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
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

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>FOREWORD</td>
<td>ii</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>ANALYSIS</td>
<td>3</td>
</tr>
<tr>
<td>Model Description</td>
<td>3</td>
</tr>
<tr>
<td>Dynamics</td>
<td>6</td>
</tr>
<tr>
<td>Injection</td>
<td>6</td>
</tr>
<tr>
<td>Ignition</td>
<td>8</td>
</tr>
<tr>
<td>Combustion</td>
<td>8</td>
</tr>
<tr>
<td>Performance</td>
<td>8</td>
</tr>
<tr>
<td>Heat Transfer</td>
<td>9</td>
</tr>
<tr>
<td>Analysis Methods</td>
<td>9</td>
</tr>
<tr>
<td>Dynamics</td>
<td>9</td>
</tr>
<tr>
<td>Injection</td>
<td>14</td>
</tr>
<tr>
<td>Ignition</td>
<td>17</td>
</tr>
<tr>
<td>Mixing and Combustion</td>
<td>17</td>
</tr>
<tr>
<td>Performance</td>
<td>40</td>
</tr>
<tr>
<td>Heat Transfer</td>
<td>52</td>
</tr>
<tr>
<td>SAMPLE CASE RESULTS</td>
<td>58</td>
</tr>
<tr>
<td>CONCLUSIONS AND RECOMMENDATIONS</td>
<td>67</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>68</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>69</td>
</tr>
<tr>
<td>APPENDIX A - PROGRAM INPUT</td>
<td>75</td>
</tr>
<tr>
<td>Input Forms</td>
<td>77</td>
</tr>
<tr>
<td>Input Parameter Descriptions</td>
<td>89</td>
</tr>
<tr>
<td>APPENDIX B - SAMPLE CASES</td>
<td>99</td>
</tr>
<tr>
<td>Combustion, Injection, &amp; Performance</td>
<td>100</td>
</tr>
<tr>
<td>Dynamics</td>
<td>111</td>
</tr>
<tr>
<td>Heat Transfer</td>
<td>114</td>
</tr>
<tr>
<td>APPENDIX C - PROGRAM SUBROUTINES</td>
<td>119</td>
</tr>
<tr>
<td>Subroutine Tree</td>
<td>120</td>
</tr>
<tr>
<td>Subroutine Descriptions</td>
<td>121</td>
</tr>
<tr>
<td>Subroutine Flow Charts</td>
<td>129</td>
</tr>
<tr>
<td>Subroutine Listings</td>
<td>199</td>
</tr>
<tr>
<td>FIGURE</td>
<td>TITLE</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>GH₂/GO₂ Reaction Control System - Schematic</td>
</tr>
<tr>
<td>2</td>
<td>GO₂/GH₂ Engine Dynamics Model</td>
</tr>
<tr>
<td>3</td>
<td>Subroutine DYNAM Variable Flow Area Model</td>
</tr>
<tr>
<td>4</td>
<td>Spark Breakdown Voltage Criteria</td>
</tr>
<tr>
<td>5</td>
<td>Minimum Ignition Energy</td>
</tr>
<tr>
<td>6</td>
<td>Flame Quenching Parameter</td>
</tr>
<tr>
<td>7</td>
<td>Model of Initial Conditions</td>
</tr>
<tr>
<td>8</td>
<td>Finite Difference Grid Network</td>
</tr>
<tr>
<td>9</td>
<td>Schematic of Multiple Slot Cooling Configuration</td>
</tr>
<tr>
<td>10</td>
<td>Axisymmetric Perfect Nozzle Contours</td>
</tr>
<tr>
<td>11</td>
<td>Nozzle Area Ratio Gradients of Selected Perfect Nozzle Contours</td>
</tr>
<tr>
<td>12</td>
<td>Normalized Graphical Solution for Freezing Area Ratio Using Modified Bray Analysis</td>
</tr>
<tr>
<td>13</td>
<td>Effect of Freezing Area Ratio on Nonequilibrium Performance for Hydrogen-Oxygen Propellant System</td>
</tr>
<tr>
<td>14</td>
<td>Effective Nozzle Wall Angle for Contoured Nozzle</td>
</tr>
<tr>
<td>15</td>
<td>Boundary Layer Shape Factor</td>
</tr>
<tr>
<td>16</td>
<td>Growth of the Boundary Layer Momentum Thickness</td>
</tr>
<tr>
<td>17</td>
<td>Nozzle 1 - Dimensional Exit Mach Number vs Area Ratio</td>
</tr>
<tr>
<td>18</td>
<td>Divergence Efficiency for Contoured Nozzles</td>
</tr>
<tr>
<td>19</td>
<td>Heat Transfer Model Cooling Methods</td>
</tr>
<tr>
<td>20</td>
<td>Heat Transfer Modeling</td>
</tr>
<tr>
<td>21</td>
<td>Impinging Element Injector Manifolding and Flow Schematic</td>
</tr>
<tr>
<td>22</td>
<td>Heat Sink Chamber</td>
</tr>
<tr>
<td>23</td>
<td>GO₂/GH₂ Engine Start Sequence and Pressure History</td>
</tr>
<tr>
<td>24</td>
<td>Dynamic Response - Aerojet GO₂/GH₂ Engine</td>
</tr>
<tr>
<td>25</td>
<td>Specific Impulse vs Mixture Ratio</td>
</tr>
<tr>
<td>26</td>
<td>Aerojet Regeneratively Cooled Thrust Chamber</td>
</tr>
<tr>
<td>27</td>
<td>Thermal Characteristics of Regeneratively Cooled Thrust Chamber Design</td>
</tr>
<tr>
<td>28</td>
<td>Temperature vs Location - Aerojet Regeneratively Cooled Thrust Chamber Comparison of Analysis</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Influence of System or Design Variable on Performance Parameter</td>
<td>4</td>
</tr>
<tr>
<td>II</td>
<td>$\text{H}_2/\text{O}_2$ Chemical System</td>
<td>24</td>
</tr>
<tr>
<td>III</td>
<td>Wall Boundary Conditions</td>
<td>28</td>
</tr>
</tbody>
</table>
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 \( \text{GO}_2/\text{GH}_2 \) 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 \( \text{O}_2/\text{H}_2 \) 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 \( \text{GO}_2/\text{GH}_2 \) 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 \( \text{GO}_2/\text{GH}_2 \) 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 \( \text{GO}_2/\text{GH}_2 \) 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 \( \text{Isp} \) 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 \( \text{Isp} \). The final performance calculations return an updated thrust coefficient value to the dynamics model where accumulated impulse and \( \text{Isp} \) are determined at each iteration.
### TABLE I. INFLUENCE OF SYSTEM OR DESIGN VARIABLE ON PERFORMANCE PARAMETER

