                                                         NOZ 90
      XAT=RE/RT                                                         NOZ100
      ADES=10**(32.22-1.034*PERBEL+.01132*PERBEL**2-(4.167E-5)*PERBEL   NOZ110
     1**3)                                                              NOZ120
      AR(1)=1.001
      DO 30 I=1,10                                                      NOZ121
      AR(I+1)=AR(I)+.05                                                 NOZ122
   30 CONTINUE
      DO 31 I=11,15                                                     NOZ124
   31 AR(I+1)=AR(I)+.1
      DO 32 I=16,21                                                     NOZ127
   32 AR(I+1)=AR(I)+.5
      ADES=ADES*(RE/RT)**2                                              NOZ131
      DO 10 I=1,22                                                      NOZ132
      IF(AR(I).LE.1.15) GO TO 20                                        NOZ140
      DADX(I)=-.9203+(.0029*ADES)+(2.1024*AR(I))-((7.875E-6)*ADES**2)   NOZ150
     1-(.632*AR(I)**2)+((4.832E-9)*ADES**3)+(.068*AR(I)**3)+(.0005376
     2*ADES*AR(I))-((5.692E-8)*(ADES*AR(I))**2)                         NOZ170
      GO TO 10
   20 DADX(I)=(AR(I)-1)*6.2                                             NOZ190
   10 CONTINUE
      RETURN                                                            NOZ210
      END                                                               NOZ220
```

```
 * ELT PADE,1,720512, 51623     , 1

             SUBROUTINE PADE
             COMMON/FCG/TMP,HC,ALPHI(9),KPSI,A(8,8),AW(9),B(8),RHOB,TIMEF,DT   PADE
             COMMON/FGY/C(8,8),AS(8,8),N,L,D
             DOUBLE PRECISION S
             C1=DT*DT/12.
             C2=.5*DT
             DO 30 I=1,N
             DO 30 J=1,N
             S=0.0
          20 DO 25 K=1,N
          25 S=S+A(I,K)*A(K,J)
          30 AS(I,J)=S
             DO 50 I=1,N
             B(I)=DT*B(I)
             DO 40 J=1,N
          40 C(I,J)=C1*AS(I,J)-C2*A(I,J)
          50 C(I,I)=C(I,I)+1.0
          60 CALL PICRT
             RETURN
             END
```

```
00101            SUBROUTINE PERF(RT,RE,XN,PERBEL,PT,EY,OFS,RU,R,W,MW,TG,NS,CF,
00101           1TIME,PC,OFINPT,EINPOT)
00103            COMMON/FCF/T,PER
00103      C  PER IS THE PERCENT/100  OF FUEL FLOW AT THE INJECTION POINTS
00103      C  T  IS THE INJECTION TEMPERATURE OF THE COOLANT HYDROGEN
00104            DIMENSION DADX(22),AR(22),EY(200),SIF(200),SIS(200),OFS(200)
00105            DIMENSION A(200),W(200),DMIN(6,22),AFRZ(6),ISPK(6),SIK(200),R(200)
00106            DIMENSION RU(200)
00107            REAL ISPK,MW
00107      C  THE NEXT 7 STATEMENTS CONVERTS THE OUTPUT FROM THE COMBUSTION
```

```
00107      C PROGRAM WHICH ARE IN THE SI UNITS TO THE OLD FASHIONED LB SEC IN UNITS
00110            DO 20 I=1,NS
00113            EY(I)=EY(I)*(((OFS(I)/(1.+OFS(I)))*2.713E5)+(4.2006E6/OFS(I)))
00114            EY(I)=EY(I)/4.184E3
00115            R(I)=R(I)*39.37
00116            RU(I)=RU(I)*.00014503
00117         20 W(I)=W(I)*.001422
00121            PT=PT*.00014503
00122            EINPUT=EINPOT/1.8
00123            TG=TG*1.8
00123      C  RT   RADIUS THROAT  INCHES
00123      C  RE   RADIUS EXIT   INCHES
00123      C  XN   LENGTH FROM THROAT TO EXIT  INCHES
00123      C  PERBEL   PERCENT BELL   PERCENT
00123      C  PT  TOTAL AVERAGE PRESSURE FROM COMBUSTION PROGRAM   PSIA
00123      C  EY   TOTAL ENTHALPY  EACH STREAMLINE(NS)  CAL/GRAM
00123      C  OFS  MIXTURE RATIO OF EACH STREAMLINE  (NS)
00123      C  CSTAR  CHARACTERISTIC VELOCITY  FT/SEC
00123      C  R  RADIUS DIMENSION OF EACH STREAMLINE  INCHES
00123      C  W   W/A  EACH STREAAMLINE  LB/IN SQ SEC
00123      C  MW  MOLECULAR WEIGHR   AVE
00123      C  TG   GAS TEMPERATURE  TOTAL AVE  DEG R
00123      C  TIME  TIME FROM START OF VALVE ELECTRICAL SIGNAL
00123      C  CF  THRUST COEFICIENT BASED ON PC
00123      C  PC  CHAMBER PRESSURE FROM DYNAMIC PROGRAM
00124            C=(RE/RT)**2
00125            CALL NOZ(RT,RE,XN,DADX,AR,PERBEL)
00126            CALL SI(PT,C ,EY,SIF,SIS,OFS,NS)
00127            WRITE(6,200) (EY(I),I=1,200)
00135            WRITE(6,200) (SIF(I),I=1,200)
00143            WRITE(6,200) (SIS(I),I=1,200)
00151            WRITE(6,200) (OFS(I),I=1,200)
00157            WRITE(6,200) PT,A
00166            CALL GRAD(DADX,RT,AR,NS,PT,DMIN)
00167            CALL CRSPLT(DMIN,AR,AFRZ,PT)
00170            CALL PERSPF(AFRZ,PT,C,ISPK)
00171            CALL SIDEL(SIF,SIS,PT,OFS,NS,ISPK,SIK)
00172            WRITE(6,200)(SIS(I),I=1,NS)
00200            WRITE(6,200)(SIF(I),I=1,NS)
00206            WRITE(6,200)(OFS(I),I=1,NS)
00214            WRITE(6,200)(SIK(I),I=1,NS)
00222            WRITE(6,200)(ISPK(I),I=1,6)
00230            CALL DIV(RE,RT,DVL)
00231            A(1)=(((R(2)+R(1))/2.)**2)*3.14159
00232            NN=NS-1
00233            DO 10 I=2,NN
00236         10 A(I)=((((R(I)+R(I+1))/2.)**2)-(((R(I)+R(I-1))/2.)**2))*3.14159
00240            A(NS)=((R(NS)**2)-((R(NS)+R(NS-1))/2.)**2)*3.14159
00241            WW=0
00242            S=0
00243            DO 11 I=1,NS
00246            W(I)=W(I)*A(I)
00247            S=SIK(I)*W(I)+S
00250         11 WW=W(I)+WW
00250      C  SPECIFIC IMPLLSE AFTER KINETIC LOSS
00252            SDK=S/WW
00252      C   SPECIFIC IMPULSE LOSS DUE TO BOUNDARY LAYER
00253            CALL BL(RE,RT,PT,WW,MW,TG,DF)
00254            SDD=DF/WW
00255            SS=0
```

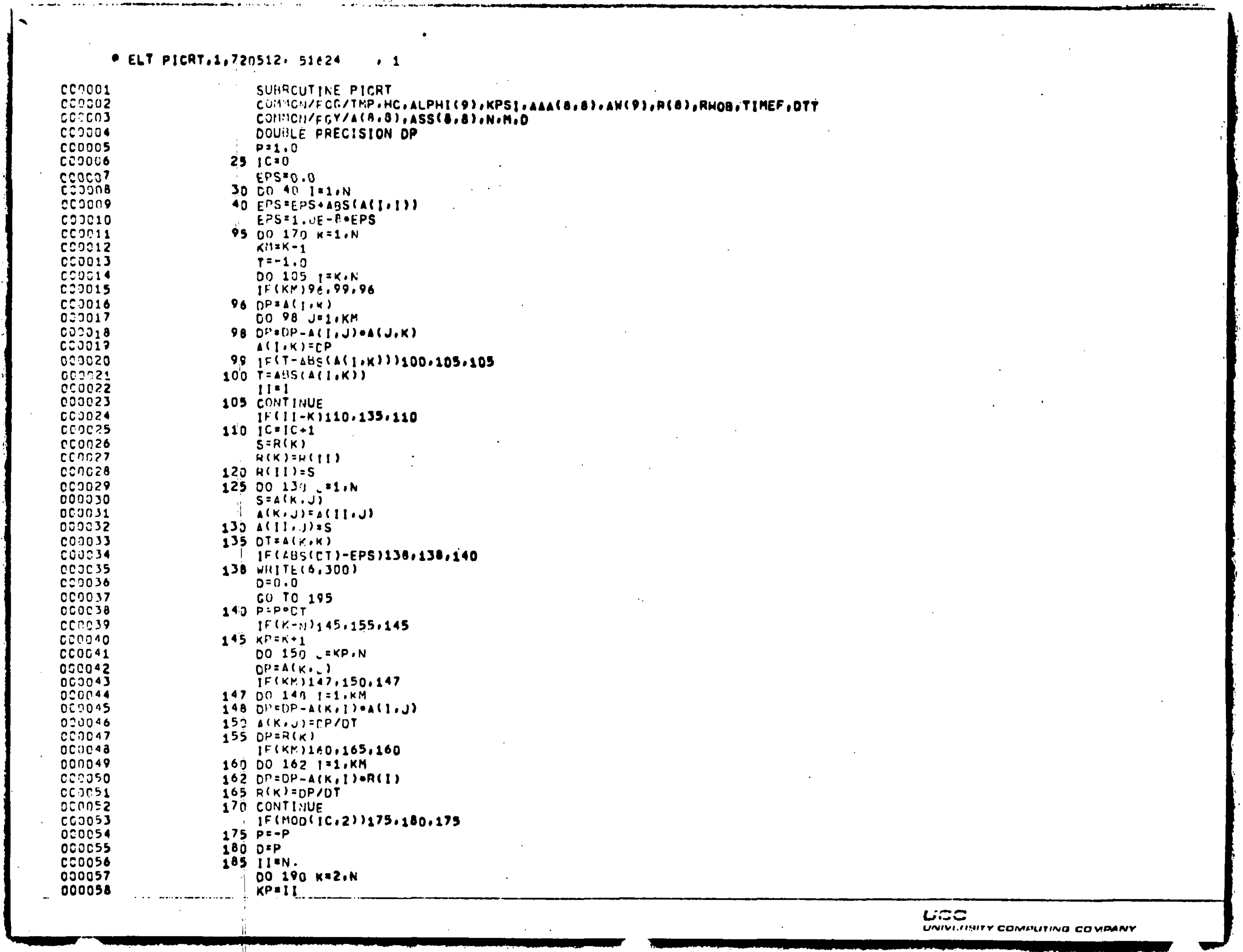
```
00256     71.          DO 13 I=1,NS
00261     72.       13 SS=SIS(I)*W(I)+SS
00263     73.          SSCOMB=SS/WW
00264     74.          OF=0
00265     75.          DO 14 I=1,NS
00270     76.       14  OF=OFS(I)*W(I)+OF
00272     77.          OF=OF/WW
00272     78.     C  EINPUT AND OFINPT ARE THE ENTHALPY AND MIXTURE RATUO AT THE INJECTOR
00273     79.          CALL SI(PT,C ,EINPUT,SIF,SIS,OF     ,1)
00274     80.          ST=SIS(1)
00274     81.     C  SPECIFIC IMPULSE LOSS DUE TO DIVERGENCE
00275     82.          SDIV=SI*(1.-DVL)
00276     83.          SDEL=SDK-SDD-SDIV
00277     84.          FDEL=SDEL*WW
00300     85.          WRITE(6,102)
00302     86.      102 FORMAT(1H1)
00303     87.          WRITE (6,100) TIME,SDEL,FDEL,OF
00311     88.      100 FORMAT(44H THE PERFORMANCE OF THE ENGINE AT TIME= ,F6,3,
00311     89.        18H SEC IS ,F6.2,//15H THE THRUST IS ,F8.2,4H LBS,//22H THE MIXTURE
00311     90.        1 RATIO IS ,F5.2)
00312     91.          WRITE(6,106) SSCOMB
00315     92.      106 FORMAT(57H THE ISP AFTER COMBUSTION BUT WITHOUT NOZZLE LOSSES IS =
00315     93.        1 ,F7.2)
00316     94.          WRITE(6,101) ST, SDK,SDD,SDIV
00324     95.      101  FORMAT(22H THEORETICAL ISP IS = ,F6.2/15H KINETIC ISP = ,F6.2/
00324     96.        U 30H LOSS DUE TO BOUNDARY LAYER = ,F6.2/
00324     97.        2 30H LOSS DUE TO DIVERGENCE     = ,F6.2)
00325     98.          IF(PER.LT..01) GO TO 51
00327     99.          ARA=(RE/RT)**2
00330    100.          ARA=ALOG(ARA)
00331    101.          S=211.75+42.07*ARA-10.94*ARA**2+.94066*ARA**3
00332    102.          S=S*(T/530.)**.5
00333    103.          WWS=(PER*WW)/((1.-PER)*(1.+OF))
00334    104.          FF=S*WWS
00335    105.          WWT=WWS+WW
00336    106.          FF=FF+FDEL
00337    107.          DS=FF/WWT
00340    108.          WRITE(6,105)FF,DS
00344    109.      105 FORMAT(//58H DUE TO MASS ADDITION IN THE SUPERSONIC REGION THE THR
00344    110.        1UST /18H HAS INCREASED TO ,F7.2,4H LBS//26H THE DELIVERED ISP IS N
00344    111.        20W ,F7.2,5H SEC,)
00345    112.          FDEL=FF                                                       *NEW
00346    113.       51 CF=FDEL/(PC*(RT**2)*3.14159)
00347    114.          RETURN                                                        ***4
00350    115.          END