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

<table>
<thead>
<tr>
<th>X</th>
<th>Important</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Secondary</td>
</tr>
<tr>
<td>-</td>
<td>No effect</td>
</tr>
</tbody>
</table>
FIG. 1

GH2/O2 REACTION CONTROL SYSTEM
SYSTEM SCHEMATIC

EXIT
MOZZLE

TRANSONIC

COMBUSTOR

IGNITOR

MANIFOLD

PILOT

MANIFOLD

MANIFOLD

MANIFOLD

MANIFOLD

MANIFOLD

INJECTOR

VALVE (MAIN)

VALVE (MAIN)

ISOLATION
VALVE

ISOLATION
VALVE

CARBURATOR
OR
REGULATOR

H2 TANK

O2 TANK

INTERFACE
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 GO$_2$/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.

Instantaneous 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.
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 $\text{O}_2/\text{H}_2$ 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
2. Heat sink cooling
3. Radiation cooling
4. Film cooling
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

\[ \dot{w} = A_D (P_{o1} - P_{o2}) \]  \hspace{1cm} (1)

For orifices, the admittance is given by

\[ A_D = \frac{A \delta P_{o1}}{(P_{o1} - P_{o2})} \sqrt{\frac{\gamma}{RT_{o1}}} F(\gamma) \]  \hspace{1cm} (2)

\[ F(\gamma) = \sqrt{\frac{2}{\gamma + 1}} \frac{\gamma + 1}{\gamma - 1} \text{ for choked flow} \]  \hspace{1cm} (3)

\[ = \sqrt{\frac{2}{\gamma - 1}} \sqrt{\left(\frac{P_{o2}}{P_{o1}}\right)^{2\gamma} - \left(\frac{P_{o2}}{P_{o1}}\right)^{\gamma + 1}} \text{ for unchoked flow} \]

The choking pressure ratio is given by

\[ \left(\frac{P_{o2}}{P_{o1}}\right)_{\text{choking}} = \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma}{\gamma - 1}} \]  \hspace{1cm} (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.

\[ \left(\frac{P_{o2}}{P_{o1}}\right)_{\text{choking}} = C_1 + C_2 \left(\frac{4fL}{D}\right) + C_3 \left(\frac{4fL}{D}\right)^2 + \ldots \]  \hspace{1cm} (5)
The flow admittance is also obtained from the results shown in Reference 3.

\[ A_D = \frac{AB P_0}{(P_0 - P_0^2) \sqrt{RT_0}} \]  

\[ (6) \]

B is a function of \( (4fL/D) \) for choked flow and a function of \( 4fL/D \) and \( P_0^2/P_0 \) for unchoked flow. The dynamics model uses B in the form of polynomial equations.