         END OF UCC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)

PHASE 1 TIME   00:00.325
PHASE 2 TIME   00:00.033
PHASE 3 TIME   00:00.744
PHASE 4 TIME   00:00.058
PHASE 5 TIME   00:00.494
PHASE 6 TIME   00:00.366

TOTAL COMPILATION TIME =   00:02.026
    PERF             SYMBOLIC                    (FN5)   12 MAY 72  14:20:27  0  00310224    14  126  (DELETED)
    PERF     CODE    RELOCATABLE                 (FN5)   12 MAY 72  14:20:27  1  00313570    36    1  (DELETED)
                                                                              0  00313634    14   80
```

UCC
UNIVERSITY COMPUTING COMPANY

```
 • ELT PERSHF,1,720512, 51634   , 1

              SUBROUTINE PERSHF(AFRZ,PT,A,ISPK)
              DIMENSION AFRZ(6),ISPK(6),ISP(6,2)                             PER  20
        C THIS SUBROUTINE CALCULATES THE PERCENT OF RECOMBINATION             PER  30
        C THE FIRST INTERGER OF ISP  IS THE MIXTURE RATIO                     PER  40
        C AT O/F= 4.6,AND 8, AT THE LOW + HIGH PRESSURE RESPECTIVELY          PER  50
        C I.E.  OF=4,6,8 AT PT=LOW,THAN OF=4,6,8 AT PT=HIGH, THE SECOND       PER  60
        C INTERGER IS THE LOW + HIGH AREA RATIO                               PER  70
              REAL ISP
              REAL ISPK
              IF(PT.LT.500.) GO TO 10
              P1=AFRZ(1)                                                     PER 100
              P2=AFRZ(2)                                                     PER 110
              P3=AFRZ(3)                                                     PER 120
              P4=AFRZ(4)                                                     PER 130
              P5=AFRZ(5)                                                     PER 140
              P6=AFRZ(6)                                                     PER 150
              GC TC 11                                                       PER 160
          10  P1=AFRZ(4)                                                     PER 170
              P2=AFRZ(5)                                                     PER 180
              P3=AFRZ(6)                                                     PER 190
              P4=AFRZ(1)                                                     PER 200
              P5=AFRZ(2)                                                     PER 210
              P6=AFRZ(3)                                                     PER 220
          11  CCNTINJE                                                       PER 230
        C  P1,P2,P3 ARE 500 PSIA VALUES                                       PER 240
        C  ISP AT PT=500  AR= 40 , OF= 4,6,8                                  PER 250
              ISP(1,1)=.1879+.6958*P1-.20359*P1**2+.01938*P1**3
              ISP(2,1)=-.918+1.7605*P2-.54961*P2**2+.054674*P2**3
              ISP(3,1)=-.969+1.7861*P3-.55411*P3**2+.054977*P3**3
              IF(A.LT.40.) GO TO 20
              ISP(1,2)=-.645+1.2996*P1-.38278*P1**2+.0369409*P1**3
              ISP(2,2)=-.762+1.3555*P2-.39509*P2**2+.037618*P2**3
              ISP(3,2)=-.671+1.1912*P3-.34658*P3**2+.0333923*P3**3
              GC TC 30
          20  ISP(1,2)=ISP(1,1)
              ISP(2,2)=ISP(2,1)
              ISP(3,2)=ISP(3,1)
              ISP(1,1)=-6.552+14.96*P1-10.2*P1**2+2.375*P1**3
              IF(ISP(1,1).GT.1.) ISP(1,1)=1.
              IF(P1.GT.1.6) ISP(1,1)=1.
             ISP(2,1)=-14.96+34.3883*P2-24.925*P2**2+6.0417*P2**3
              IF(ISP(2,1).GT.1.)  ISP(2,1)=1.
              IF(P2.GT.1.6) ISP(2,1)=1.
             ISP(3,1)=-4.852+8.5725*P3-3.1375*P3**3
              IF(ISP(3,1).GT.1.) ISP(3,1)=1.
              IF(P3.GT.1.4) ISP(3,1)=1.
          30  IF(PT.LT.500.) GO TO 100
              ISP(4,1)=-.2455+.81319*P4-.16853*P4**2+.010798*P4**3
              ISP(5,1)=-.9741+1.883*P5-.67787*P5**2+.103794*P5**3-.00557845*
           1  P5**4
              ISP(6,1)=-.812+1.69844*P6-.61506*P6**2+.095198*P6**3-.005157*P6
           1  **4
              IF(A.LT.40) GO TO 40
              ISP(4,2)=-.951+1.8118*P4-.680*P4**2+.108416*P4**3-.005993*P4**4
              ISP(5,2)=-.861+1.6185*P5-.604*P5**2+.096218*P5**3-.005311*P5**4
              ISP(6,2)=-.707+1.3259*P6-.475*P6**2+.074226*P6**3-.004057*P6**4
              DO 41 I=1,6
              ISPK(I)=ISP(I,1)+(ALOG(A)-3.68888)*((ISP(I,2)-ISP(I,1))/.9163)
```

```
      IF(ISPK(I).GT.1.) ISPK(I)=1.
      IF(ISPK(I).LT.0) ISPK(I)=0
   41 CONTINUE
         GO TO 50
   40    ISP(4,2)=ISP(4,1)
         ISP(5,2)=ISP(5,1)
         ISP(6,2)=ISP(6,1)
         ISP(4,1)=-1.975+3.7875*P4-1.1875*P4**2
         IF(ISP(4,1).GT.1.) ISP(4,1)=1.
         IF(P4.GT.1.4) ISP(4,1)=1.
         ISP(5,1)=-7.+15.833*P5-10.9375*P5**2+2.604167*P5**3
         IF(ISP(5,1).GT.1.) ISP(5,1)=1.
         IF(P5.GT.1.6) ISP(5,1)=1.
         ISP(6,1)=-13.15+29.91*P6-21.25*P6**2+5.05208*P6**3
         IF(ISP(6,1).GT.1.) ISP(6,1)=1.
         IF(P6.GT.1.6) ISP(6,1)=1.
         DO 31  I=1,6
         ISPK(I)=ISP(I,1)+(ALOG(A)-2.99573)*((ISP(I,2)-ISP(I,1))/.69314)
      IF(ISPK(I).GT.1.) ISPK(I)=1.
      IF(ISPK(I).LT.0) ISPK(I)=0
   31 CONTINUE
         GO TO 50
  100    ISP(4,1)=ISP(1,1)                                              PER 740
         ISP(5,1)=ISP(2,1)                                              PER 750
         ISP(6,1)=ISP(3,1)                                              PER 760
         ISP(4,2)=ISP(1,2)                                              PER 770
         ISP(5,2)=ISP(2,2)                                              PER 780
         ISP(6,2)=ISP(3,2)                                              PER 790
         ISP(1,1)=-.2195+.9871*P1-.2763*P1**2+.025598*P1**3
         ISP(2,1)=-.432+1.107*P2-.3057*P2**2+.027976*P2**3
         ISP(3,1)=-.573+1.2104*P3-.3366*P3**2+.031078*P3**3
         IF(A.LT.40.) GO TO 60
         ISP(1,2)=.035+.60391*P1-.1233*P1**2+.007828*P1**3
         ISP(2,2)=-.413+.9752*P2-.2636*P2**2+.023945*P2**3
         ISP(3,2)=-.413+.8706*P3-.2304*P3**2+.020769*P3**3
         DO 61  I=1,6
         ISPK(I)=ISP(I,1)+(ALOG(A)-3.68888)*((ISP(I,2)-ISP(I,1))/.9163)
      IF(ISPK(I).GT.1.) ISPK(I)=1.
      IF(ISPK(I).LT.0) ISPK(I)=0
   61 CONTINUE
         GO TO 50
   60    ISP(1,2)=ISP(1,1)                                              PER 940
         ISP(2,2)=ISP(2,1)                                              PER 950
         ISP(3,2)=ISP(3,1)                                              PER 960
         ISP(1,1)=-.055+1.1625*P1-.3125*P1**2
         IF(ISP(1,1).GT.1.) ISP(1,1)=1.
         IF(P1.GT.1.6) ISP(1,1)=1.
         ISP(2,1)=-.583+1.8175*P2-.523214*P2**2
         IF(ISP(2,1).GT.1.) ISP(2,1)=1.
         IF(P2.GT.1.8) ISP(2,1)=1.
         ISP(3,1)=-.7386+1.895*P3-.51786*P3**2
         IF(ISP(3,1).GT.1.) ISP(3,1)=1.
         IF(P3.GT.1.) ISP(3,1)=1.
         DO 101 I=1,6
         ISPK(I)=ISP(I,1)+(ALOG(A)-2.99573)*((ISP(I,2)-ISP(I,1))/.69314)
      IF(ISPK(I).GT.1.) ISPK(I)=1.
      IF(ISPK(I).LT.0) ISPK(I)=0
  101 CONTINUE
   50    CONTINUE
         RETURN                                                         PER1700
```

```
003119          END                                   PER1800
```



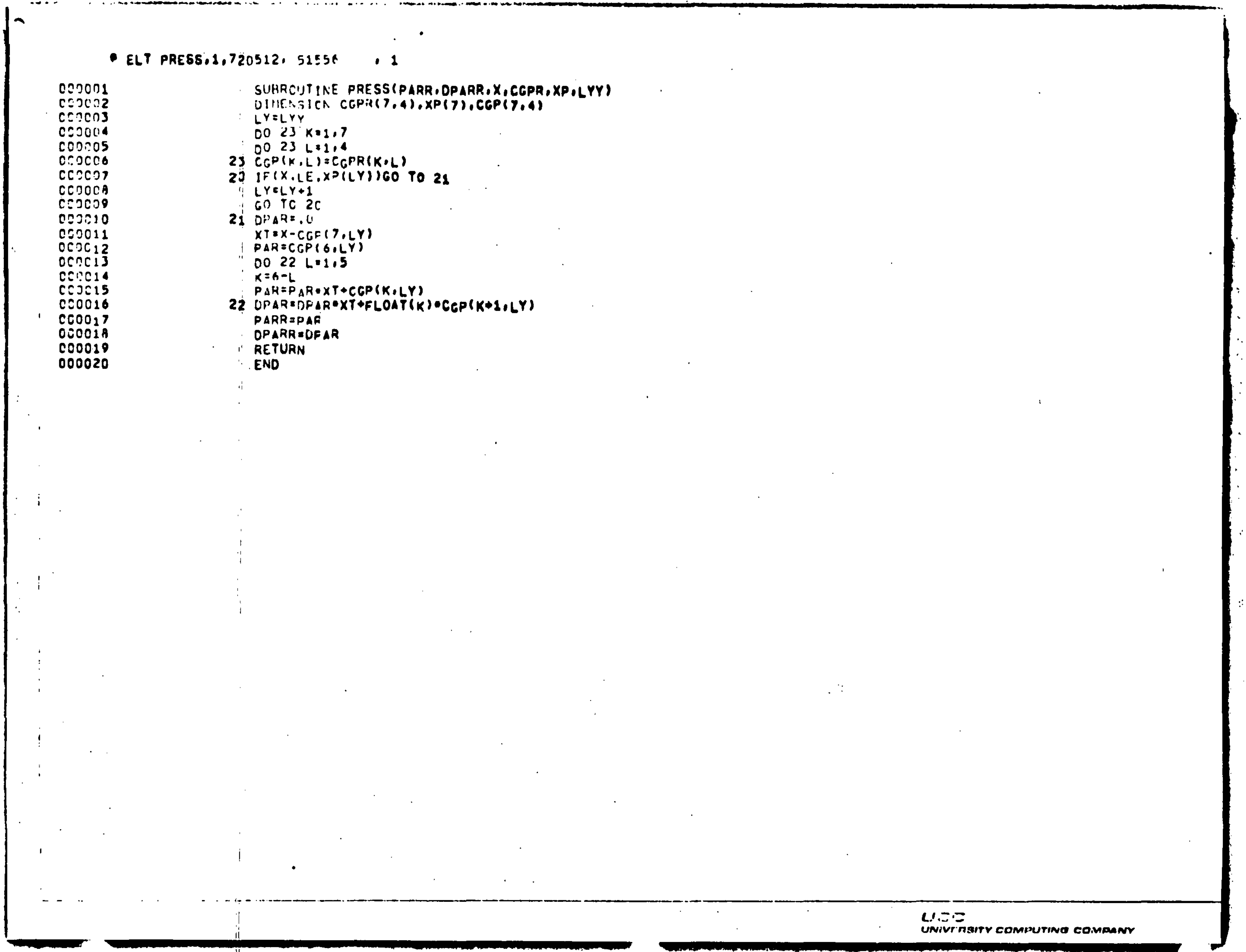
```
 • ELT PICRT,1,720512, 51624     , 1

              SUBROUTINE PICRT
              COMMON/FCG/TMP,HC,ALPHI(9),KPSI,AAA(8,8),AW(9),R(8),RHOB,TIMEF,DTT
              COMMON/FGY/A(8,8),ASS(8,8),N,M,D
              DOUBLE PRECISION DP
              P=1.0
           25 IC=0
              EPS=0.0
           30 DO 40 I=1,N
           40 EPS=EPS+ABS(A(I,I))
              EPS=1.0E-8*EPS
           95 DO 170 K=1,N
              KM=K-1
              T=-1.0
              DO 105 I=K,N
              IF(KM)96,99,96
           96 DP=A(I,K)
              DO 98 J=1,KM
           98 DP=DP-A(I,J)*A(J,K)
              A(I,K)=DP
           99 IF(T-ABS(A(I,K)))100,105,105
          100 T=ABS(A(I,K))
              II=I
          105 CONTINUE
              IF(II-K)110,135,110
          110 IC=IC+1
              S=R(K)
              R(K)=R(II)
          120 R(II)=S
          125 DO 130 J=1,N
              S=A(K,J)
              A(K,J)=A(II,J)
          130 A(II,J)=S
          135 DT=A(K,K)
              IF(ABS(DT)-EPS)138,138,140
          138 WRITE(6,300)
              D=0.0
              GO TO 195
          140 P=P*DT
              IF(K-N)145,155,145
          145 KP=K+1
              DO 150 J=KP,N
              DP=A(K,J)
              IF(KM)147,150,147
          147 DO 148 I=1,KM
          148 DP=DP-A(K,I)*A(I,J)
          150 A(K,J)=DP/DT
          155 DP=R(K)
              IF(KM)160,165,160
          160 DO 162 I=1,KM
          162 DP=DP-A(K,I)*R(I)
          165 R(K)=DP/DT
          170 CONTINUE
              IF(MOD(IC,2))175,180,175
          175 P=-P
          180 D=P
          185 II=N.
              DO 190 K=2,N
              KP=II
```

UCC
UNIVERSITY COMPUTING COMPANY

```
      II=II-1
      DP=R(II)
      DO 188 I=KP,N
  188 DP=DP-A(II,I)*R(I)
  190 R(II)=DP
  195 CONTINUE
  200 RETURN
  300 FORMAT(21HONEAR SINGULAR MATRIX)
      END
```

```
● ELT PREPAR,1,720512, 51599   , 1

      SUBROUTINE PREPAR
C     HYDROGEN/OXYGEN FINITE RATE CHEMISTRY WITH
C     HO2,H2O2 QUENCHING REACTIONS-ORDER OF SPECIES
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XMU(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEC(200)
      COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
     1,IDA(200),TCHEM(200),WDT(9,200)
      COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TWX(7,4)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
     1OR,USTOR,RAY,RG,AK,AKA
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      COMMON/FCG/TMP,HC,AL(9),I,AA(8,8),AW(9),C(8),RHKMS,DT,DTT         PREPAR
      CTM=.0
      KTM=0
      IF(ICUT.EQ.1)HDX=XMPS
      HHDX=XMPS
      IF(ICUT.GT.1)GO TO 20
   29 DO 31 I=1,MF
   31 IDA(I)=0
   11 FORMAT(7E10.6)
    1 FORMAT(8E15.7)
   20 DO 100 I=1,MPSI                                                   WDOT0290
      IM=I-1
      DT=DX/U(I)                                                        WDOT0310
      TMP=T(I)
      DO 30 J=1,NSPC                                                    WDOT0330
   30 AL(J)=YSPEC(J,I)
      HC=SMALLH(I)
      IF((AL(NNT)+1.E-8).GT.1.)GO TO 50
      IF(I.EQ.1)GO TO 32
      IF (ABS (T(I)-T(I-1)) .GT.  .0001)  GO TO 32
      DO 33 J=1,NSPC
      IF (ABS (YSPEC(J,I)-YSPEC(J,IM))  .GT. .00001)  GO TO 32
   33 CONTINUE
      TCHEM(I)=TCHEM(IM)
      DO 34 J=1,NSPC
   34 WDT(J,I)=WDT(J,IM)
      GO TO 100
   32 CONTINUE
      IF(TMP.LT.500.) GO TO 50
      RHKMS=RHO(I)
      CALL HONC
      CTM=CTM+FIX(I)*U(I)
      KTM=KTM+1
      IF(TMP.LT..0)GO TO 22
   50 TCHEM(I)=TMP-T(I)
      DO 60 J=1,NSPC                                                    WDOT0440
   60 WDT(J,I)=AL(J)-YSPEC(J,I)
  100 CONTINUE                                                          WDOT0460
```

```
      IF(KTM.EQ.0)GO TO 200
       DTM=CTM/FLOAT(KTM)*XK(5)
   41 XMPS=DX/DTM*XMPS
      IF(XMPS.LT.HDX) XMPS=HDX
      IF (ABS (HHDX-XMPS) .LT.  .00001)  GO TO 200
      WRITE(6,4)XMPS,X,DX
    4 FORMAT(/40X,15HXMPS CHANGED TO,1P3E15.7)
      GO TO 200
   22 IFINIS=2
   40 WRITE (6,300)                                          WDOT0390
  300 FORMAT(30HOTEMPERATURE DOES NOT CONVERGE)              WDOT0480
  200 RETURN                                                 WDOT0470
      END                                                    WDOT0490
```



```
● ELT PRESS,1,720512, 51556    , 1

                  SUBROUTINE PRESS(PARR,DPARR,X,CGPR,XP,LYY)
                  DIMENSION CGPR(7,4),XP(7),CGP(7,4)
                  LY=LYY
                  DO 23 K=1,7
                  DO 23 L=1,4
               23 CGP(K,L)=CGPR(K,L)
               20 IF(X.LE.XP(LY))GO TO 21
                  LY=LY+1
                  GO TO 20
               21 DPAR=.0
                  XT=X-CGP(7,LY)
                  PAR=CGP(6,LY)
                  DO 22 L=1,5
                  K=6-L
                  PAR=PAR*XT+CGP(K,LY)
               22 DPAR=DPAR*XT+FLOAT(K)*CGP(K+1,LY)
                  PARR=PAR
                  DPARR=DPAR
                  RETURN
                  END
```

UCC
UNIVERSITY COMPUTING COMPANY

ELT PRINT,1,720512, 51584 , 1

```
      SUBROUTINE PRINT
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HPLD(200),FIX(200)
     1,IDA(200),TCHEM(200),WDT(9,200)
      COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TPX(7,4)
      COMMON/ZC/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISRATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NFTW,LUV,MP,ISOBAT
     4,NEW,NQ,NR,LN,NNT,JR,NSLCR
      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
     1OR,USTOR,RAY,RG,AK,AKA
      COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      COMMON/ZL/QLX(7,4),          TCX(7,4),QL,TC,PRB,GAMW,RUC,RUCX(7,4)
     1,SCH,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
     2,GG,UB(21),LBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
     3,VWS(21),RWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,CGG(21)
     4,CPU,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
     5,NCL,NP,PYE,ANP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),WC,HCS(21)
      COMMON/ZR/OF(200),HSEN(200)
      COMMON/ZF/SBL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL,UTBL,CPBL,HEBL
     1,RHOBL,RUTBL,WMBL,TBL,GMBL,SSBL,EMBL,PTBL,TTBL
      DIMENSION FR(200),GR(200),RE(200),UCHAR(200),REG(200),GRRE(200)
     1,GRREQ(200),URE(200),XML(9,200),CPGAS(200),WTGAS(200),SUM(200)
     2,SOX(200),TAG(11)
      EQUIVALENCE (WDT(1),FR),(WDT(201),GR),(WDT(401),RE),(WDT(601),REQ)
     1,(WDT(801),UCHAR),(WDT(1001),GRRE),(WDT(1201),GRREQ)
     2,(WDT(1401),URE)
      EQUIVALENCE (RYSPEC,XML),(CPGAS,CPBAR),(WTGAS,WTMIX),(SOX,FEE)
C     SPECIES ARE 1-H 2-O 3-H2O 4-H2 5-O2 6-OH 7-HO2 8-H2O2 9-DILUENT
      IF(IOUT.GT.1)GO TO 600
      DATA TAG/6HFROZEN,6HEQUILB,6HFINITE,6HAXISYM,6H 2-DIM,6H FREE
     1,6HP SPEC,6HRW SPC,6H   JET ,6H WALL ,6H WALL /
      GC=9.80665
      PG=PYE*GC
    1 FORMAT(8E15.7)
  600 DO 10 I=1,MPSI
C     RH(I) IS TOTAL SENSIBLE ENTHALPY
C
      SUMHF=.0
      DO 40 J=1,NSPC
   40 SUMHF=SUMHF+YSPEC(J,I)*DEL(J)
      RH(I)=H(I)-SUMHF
C
      DM=CPGAS(I)*WTGAS(I)/RG
C
C     RALPHA(1,I) IS RATIO OF SPECIFIC HEATS
      RALPHA(1,I)=DM/(DM-1.)
C     RALPHA(2,I) IS SPEED OF SOUND
      RALPHA(2,I)=SQRT(RALPHA(1,I)*RG/WTGAS(I)*T(I))
C     RALPHA(3,I) IS MACH NUMBER
      RALPHA(3,I)=U(I)/RALPHA(2,I)
      DN=RALPHA(1,I)-1.
      DM=1.+DN/2.*RALPHA(3,I)**2
C     RU(I) IS FROZEN STAGNATION PRESSURE
      RU(I)=P*DM**(RALPHA(1,I)/DN)
C     RT(I) IS FROZEN STAGNATION TEMPERATURE
      RT(I)=T(I)*DM
C
      IF(MPSI.EQ.1)XMU(I)=1.E-6
C
C     COMPUTE SIGMA ALPHA AS A CHECK SUM
      SUM(I)=0.0
      DO 10 J=1,NSPC
      XML(J,I)=YSPEC(J,I)*WTMIX(I)/WTMOLE(J)
   10 SUM(I)=SUM(I)+YSPEC(J,I)
      IF(MPSI.EQ.1)  GO TO 25
C
C
      IF(MB.NE.1)  GO TO 221
      IF(NEW.NE.0)  GO TO 222
      IF(NTYPE.NE.0)  GO TO 223
      YW=SQRT(Y(LR)**2+(PSIW**2-PSI(LR)**2)/RUT(LR))
      GO TO 221
  223 YW=Y(LR)+(PSIW-PSI(LR))/RUT(LR)
      GO TO 221
  222 YW=Y(LR)
  221 CONTINUE
      IF(EBL(LO).LE..0)GO TO 181
      FEEBL=7.9365079*EBL(LH)/EBL(LO)
      GO TO 182
  181 FEEBL=1.E30
  182 IF(EBL(LH).LE..0)GO TO 183
      OFBL=EBL(LO)/EBL(LH)
      GO TO 184
  183 OFBL=1.E30
  184 CONTINUE
      SUMHF=.0
      DO 200 J=1,NSPC
  200 SUMHF=SUMHF+SHL(J)*DEL(J)
      HSBL=HBL-SUMHF
      XMBL=CPBL*WMBL
      GMBL=XMBL/(XMBL-RG)
      SSBL=SQRT(GMBL*RG/WMBL*YBL)
      EMBL=UBL/SSBL
      DN=GMBL-1
      DM=1.+.5*DN*EMBL*EMBL
      PTBL=P*DM**(GMBL/DN)
      TTBL=TBL*DM
      EMD=PSIW
      ARE=YW
      IF(NTYPE.NE.0)GO TO 20
      ARE=ARE*PYE*YW
      EMD=EMD*PYE*PSIW
   20 FI=P*ARE+EMD*UBL
      IF(ICUT.EQ.1)FIH=FI
      FIN=FI-FIH
   25 CONTINUE
      PSIA=P/1.01325E5
      XD=X/XK(7)
      IPAGE=IPAGE+1
C
```

```
      WRITE(6,81)TAG(ICHEM),TAG(NTYPE+4),TAG(MB+6),TAG(MB+9)
   81 FORMAT(////10X,44HHYDROGEN/OXYGEN THRUST CHAMBER(K,M,S. UNITS),5X,
     11A6,2X,9HCHEMISTRY,5X,1A6,2X,4HFLOW,5X,2A6)
      WRITE (6,201)X,(TITLE(I),I=1,12),IPAGE
  201 FORMAT(10X,2HX=E15.7,7H METERS,5X,12A6,5X,4HPAGEI4)
      WRITE (6,102)
  102 FORMAT(6X,9HP(N/M**2),4X,12HLEWIS NUMBER,1X,14HPRANDTL NUMBER,2X,
     1 13HVISC(KG/M-SC),3X,12HSTEP SIZE(M),5X,5HSTEPS)
      WRITE (6,103)P,XLE(1),SIGMA(1),XMU(1),DX,IOUT
  103 FORMAT(1X,5E15.7,5X,I5)
      WRITE (6,107)
  107 FORMAT(4H  PT,5X,3HPSI,11X,8HVEL(M/S),6X,8HT(DEG K),6X,9HCP J/KG-K
     1,5X,9HRHO KG/M3,5X,4HR(M),10X,5HRW/-U,6X,12HMOLECULAR WT,4H  PT)
      DO 70 I=1,MPSI
   70 WRITE(6,209)I,PSI(I),U(I),T(I),CPBAR(I),RHO(I),Y(I),RUT(I),WTMIX(I
     1),I
  209 FORMAT(I4,8E14.6,I4)
      WRITE(6,112)
  112 FORMAT(/6H  PT  ,8HSTATIC H,6X,7HTOTAL H,7X,12HTOTAL SEN. H,2X,5HG
     1AMMA,9X,9HSTOICH O2,5X,7HMACH NO,7X,7HP TOTAL,7X,7HT TOTAL,7X,
     22HPT)
      DO 212 I=1,MPSI
  212 WRITE(6,209)I,SMALLH(I),H(I),RH(I),RALPHA(1,I),SOX(I),RALPHA(3,I),
     1RU(I),RT(I),I
      WRITE(6,9)PSIA,XD,YW,YWP,DYWDX
    9 FORMAT(3H P=E15.7,11H ATM   X/D=E15.7,6H   YW=E15.7,5H  YW=E15.7,
     1 8H  DYWDX=E15.7)
      IF(MPSI.EQ.1)GO TO 31
      WRITE(6,2)
    2  FORMAT(40X,34HBULK MASS'MEAN AVERAGED QUANTITIES)
      WRITE(6,6)TBL,UBL,WMBL,FEEBL,GMBL,EMBL,EBL(LH),CPBL
    6 FORMAT(/5X,4HTBAR,11X,4HUBAR,11X,4HWBAR,11X,6HPHIBAR,9X,9HGAMMA-BA
     1R,6X,4HMBAR,11X,6HELEM H,9X,5HCPBAR/1P8E15.7)
      WRITE(6,3)XMU(1),HBL,OFBL,HEBL,RUTBL,PTBL,TTBL
    3 FORMAT(5X,4HVISC,11X,2HHT,13X,10H O/F RATIO,5X,2HHE,13X, 5HRHO'U
     1,9X,7HTOTAL P,8X,7HTOTAL T/8E15.7)
      WRITE(6,8)EBL(LO),HSBL,RHOBL,XMU(1),GMBL
     1,FI,FIN
    8 FORMAT(2X,7HELEM O=E15.8,2X,3HHS=E15.8,2X,4HRHO=E15.8,2X,
     15HVISC=E15.7,2X,6HGAMMA=E15.7/10H IMP FUNC=E15.7,2X,11HNET THRUST=
     2E15.7)
   31 CONTINUE
C     SPECIES ARE 1-H 2-O 3-H2O 4-H2 5-O2 6-OH 7-HO2 8-H2O2 9-DILUENT
      WRITE (6,108)
  108 FORMAT(4H  PT,5X,1HH,13X,1HO,13X,3HH2O,11X,2HH2,12X,2HO2,12X,2HOH,
     112X,3HHO2,11X,4HH2O2,7X,2HPT)
      DO 80 I=1,MPSI
   80 WRITE(6,209)I,(YSPEC(J,I),J=1,8),I
      DO 110 I=1,MPSI
  110 WRITE(6,209)I,(XML(J,I),J=1,8),I
      WRITE(6,109)
  109 FORMAT(4H  PT,3X,11HTRACE(MASS),3X,11HTRACE(MOLE),5X,6HELEM H,8X
     1,6HELEM C,7X,10H O/F RATIO,4X,10HSUM SPECIE,5X,9HTIME(SEC),3X
     2,11HMOUT(KG/SC),4H  PT)
      DO 213 I=1,MPSI
  213 WRITE(6,209)I,YSPEC(9,I),XML(9,I),ALPHA(LH,I),ALPHA(LO,I),OF(I),
     2SUM(I),
     1TELAP(I),EMDT(I),I
      IF(ISOBAT.EQ.2)WRITE(6,19)QL,TW
   19 FORMAT(23HHEAT TRANS(J/SEC-M**2)=E15.7,5X,6HTW(K)=E15.7)
```

```
      IF(ISOBAT.EG.3)WRITE(6,18)TC,RUC,FF,STAN,GG,RYNS,PSIWA,PSIW
   18 FORMAT(10HT-COOL(K)=E15.7,5X, 9H(RHO-U)C=E15.7,6X,2HF=E15.7,5X,
     13HST=E15.7/3H G=E15.7,5X,4HRE*=E15.7,5X,6HPSIWA=E15.7,5X,
     25HPSIW=E15.7)
      IF(ISOBAT.EG.4)WRITE(6,17)CS,TAW,TW,RSTAR,T(LR),CPC,PRIWA,PSIW
   17 FORMAT(4H C*=E15.7,5X,4HTAW=E15.7,5X,3HTW=E15.7,5X,4HRE*=E15.7/
     14H TE=E15.7,5X,4HCPC=E15.7,5X,6HPSIWA=E15.7,5X,5HPSIW=E15.7)
      RETURN
      END
```

```
 • ELT RNTI,1,720512, 51611      , 1

                 SUBROUTINE RTNI(X,F,DERF,FCT,XST,EPS,IEND,IER)                RTNI 056
       C                                                                       RTNI 057
       C                                                                       RTNI 058
       C         PREPARE ITERATION                                             RTNI 059
                 IER=0                                                         RTNI 060
                 X=XST                                                         RTNI 061
                 TOL=X                                                         RTNI 062
                 CALL FCT(TOL,F,DERF)                                          RTNI 063
                 TOLF=100.*EPS                                                 RTNI 064
       C                                                                       RTNI 065
       C                                                                       RTNI 066
       C         START ITERATION LOOP                                          RTNI 067
                 DO 6 I=1,IEND                                                 RTNI 068
                 IF(F)1,7,1                                                    RTNI 069
       C                                                                       RTNI 070
       C         EQUATION IS NOT SATISFIED BY X                                RTNI 071
               1 IF(DERF)2,8,2                                                 RTNI 072
       C                                                                       RTNI 073
       C         ITERATION IS POSSIBLE                                         RTNI 074
               2 DX=F/DERF                                                     RTNI 075
                 X=X-DX                                                        RTNI 076
                 TOL=X                                                         RTNI 077
                 CALL FCT(TOL,F,DERF)                                          RTNI 078
       C                                                                       RTNI 079
       C         TEST ON SATISFACTORY ACCURACY                                 RTNI 080
                 TOL=EPS                                                       RTNI 081
                 A=ABS(X)                                                      RTNI 082
                 IF(A-1.)4,4,3                                                 RTNI 083
               3 TOL=TOL*A                                                     RTNI 084
               4 IF(ABS(DX)-TOL)5,5,6                                          RTNI 085
               5 IF(ABS(F)-TOLF)7,7,6                                          RTNI 086
               6 CONTINUE                                                      RTNI 087
       C         END OF ITERATION LOOP                                         RTNI 088
       C                                                                       RTNI 089
       C                                                                       RTNI 090
       C         NO CONVERGENCE AFTER IEND ITERATION STEPS, ERROR RETURN,      RTNI 091
                 IER=1                                                         RTNI 092
               7 RETURN                                                        RTNI 093
       C                                                                       RTNI 094
       C         ERROR RETURN IN CASE OF ZERO DIVISOR                          RTNI 095
               8 IER=2                                                         RTNI 096
                 RETURN                                                        RTNI 097
                 END                                                           RTNI 098
```

```
• ELT SI,1,720517, 56380 , 1

           SUBROUTINE SI  (PT,A    ,EY,SIF,SIS,OFS,NS)                       SI10
         DIMENSION EY(200),SIF(200),SIS(200),OFS(200),SIFD(2),SISD(2)
C  CALCULATE FROZEN AND SHIFTING SPECIFIC IMPULSE AT EACH STREAMLINE        S130
C  PT = TOTAL PRESSURE PSIA                                                  S140
C  A      = AREA EXIT / AREA THROAT                                          S150
C  EY = TOTAL ENTHALPY OF EACH STREAMLINE = BTU/LB                           S160
C  SIF= FROZEN SPECIFIC IMPULSE - SEC                                        S170
C  SIS= SHIFTING SPECIFIC IMPULSE -SEC                                       S180
C  OFS= MIXTURE RATIO OF EACH STREAMLINE                                     S190
  201 FORMAT(I3)
  200 FORMAT(*E15.8)
         WRITE(6,201) NS
         WRITE(6,200) (EY(I),I=1,200)
         WRITE(6,200) (SIF(I),I=1,200)
         WRITE(6,200) (SIS(I),I=1,200)
C     TRY TO FORCE I TO BE RESTORED
         WRITE(6,200) (OFS(I),I=1,200)
         WRITE(6,200) PT,A
           C=ALCG10(PT)
           D=ALCG10(A)
         I=0
  300 I=I+1
C  CALCULATE ISP FOR NS STREAMLINES                                          S1110
           OF1= AINT(OFS(I))                                                 S1120
         WRITE(6,201) I
         WRITE(6,200) C,D,OF1
           II=1                                                              S1130
    9      IF(CF1.NE.0) GO TO 11                                             S1140
C SPECIFIC IMPULSE FOR OF=0                                                  S1150
         SIFD(II)=262.4+(1.227*A)+(.0617*EY(I))+((4.436E-5)*A*EY(I))
        1-(.0234*A**2)-((5.139E-6)*EY(I)**2)+(.000135*A**3)+(2.454E-9)
        1*EY(I)**3
           SISD(II)=SIFD(II)                                                 S1180
           GC TC 21                                                          S1181
   11      IN=OF1                                                            S1190
         WRITE(6,201) I
         WRITE(6,201) II
         WRITE(6,201) IN
  115      IF(OF1.GT.7.) GO TO 60
           GC TC (12,13,14,15,16,17,18,19,20),IN                             S1200
   12    SISD(II)=366.7-5.5285*C+52.11*D+.0994*EY(I)+2.0273*C*D-.00178*D*
        1  EY(I)-.0018537*C*EY(I)+.81036*C**2-9.337*D**2+7.341E-7*EY(I)**2
           SIFD(II)=SISD(II)                                                 S1240
           GC TC 21                                                          S1250
   13    SISD(II)=372.7+.02725*C+79.49*D+.0674*EY(I)-.0364 *C*D+.00795*D*
        1  EY(I)+9.17E-5 *C*EY(I)+.02166*C**2-15.53*D**2-9.908E-6*EY(I)**2
         SIFD(II)=360.1+1.9402*C+81.08*D+.0616*EY(I)-.10722*C*D+.00849*D*
        1  EY(I)+.0012153*C*EY(I)-.22384*C**2-16.03*D**2-1.203E-5*EY(I)**2
           GC TC 21                                                          ST310
   14    SISD(II)=370.3+1.249 *C+88.65*D+.0551*EY(I)-.29645*C*D+.01069*D*
        1  EY(I)+.0005636*C*EY(I)-.07656*C**2-16.31*D**2-7.914E-6*EY(I)**2
         SIFD(II)=350.8+10.655*C+86.28*D+.0304*EY(I)+.67400*C*D+.00791*D*
        1  EY(I)+.007326 *C*EY(I)-1.2799*C**2-16.83*D**2-1.738E-5*EY(I)**2
           GC TC 21                                                          ST370
   15    SISD(II)=352.7+2.6874*C+104.1*D+.0459*EY(I)-.57293*C*D+.01195*D*
        1  EY(I)+.0009410*C*EY(I)-.05402*C**2-19.18*D**2-9.353E-6*EY(I)**2
         SIFD(II)=326.4+16.810*C+86.84*D+.0110*EY(I)+1.788 *C*D+.00641*D*
        1  EY(I)+.0103383*C*EY(I)-1.7611*C**2-17.04*D**2-1.648E-5*EY(I)**2
```

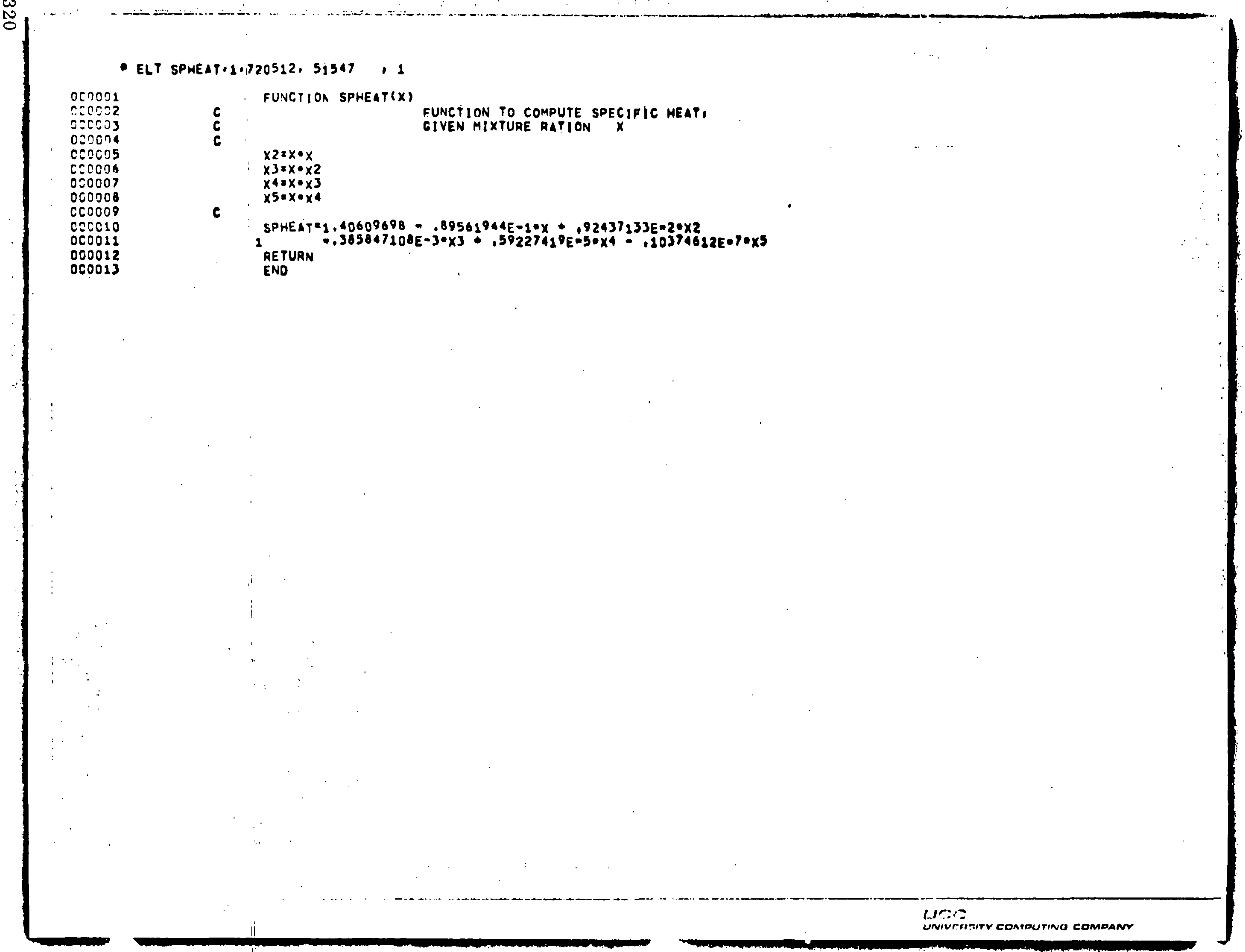
```
      WRITE(6,200) (SISD(II),SIFD(II))
         GO TO 21                                                        S1460
16       SISD(II)=325.9+9.2864*C+114.9*D+.0275*EY(I)-2.1312*C*D+.01476*D*
       1 EY(I)+.0035000*C*EY(I)-.06010*C**2-19.48*D**2-1.149E-5*EY(I)**2
         SIFD(II)=309.2+15.59 *C+85.36*D+.0135*EY(I)+2.6254*C*D+.00394*D*
       1 EY(I)+.0078631*C*EY(I)-.96139*C**2-16.84*D**2-8.546E-6*EY(I)**2
      WRITE(6,200) (SISD(II),SIFD(II))
         GO TO 21                                                        S1550
17       SISD(II)=299.8+13.498*C+122.2*D+.0127*EY(I)-2.8110*C*D+.01693*D*
       1 EY(I)+.0054181*C*EY(I)-.74417*C**2-19.29*D**2-1.156E-5*EY(I)**2
         SIFD(II)=293.8+16.056*C+81.87*D+.0149*EY(I)+3.5827*C*D+.00170*D*
       1 EY(I)+.0059334*C*EY(I)-1.0177*C**2-16.55*D**2-5.284E-6*EY(I)**2
         GO TO 21                                                        S1630
18       SISD(II)=277.8+16.664*C+123.1*D+.0034*EY(I)-2.6984*C*D+.01678*D*
       1 EY(I)+.006202 *C*EY(I)-.96994*C**2-18.13*D**2-1.155E-5*EY(I)**2
         SIFD(II)=283.9+13.471*C+77.62*D+.0135*EY(I)+3.7087*C*D+.00129*D*
       1 EY(I)+.0057789*C*EY(I)-.36765*C**2-15.64*D**2-5.778E-6*EY(I)**2
         GO TO 21                                                        S1700
19       SISD(II)=266.8+15.623*C+117.9*D+.0091*EY(I)-1.5026*C*D+.01336*D*
       1 EY(I)+.005327 *C*EY(I)-.9825 *C**2-17.03*D**2-7.546E-6*EY(I)**2
         SIFD(II)=274.5+11.734*C+74.78*D+.0156*EY(I)+3.8499*C*D+.00145*D*
       1 EY(I)+.0051689*C*EY(I)-.10738*C**2-15.10*D**2-4.772E-6*EY(I)**2
         GO TO 21                                                        S1770
20       SISD(II)=260.8+14.710*C+111.1*D+.0050*EY(I)-1.8082*C*D+.01622*D*
       1 EY(I)+.0059971*C*EY(I)-.85122*C**2-15.61*D**2-1.197E-5*EY(I)**2
         SIFD(II)=261.9+12.172*C+75.69*D+.0124*EY(I)+3.1568*C*D+.00185*D*
       1 EY(I)+.0059582*C*EY(I)-.18670*C**2-15.21*D**2-8.521E-6*EY(I)**2
21         II=II+1                                                       S1850
      WRITE(6,201) I
           IF(II.GT.2) GO TO 50                                          S1860
           OF1=AINT (OFS(I)+1)                                           S1870
      WRITE(6,200) OF1
           GO TO 11                                                      S1880
50         UP=OFS(I)-AINT(OFS(I))                                        S1890
           SIS(I)=SISD(1)+UP*(SISD(2)-SISD(1))                           S1891
           SIF(I)=SIFD(1)+UP*(SIFD(2)-SIFD(1))                           S1892
      WRITE(6,201) I
      WRITE(6,200) (UP,SIS(I),SIF(I))
10    IF (I.LT.NS)  GO TO 300
      WRITE(6,201) I
      WRITE(6,200) (EY(I),I=1,200)
      WRITE(6,200) (SIF(I),I=1,200)
      WRITE(6,200) (SIS(I),I=1,200)
      WRITE(6,200) (OFS(I),I=1,200)
      WRITE(6,200) PT,A
           RETURN                                                        S1970
60         OF1=9.                                                        S1900
           GO TO 11
           END                                                           S1960
```

```
• ELT SIDEL,1,720512, 51630     , 1

              SUBROUTINE  SIDEL(SIF,SIS,PT,OFS,NS,ISPK,SIK)            SID 10
             DIMENSION SIF(200),SIS(200),OFS(200),ISPK(6),SIK(200),SIKD(3)
              REAL ISPK
         C PROGRAM TO CALCULATE THE THEORETICAL SHIFTING ISP + KINETIC ISP    SID 30
         C OF EACH STREAMLINE - LINEAR INTERPOLATION USED TO PREDICT ISP      SID 40
         C CALCULATE KINETIC RECOMB PERCENT AT O/F=4,6,8                      SID 50
              IF(PT.GE.500.) GO TO 110
              AA=.69897
              P1=100.
              GO TO 120
      110     AA=.30103
              P1=500.
      120     CONTINUE
              DO 20 I=1,3                                                     SID 60
       20 SIKD(I)=ISPK(I)+((ISPK(I+3)-ISPK(I))/AA)*(ALOG10(PT)-ALOG10(P1))   SID 70
              DO 10 I=1,NS                                                    SID 90
              IF(OFS(I).LE.2.) GO TO 30
              IF(OFS(I).LE.4.) GO TO 40
              IF(OFS(I).LE.6.) GO TO 50
              IF(OFS(I).LE.8.) GO TO 60
              IF(OFS(I).GE.8.) GO TO 70
       30     SIK(I)=1.0                                                      SID 150
              GO TO 100                                                       SID 160
       40     SIK(I)=     1.+((OFS(I)-2.)/2.)*(SIKD(1)-1.0)                    SID 170
              GO TO 100                                                       SID 180
       50     SIK(I)= SIKD(1)+((OFS(I)-4.)/2.)*(SIKD(2)-SIKD(1))               SID 190
              GO TO 100                                                       SID 200
       60     SIK(I)=SIKD(2)+((OFS(I)-6.)/2.)*(SIKD(3)-SIKD(2))                SID 210
              GO TO 100                                                       SID 220
       70     SIK(I)=SIKD(3)                                                  SID 230
      100     CONTINUE                                                        SID 240
       10     CONTINUE                                                        SID 250
              DO 80 I=1,NS                                                    SID 260
              IF(SIK(I).LT.0) SIK(I)=0
       80 SIK(I)=SIF(I)+(SIK(I)*(SIS(I)-SIF(I)))
              RETURN                                                          SID 280
              END                                                             SID 290
```

```
 ● ELT SLOT,1,720512, 51577      , 1

      SUBROUTINE SLOT
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLW(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TWX(7,4)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISRATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,NINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
     1OR,USTOR,RAY,RG,AK,AKA
      COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      COMMON/ZL/QLX(7,4),           TCX(7,4),QL,TC,PRB,GAMW,RUC,RUCX(7,4)
     1,SCB,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
     2,GG,UB(21),UBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
     3,VWS(21),RWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,GGG(21)
     4,CPB,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
     5,NCL,NP,FYE,AMP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),HC,HCS(21)
      COMMON/STORNP/RRUSE,NUSE,RPRB,RRUT,RALAW(9),RHEE,RHAW,RUE,RRHO,RCP
     1,RVB,RVS
      COMMON/ZB/OF(200),HSEN(200)
      COMMON/ZF/SPL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL,UTBL,CPBL,HEBL
     1,RHOBL,FLTBL,WMBL,TBL,GMBL,SSBL,EMBL,PTBL,TTBL
    1 FORMAT(8E15.7)
C     SLOT COOLING
      IF(NUSE.GT.0)GO TO 100
      NSLCR=1
      HTCG=AST**.9 * .026/DST**.2 * SIGMA(1)**(-.6) *(DST/RSTAR)**.1
      EPZ=.001
      IWTLM=10
      DO 101 K=1,NSLOT
  101 TWS(K)=T(LR)
  100 CONTINUE
C     CHOOSE BULK CONDITIONS AS FOR TRANSPIRATION
      HAW=HEBL+PRT*UBL*UBL
      TAW=TBL
      DUM=.0
      DO 120 J=1,NSPC
  126 ALS(J)=YSPEC(J,LR)
      DUM=DUM+ALS(J)/WTMOLE(J)
  120 ALAW(J)=SBL(J)
      WW=1./DUM
      CALL INVERT(TAW,HAW,ALAW,CPAW)
      DO 121 K=NSLCR,NSLOT
      IF(X.GT.XS(K))GO TO 121
      NSLCR=K-1
      GO TO 122
  121 CONTINUE
      NSLCR=NSLOT
  122 CONTINUF
      CALL VASC(VS,TBL,ALAW)
      IF(NUSE.EQ.0)GO TO 250
  252 CONTINUE
```

```
      IF(MY.GT.0)WRITE(6,4)
    4 FORMAT(/40X,9HFROM SLOT)
      IF(MY.GT.0)WRITE(6,3)IOUT,NUSE,NSLCR,HTCG,HAW,TAW,WW,RUE,RRUT,VS
    3 FORMAT(3I5,7E15.7)
      DO 123 K=1,NSLCR
      UB(K)=RLE/UC(K)
      FFF(K)=RUCF(K)/RRUT
      UGA=UB(K)**.125
      IF(UB(K).GT.1.)GO TO 124
      UGB=UB(K)-1.
      SF(K)=UGA*(1.+.4*ATAN(UGB))
      GO TO 125
  124 CONTINUE
      UGC=1.5*(1.-1./UB(K))
      SF(K)=UGA*UB(K)**UGC
  125 CONTINUE
      IF(MY.GT.0)WRITE(6,3)IOUT,NSLOT,K,UC(K),UB(K),RUCF(K),FFF(K),UGA
     1,UGC,SF(K)
  123 CONTINUE
      ANW=YW*(PYE*YW)**NP
      EMW=PSIW*(PYE*PSIW)**NP
      CS=P*AST/EMW
      IF(MY.GT.0)WRITE(6,3)IOUT,NP,IWTLM,ANW,EMW,CS,RRHO,RCP,RVS
     1,TWS(1)
      DO 130 K=1,NSLCR
      CALL HOCT(TCS(K),HCS(K),ALC,CPC)
      CALL VASC(VC,TCS(K),ALC)
      IW=0
  140 IW=IW+1
      CALL VASC(VWS(K),TWS(K),ALS)
      RWS(K)=P*WW/RG/TWS(K)
      SG(K)=(.25*(1.+RWS(K)/RRHO)*(1.+VWS(K)/RVS))**.8
      IF(MY.GT.0)WRITE(6,3)K,IW,KM,HCS(K),VC,VWS(K),TCS(K),TWS(K),RWS(K)
     1,SG(K)
      HTC(K)=HTCG*SG(K)*ANW**(-.9)*RVS**.2*RCP*(P/CS)**.8
      ST(K)=HTC(K)/RRUT/RCP
      GGG(K)=CPC*FFF(K)/RCP/ST(K)
      RHC=P*WTMOLE(NCL)/RG/TCS(K)
      RYNC(K)=RHC*UC(K)*SH(K)/VC
      GRPZ=(X-XS(K))/GGG(K)/SH(K)
      ETALG=(RYNC(K)*PRC)**.125*SF(K)*(XK(4)-GRPZ)
      ETA(K)=EXP(ETALG)
      KM=K-1
      IF(MY.GT.0)WRITE(6,3)K,IW,KM,HTC(K),ST(K),GGG(K),RHC,RYNC(K),ETALG
     1,ETA(K)
      IF(ETA(K).GT.1.)ETA(K)=1.
      IF(K.EQ.1)GO TO 131
      TWS(K)=TWS(KM)+ETA(K)*(TCS(K)-TWS(KM))
      IF(IW.LE.1)TWB=TWS(KM)
      GO TO 132
  131 TWS(1)=TAW+ETA(1)*(TCS(1)-TAW)
      IF(IW.LE.1)TWB=TCS(1)
  132 CONTINUE
      EPW=ABS(TWS(K)/TWB-1.)
      IF(MY.GT.0)WRITE(6,3)IOUT,K,IW,EPW,TWB,TWS(K)
     1,GRPZ
      IF(IW.GT.IWTLM)GO TO 134
      IF(EPW.LE.EPZ)GO TO 130
      GO TO 133
  134 CONTINUE
```

```
      WRITE(6,2)ICUT,ISOBAT,K,TWS(K),TWB,ETA(K),RYNC(K),RHC,HTC(K),SG(K)
    2 FORMAT(40X,36HTOO MANY WALL TEMPERATURE ITERATIONS/3I5,7E15.7)
      GO TO 130
  133 CONTINUE
      TWB=TWS(K)
      GO TO 140
  130 CONTINUE
      TW=TWS(NSLCR)
      NUSE=NUSE+1
  250 CONTINUF
      RRUT=RUTBL
      RUE=UBL
      RRHO=RHOBL
      RCP=CPBL
      RVS=VS
      DO 251 J=1,NSPC
      RALAW(J)=ALAW(J)
  251 CONTINUE
      IF(NUSE.EQ.0)GO TO 252
      RETURN
      END
```



```
@ ELT SPHEAT,1,720512, 51547   , 1
                  FUNCTION SPHEAT(X)
         C                          FUNCTION TO COMPUTE SPECIFIC HEAT,
         C                          GIVEN MIXTURE RATION   X
         C
                  X2=X*X
                  X3=X*X2
                  X4=X*X3
                  X5=X*X4
         C
                  SPHEAT=1.40609698 - .89561944E-1*X + .92437133E-2*X2
                 1      -.385847108E-3*X3 + .59227419E-5*X4 - .10374612E-7*X5
                  RETURN
                  END
```

UCC
UNIVERSITY COMPUTING COMPANY

```
● ELT STEP,1,720512, 51581      , 1

      SUBROUTINE STEP
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEF(200)
      COMMON/EXTRA/JAM(17),DAMP(17),NREAC,THLD(200),HHLD(200),FIX(200)
     1,IDA(200),TCHEM(200),WDT(9,200)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,WST
     1OR,USTOR,RAV,RG,AK,AKA
      COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
      IF(MPSI.NE.1)GO TO 300
      IF(ICHEM.NE.3) GO TO 20
      DX=FIX(1)*U(1)
      GO TC 301
   20 IF(ICUT.GT.1)GO TO 21
      DX=.1*XMAX/XMPS
      YW=Y(1)
      DLRUT=RLT(1)
      DX=AMIN1(DX,YW)
   21 DLU=ABS(DX*DPDX/RUT(1))
      IF(DLU.GT.U(1))DX=DX*U(1)/DLU
      DEAL=ABS(RUT(1)/DLRUT-1.)
      IF(DEAL.GT..01)DX=DX*.01/DEAL
      DLRUT=RLT(1)
      GO TC 301
  300 CONTINUE
      DLSSCH=SIGMA(1)/XLE(1)*DELPSI**2
      IF(NTYPE.NE.0)GO TO 71
      XD=DLSSCH/XMU(1)/12.
      DO 10 I=2,NPSI
      DIVIS=A(I+1)+A(I-1)+A(I)+A(I)
      DELX=PSI(I)*DLSSCH/DIVIS/1.5
   10 XD=AMIN1(XD,DELX)
      GO TC 64
   71 XD=DLSSCH/XMU(1)/RUT(1)/6.
      DO 73 I=2,NPSI
      DIVIS=A(I+1)+A(I-1)+2.*A(I)
      DELX=DLSSCH/DIVIS/1.5
   73 XD=AMIN1(XD,DELX)
   64 DO 60 I=2,LR
      YD=Y(I)-Y(I-1)
   60 XD=AMIN1(XD,YD)
      DX=XD/XMPS
  301 CONTINUE
      IF(DX.GT..0)RETURN
      IFINIS=2
      WRITE(6,9)
    9 FORMAT(50X,28HFAIL WITH NEGATIVE STEP SIZE)
      WRITE(6,1)DX,X,XD,XMPS
    1 FORMAT(8E15.7)
      RETURN
      END
```

```
• ELT TGR,1,720512, 51545      , 1
          FUNCTION TGR(P,X)
C                            FUNCTION TO COMPUTE GAS TEMP IN DEG RANKIN
C                              GIVEN=
C                                1. CHAMBER PRESSURE    P
C                                2. OX MIXTURE RATIO    X
C
          DATA A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13
         1 /435.7715,2113.65,-286.074,10.44714,.243838,.691573,.869524,
         2 -.003725,3.017357E-6,-.927035E-3,.235999E-1,1.026078E-7,
         3 -1.004847E-5/
          DATA B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13
         1 /5068.58864,123.52323,-13.85322,.4943759,-.6373588E-2,-.1577078,
         2 8.8357561,-.16957177E-1,9.2102784E-6,.21736366E-3,-.10591539E-2,
         3 -5.3800C3437E-9,1.223841156E-7/
C
          X2=X*X
          X3=X*X2
          X4=X*X3
          P2=P*P
          P3=P*P2
          P4=P*P3
          IF(X .GT. 8.0)GO TO 1
C
          TGR= A1 + A2*X + A3*X2 + A4*X3 + A5*X4 + A6*X*P + A7*P + A8*P2 +
         1 A9*P3 + A10*X*P2 + A11*X2*P + A12*P3*X2 + A13*P2*X3
          RETURN
C
        1 TGR= B1 + B2*X + B3*X2 + B4*X3 + B5*X4 + B6*X*P + B7*P + B8*P2 +
         1 B9*P3 + B10*X*P2 + B11*X2*P + B12*P3*X2 + B13*P2*X3
          RETURN
          END
```

```
• ELT TRANSP,1,720512, 51574   , 1

      SUBROUTINE TRANSP
      COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(200)
      COMMON/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TPX(7,4)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,NHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NFTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
     1OR,USTOR,RAY,RG,AK,AKA
      COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
      COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      COMMON/ZL/QLX(7,4),            TCX(7,4),QL,TC,PRR,GAMW,RUC,RUCX(7,4)
     1,SCB,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
     2,GG,UB(21),UBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
     3,VWS(21),RWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,GGG(21)
     4,CPB,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
     5,NCL,NP,PYE,ANP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),HC,HCS(21)
      COMMON/STORNP/RRUSE,NUSE,RPRB,RRUT,RALAW(9),RHEE,RHAW,RUE,RRHO,RCP
     1,RVB,RVS
      COMMON/ZB/OF(200),HSEN(200)
      COMMON/ZF/SPL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL,UTBL,CPBL,HEBL
     1,RHOBL,RUTBL,WMBL,TBL,GMBL,SSBL,EMBL,PTBL,TTBL
    1 FORMAT(8E15.7)
      IF(NUSE.GT.0)GO TO 300
      EPT=.01
      PRW=.0592/SIGMA(1)**.6667
      EPZ=.001
      ITWLM=10
  300 CONTINUE
      CALL PRESS(TC,DTCDX,X,TCX,XP,LY)
      CALL HOCT(TC,HC,ALC,CPC)
      IF(ISBATY.GT.2)GO TO 25
      CALL PRESS(RUC,DRUCDX,X,RUCX,XP,LY)
      GO TO 26
   25 CALL PRESS(TW,DTWDX,X,TWX,XP,LY)
   26 CONTINUE
      IF(LZ.GT.0)WRITE(6,3)IOUT,NUSE,ISBATY,TC,CPC,HC,TW,RUC
    3 FORMAT(40X,11HFROM TRANSP/3I5,7E15.7)
      GO TO(20,50,20,50),ISBATY
C     COOLANT FLOW EDGE CONDITIONS ARE USED INSTEAD OF BULK CONDITIONS
   20 CONTINUE
      NED=LR-1
      IF(YSPEC(NNT,LR).LE..0)GO TO 22
      DO 21 J=2,LR
      K=LR-J+1
      IF(YSPEC(NNT,K).LT.YSPEC(NNT,LR)*EPT)GO TO 22
      NED=K
   21 CONTINUE
   22 CONTINUE
      HAW=SMALLH(NED)+PRT*U(NED)*U(NED)
      DO 51 J=1,NSPC
   51 ALAW(J)=YSPEC(J,NED)
```

```
      RUTX=RUT(NED)
      UX=U(NED)
      RHOX=RHO(NED)
      CPX=CPBAR(NED)
      HEX=SMALLH(NED)
      TAW=T(NED)
      GO TO 150
   50 CONTINUE
      HAW=HEBL+PRT*UBL*UBL
      DO 151 J=1,NSPC
  151 ALAW(J)=SBL(J)
      RUTX=RUTBL
      UX=UBL
      RHOX=RHOBL
      CPX=CPBL
      HEX=HEBL
      TAW=TBL
  150 CONTINUE
      CALL INVERT(TAW,HAW,ALAW,CPAW)
      IF(NUSE.EQ.0)GO TO 250
  252 CONTINUE
      IF(LZ.GT.0)WRITE(6,4)NED,ITWLM,NCL,RRUT,RUE,RRHO,RCP,RHEE,RHAW,
     1RAWNCL
    4 FORMAT(3I5,7E15.7)
      FF=RUC/RRUT
      DUM=.0
      DO 100 J=1,NSPC
  100 DUM=DUM+YSPEC(J,LR)/WTMOLE(J)
      WW=1./DUM
      DUM=.0
      BUM=.0
      DO 52 J=1,NSPC
      BUM=BUM+RALAW(J)/WTMOLE(J)
      IF(J.EQ.NCL)GO TO 52
      DUM=DUM+RALAW(J)/WTMOLE(J)
   52 CONTINUE
      WE=1./BUM
      WEB=RAWNCL/DUM
      WM=WE/WW
      RCW=1.-YSPEC(NCL,LR)
      IF(LZ.GT.0)WRITE(6,4)IOUT,NED,LR,FF,WW,WE,WM,WEB,RCW
      IF(RCW.GE.1.)GO TO 79
      IF(RAWNCL.LT.1.)GO TO 62
      ALSB=1.
      GO TO 61
   79 ALSB=.0
      GO TO 61
   62 CONTINUE
      ALSB=WEE/(WEB-WTMOLE(NCL))
      ALSB=ALSB*ALOG(WM)/ALOG(WM*RAWNCL/RCW)
      IF((ALSB.GE..0).AND.(ALSB.LE.1.))GO TO 61
      WRITE(6,7)ICUT,NED,LR,ALSB,WEB,WE,WW,RAWNCL,YSPEC(NCL,LR)
    7 FORMAT(40X,26HALSB IS IMPROPER IN TRANSP,3I5,7E15.7)
      IF(LZ.LT.1)LZ=1
   61 CONTINUF
      ALSG=ALSB/RAWNCL
      ALS(NCL)=1.-ALSB
      DUM=.0
      DO 53 J=1,NSPC
      IF(J.EQ.NCL)GO TO 54
      ALS(J)=RALAW(J)*ALSG
   54 DUM=DUM+ALS(J)/WTMOLE(J)
   53 CONTINUE
      WS=1./DUM
      IF(ISBATY.GT.2)T(LR)=TW
C     ITERATE FOR TW(RX)
      ITW=0
      TWB=T(LR)
      TSTAR=TAW
      IF(LZ.GT.0)WRITE(6,1)FF,WW,WEB,ALSB,ALSG,WS,TWB,TSTAR
   58 ITW=ITW+1
      KOPT=1
      CALL HEAT(LR,LR)
      HSTAR=.5*(H(LR)+RHEE)+.22*(RHAW-RHEE)
      CALL INVERT(TSTAR,HSTAR,ALS,CPSTAR)
      CALL VASC(VSTAR,TSTAR,ALS)
      RHST=P*WS/RG/TSTAR
      RYNS=RHST*RUE/VSTAR*X
      IF((LZ.LT.1).AND.(RYNS.LT..0))GO TO 69
      GO TO 63
   69 LZ=1
      WRITE(6,6)
    6 FORMAT(40X,37HTRANSP IS BLOWING UP WITH NEGATIVE RE)
      GO TO 300
   63 CONTINUE
      IF(RYNS.GT..0)GO TO 163
      STAN=1.
  163 CONTINUE
      STAN=PRW*RHST/RRHO/RYNS**.2
      IF(STAN.GT.1.)STAN=1.
      IF(LZ.GT.0)WRITE(6,4)ITW,LR,MPSI,HSTAR,CPSTAR,VSTAR,RHST,RYNS,STAN
     1,TWB
      IF(ISBATY.GT.2)GO TO 60
C     (RHO-U)C AND TC ARE SPECIFIED - FIND TW
      GG=CPC*FF/RCP/STAN
      TWB  =TAW+(TC-TAW)*(1.-(1.+GG/3.)**(-3.))
      IF(ITW.LE.1)GO TO 56
      EPW=ABS(TWB/TWA-1.)
      IF(EPW.LE.EPZ)GO TO 59
      IF(ITW.GT.ITWLM)GO TO 57
   56 TWA=TWB
      T(LR)=TWB
      GO TO 58
   57 CONTINUF
      WRITE(6,2)ICUT,ISOBAT,ISBATY,TWB,TWA,GG,STAN,RYNS,RHST,TSTAR
    2 FORMAT(40X,36HFROM TRANSP - TOO MANY TW ITERATIONS/3I5,7E15.7)
   59 T(LR)=TWB
      TW=TWB
      GO TO 200
C     TW AND TC ARE GIVEN - FIND COOLANT MASS FLUX
   60 CONTINUF
      GG=3.*((1.-(TAW-TW)/(TAW-TC))**(-1./3.)-1.)
      FF=GG*RCP/CPC*STAN
      RUC=FF*RRUT
  200 CONTINUE
      NUSE=NUSE+1
      SCW=2.*CELPSI*SCB*GAMW*RUC
      IF(LZ.GT.0)WRITE(6,1)FF,GG,RUC,TWA,TWB,TW,SCW
      IF(ICHEM.EQ.2)GO TO 80
C     EVALUATE SPECIE WALL B.C.
```

```
      CO 70 J=1,NSPC
      IF(J.EQ.NNT)GO TO 71
      IF(J.EQ.NCL)GO TO 72
      YSPEC(J,MPSI)=YSPEC(J,JR)-SCW*YSPEC(J,LR)
      GO TO 7C
   71 YSPEC(NNT,MPSI)=YSPEC(NNT,JR)+SCW*(ALC(NNT)-YSPEC(NNT,LR))
      GO TC 7C
   72 YSPEC(NCL,MPSI)=YSPEC(NCL,JR)+SCW*(1.-YSPEC(NCL,LR))
   70 CONTINUE
      GO TC 250
C     EVALUATE ELEMENT WALL B.C.
   80 CONTINUE
      IF((SCW.LE.1.).AND.(SCW.GE..0))GO TO 81
      WRITE(6,3)ICUT,NUSE,ISBATY,SCW,RUC,TC,ELC(LN),TW,TWB,TWA
      WRITE(6,5)
    5 FORMAT(40X,43HCOOLANT MASS FLUX IS TOO BIG OR IS NEGATIVE)
      IF(SCW.GT.1.)SCW=1.
      IF(SCW.LT.0.)SCW=0.
   81 CONTINUE
      ALPHA(LH,MPSI)=ALPHA(LH,JR)+SCW*(1.-ALPHA(LH,LR))
      ALPHA(LC,MPSI)=ALPHA(LO,JR)+SCW*(-ALPHA(LO,LR))
      ALPHA(LN,MPSI)=ALPHA(LN,JR)+SCW*(ELC(LN)-ALPHA(LN,LR))
  250 CONTINUE
      RRUT=RUTX
      RUE=UX
      RRHO=RHCX
      RCP=CPX
      RHEE=HEX
      RHAW=HAW
      DO 251 J=1,NSPC
      RALAW(J)=ALAW(J)
  251 CONTINUE
      RAWNCL=1.-RALAW(NCL)
      IF(NUSE.EQ.0)GO TO 252
  400 RETURN
      END
```

```
● ELT VASC,1,720512, 51579      , 1

      SUBROUTINE VASC(XMU,TT,ALP)
      COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
     1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISHATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NNT,NCT,NHTW,LUV,MP,IEODAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      DIMENSION ALP(9),AL(9)
    1 FORMAT(8E15.7)
      T=TT
      DO 20 J=1,NSPC
   20 AL(J)=ALP(J)
      CNV=47.87
      GO TO(30,40,50,60),LW
   30 CONTINUE
      XMUT=3.046E-8*CNV*T**1.5/(T+110.4)
      GO TO 98
   40 CONTINUE
      GO TO 98
   50 CONTINUE
      GO TO 98
   60 CONTINUE
   98 XMU=XMUT
      RETURN
      END
```

```
• ELT VISC,1,720512, 51566      , 1

      SUBRCUTINE VISC
      COMMCN/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
     1,W(9,2?C),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
     2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
     3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
     4,FEE(20C)
      COMMCN/ZC/WTMOLE(9),TITLE(12),CGP(7,4),XP(7),XK(7)
     1,GSCALE,TWX(7,4)
      COMMCN/ZC/NPSI,LR,IFINIS,ICHEM,ITURB,IPRESS,IOUT,IPAGE,MY,NTYPE,MP
     1SI,LS,LT,LU,LV,LW,LX,LY,LZ,NSPC,MA,MB,MC,MD,ME,MF,MG,MH
     2,ISBATY,MJ,MK,ML,MM,MN,MO,NSLOT
     3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
     4,NEW,MQ,MR,LN,NNT,JR,NSLCR
      COMMCN/ZE/X,XMAX,F,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DPDX,XTRA,HST
     1CR,USTCH,RAY,RG,AK,AKA
      COMMCN/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
      COMMCN/ZF/SPL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL;UTBL,CPBL,HEBL
     1,RHOBL,FUTBL,WMBL,TBL,GMBL,SSBL,EMBL,PTSL,TTBL
    1 FORMAT(PE15.7)
C     FROM SLUG/FT/SEC TO KG/M-SEC
      CNV=47.87
   32 GO TC(2C,75,76,77,78,79),ITURB
   20 CONTINUE
C     ITURB=1 - HIRSCH ROCKET CHAMBER MODEL
C     XK(1)=ALPHA  XK2=K  XK(3)=S
      XMUT=XK(1)*XK2*2.*Y(MPSI)*RUTBL/(1.+X/XK(3))
      GO TC 98
C     ITURB=2 - VISCOSITY IS AN INPUTTED CONSTANT
   75 XMUT=XK2
  271 GO TC 98
   76 CONTINUE
      GO TC 98
   77 CONTINUE
      GO TC 98
   78 CONTINUE
      GO TO 98
   79 CONTINUE
   98 DO 36 I=1,MR
   36 XMU(I)=XMUT
      RETURN
      END
  @ ELT WALL,1,720512, 51569      , 1
         SUBROUTINE WALL
         COMMON/ZA/ALPHA(3,200),RALPHA(3,200),YSPEC(9,200),RYSPEC(9,200)
        1,W(9,200),SIGMA(1),XLE(1),RU(200),CPBAR(200),XML(200),U(200)
        2,A(200),RHO(200),Y(200),PSI(200),T(200),RH(200),SMALLH(200),H(200)
        3,WTMIX(200),RT(200),TAUT(200),RUT(200),TELAP(200),EMDT(200)
        4,FEE(200)
         COMMON/ZC/WTMOLE(9),TITLE(12),CCP(7,4),XP(7),XK(7)
        1,SSCALE,TWX(7,4)
         COMMON/ZD/NPSI,MPSI,IFINIS,ICHEM,ITURB,IPRESS,ICUT,IPAGE,MY,NTYPE,
        1LR,LS,LT,LU,LV,LW,LX,LY,LZ,NSPE,MA,MB,MC,MD,ME,MF,MG,MH
        2,ISHATY,MJ,MK,ML,MM,MN,MO,NSLOT
        3,MINIT,MHALF,NGAS,KOPT,NEL,LO,LH,NHTO,NHT,NOT,NHTW,LUV,MP,ISOBAT
        4,NEW,MQ,MR,LN,NNT,JR,NSLCR
         COMMON/ZE/X,XMAX,P,XMUT,DELPSI,DX,XMPS,PRNT,PCNT,XK2,DFDX,XTRA,HST
        1OR,USTOR,RAY,RG,AK,AKA
         COMMON/ZH/DYWDX,YW,PNEW,PSIW,PSIWA,TAU,YWP
         COMMON/ZJ/GAM(3,9),GAN(27),HF(5,6,9),WTE(3),DEL(9),TW
         COMMON/ZL/QLX(7,4),            TCX(7,4),QL,TC,PRB,GAMW,RUC,RUCX(7,4)
        1,SCB,HAW,TAW,FF,WW,WEB,ALSB,ALS(9),WS,TS,VS,RHST,RYNS,STAN,CPC,CPE
        2,GG,UB(21),UBLK,UC(21),FFF(21),RUCF(21),RUBLK,SF(21),TWS(21)
        3,VWS(21),RWS(21),SG(21),CS,AST,DST,HTC(21),ST(21),RUCPB,GGG(21)
        4,CPB,RYNC(21),SH(21),XS(21),VC,PRC,ETA(21),RSTAR,TCS(21)
        5,NCL,NP,PYE,ANP,ALAW(9),XXX(9),NED,PRT,ALC(9),ELC(3),HC,HCS(21)
         COMMON/STORNP/RRUSE,NUSE,RPRB,RRUT,RALAW(9),RHEE,RHAW,RUE,RRHO,RCP
        1,RVB,RVS
         COMMON/ZB/OF(200),HSEN(200)
         COMMON/ZF/SBL(9),EBL(3),HBL,UBL,HSBL,FEEBL,OFBL,UTBL,CPBL,HEBL
        1,RHOBL,RUTBL,WMBL,TBL,GMBL,SSBL,EMBL,PTBL,TTBL
       1 FORMAT(8E15.7)
         IF(ICUT.GT.1)GO TO 30
         NUSE=0
         PRT=.5*SIGMA(1)**.3333
      30 CONTINUF
         IF(NEW.NE.0)GO TO 60
 C       SEE IF ARE AT THE WALL
         DEER=PSI(MPSI)-PSIW
         IF(DEER.LT..0)GO TO 33
         NEW=1
         IF((DELPSI/2.-DEER).LT..0)GO TO 22
         PSIW=PSI(MPSI)
         MPSI=MPSI+1
         GO TC 24
      22 PSIW=PSI(NPSI)
      24 NPSI=NPSI-1
         PCNT=PRNT
         WRITE(6,39)PSIW,T(NPSI)
      39 FORMAT(1H1,////50X,5HPSIW=E15.8//52X,3HTW=E15.8)
      60 LR=NPSI
         JR=NPSI-1
         NED=1
      50 CONTINUF
         GAMW=(PSI(LR)/Y(LR))**NP/RUT(LR)
         HAW=HEBL+PRT*UBL*UBL
         TAW=TBL
         CALLINVERT(TAW,HAW,SBL,CPAW)
         VB=XMU(NED)
         IF(NUSE.EQ.0)RVB=VB
         PRB=SIGMA(1)/RVB
      SCB=PRB/XLE(1)
      RUSE=RUTPL/XMU(1)*UBL
      IF(NUSE.EQ.0)RRUSE=RUSE
      TAU=XK(E)*RRUSE*GAMW
      U(MPSI)=U(JR)-2.*DELPSI*TAU
      IF(U(MPSI).LE..0)U(MPSI)=.1*U(LR)
   61 IF(ISOBAT.EQ.3)GO TO 200
   26 CONTINUE
      DO 122 J=1,NSPC
  122 YSPEC(J,MPSI)=YSPEC(J,JR)
      DO 21 J=1,NFL
      RALPHA(J,MPSI)=ALPHA(J,JR)
   21 ALPHA(J,MPSI)=ALPHA(J,JR)
      GO TO(27,28,27,300),ISOBAT
C     ISOENERGETIC,IMPERMEABLE WALL
   27 CONTINUE
      H(MPSI)=H(JR)
      RH(MPSI)=H(JR)
      KOPT=2
      CALL HEAT(MPSI,MPSI)
      IF(T(MPSI).LE..0)GO TO 20
      IF(ISOBAT.GT.1)GO TO 25
      NUSE=NUSE+1
      GO TO 25
C     REGENERATIVE COOLING - IMPERMEABLE WALL
   28 CONTINUE
      IF(NUSE.EQ.0)RPRB=PRB
      NUSE=NUSE+1
      IF(ISBATY.GT.1)GO TO 281
C     WALL TEMPERATURE SPECIFIED - COMPUTE WALL HEAT TRANSFER RATE
      CALL PRESS(TW,DTWDX,X,TWX,XP,LY)
      IF(TW.LE..0)GO TO 20
      T(NPSI)=TW
      T(MPSI)=T(LP)
      KOPT=1
      CALL HEAT(NPSI,MPSI)
      QL=(H(JR)-H(LR))/DELPSI/RPRB/GAMW
      GO TO 25
  281 CONTINUE
C     WALL HEAT TRANSF SPECIFIED - COMPUTE WALL TEMPERATURE
      CALL PRESS(QL,DQLDX,X,QLX,XP,LY)
      H(MPSI)=H(JR)-2.*DELPSI*QL*RPRB*GAMW
      KOPT=2
      CALL HEAT(MPSI,MPSI)
      IF(T(MPSI).LE..0)GO TO 20
      TW=T(MPSI)
      GO TO 25
C     TRANSPIRATION WALL COOLING
  200 CONTINUE
      CALL PRESS(TC,DTCDX,X,TCX,XP,LY)
      IF(TC.GE.T(NPSI))GO TO 26
      IF(ISBATY.LT.3)GO TO 201
      CALL PRESS(TW,DTWDX,X,TWX,XP,LY)
       IF(TW.GE.T(LR))GO TO 26
  201 CONTINUE
      CALL TRANSP
      T(LR)=TW
      T(MPSI)=T(LR)
      KOPT=1
      IF(ICHEM.EQ.2)CALL EQUILC(LR,MPSI)
      CALL HEAT(LR,MPSI)
      IF(T(LR).LE..0)GO TO 20
      GO TC 25
  300 CONTINUE
      IF(X.LT.XS(1))GO TO 301
      CALL SLCT
  301 CONTINUE
      H(MPSI)=H(JR)
      RH(MPSI)=H(JR)
      KOPT=2
      CALL HEAT(MPSI,MPSI)
      IF(T(MPSI).LE..0)GO TO 20
      GO TC 25
   20 IFINIS=2
      WRITE(6,4)
    4 FORMAT(////45X,      31HFAILURE DUE TO WALL B.C. = STOP)
      RETURN
   25 CALL DENSE(NPSI,MPSI)
      RRUSE=RLSE
      RPRB=PRB
      RVB=VB
      RETURN
   33 LR=MPSI
      RETURN
      END

 END CLR
```