The friction factor, \( 4f \), is a function of the flow rate 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

\[ 4f = \frac{64}{Re} \]  

\[ (7) \]

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

\[ P_0(I)_{t+\Delta t} = P_0(I)_t + \frac{\Delta t}{C(I)_{t+\Delta t}} \sum_{\text{all } J \text{ connected to } I} \dot{w}(J) \]  

\[ (8) \]

\( C(I) \) is the volume capacity defined as follows

\[ C(I)_{t+\Delta t} = \frac{V(I)MW(I)_{t+\Delta t}}{RT_0} \]  

\[ (9) \]

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

\[ T_0(I)_{t+\Delta t} = \frac{H(I)_t + \Delta t \sum_{\text{all } J \text{ connected to } I} \dot{q}(J)_t}{W(I)_t + \Delta t \sum_{\text{all } J} \dot{w}(J)_t \left(3.5 - 3.26 M(I)_{t+\Delta t}\right)} \]  

\[ (10) \]

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

\[ \dot{q}(J) = T_0(J) \dot{w}(J) \left(3.5 - 3.26 m(J)\right) \]
for cold flow. For ignited flow, \( q(J) \) is calculated using a polynomial equation in \( m(J) \). \( T_0(J) \) 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, \( \Delta t \), 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

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

where restrictions \( J \) connect volume \( I \) to volumes \( K \).

The coefficients of \( P_o(K)_t \) are always positive, but the coefficient of \( P_o(I)_t \) will be negative if

\[
\Delta t > \frac{C(I)_{t+\Delta t}}{\sum A_D(J)}
\]

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, \( \Delta t \) is made small enough to avoid negative coefficients in the system by assuring that

\[
\Delta t \leq \min_{all I} \left[ \frac{C(I)_{t+\Delta t}}{\sum J A_D(J) \text{ all } J \text{ connected to } I} \right]
\]
SUBROUTINE DYNAM VARIABLE
FLOW AREA MODEL

CAREA

TIMON

TIMOFF

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

\[ F = P_o A^* C_F \]  

(14)

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

Accumulated impulse if given by

\[ I(t) = \sum_{i=1}^{N} F_i (\Delta t)_i \]  

(15)

Integrated specific impulse if given by

\[ I_{Sp}(t) = \frac{I(t)}{\dot{w}_T(t)} \]  

(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-induced 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.

\[ \begin{align*} 
    w_{O_2}' &= 1/2 m \dot{w}_T - \delta \\
    w_{O_2}'' &= 1/2 m \dot{w}_T + \delta \\
    w_{H_2}' &= (1 - m) 1/2 \dot{w}_T + \frac{\delta}{m} \\
    w_{H_2}'' &= (1 - m) 1/2 \dot{w}_T - \frac{\delta}{m} 
\end{align*} \]  

(17)
The δ is determined by iteration to yield the input mixing factor defined as in Reference 14 by

\[
E_m = 1 - \left[ \left( \frac{\dot{m}_1}{\dot{m}_2} \right)^{m - m'} + \left( \frac{\dot{m}_2}{\dot{m}_3} \right)^{m'' - m} \right] \tag{18}
\]

where \( m' = \frac{\dot{m}_2}{\dot{m}_1 + \dot{m}_2} \)

\( m'' = \frac{\dot{m}_3}{\dot{m}_1 + \dot{m}_3} \)

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

\[
\frac{A'}{A''} = \frac{\dot{m}_2}{\dot{m}_3} \sqrt{\frac{T_0'}{T_0''} \left( \frac{MW'}{MW''} \right)} \tag{19}
\]

The total temperature in the fuel rich region is

\[
T_0' = \frac{.24 \dot{m}_1 T_{O_2} + 3.5 \dot{m}_2 T_{H_2} - 3.5 \dot{m}_3 T_{H_2}}{3.5 - 3.26 \dot{m}_3} \tag{20}
\]

and the molecular weight is

\[
MW' = \frac{64.51}{2.016 \dot{m}' + (32 (1-m'))} \tag{21}
\]

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:

\[
\frac{\dot{w}_{O_2} + \dot{w}_{H_2}}{A'} = \left( \rho_0' a_{T'} \right) \left( 1 - \frac{M^2}{5} \right)^{-\frac{5}{2}} \left( \frac{M^2}{1 + 0.2M^2} \right)^{1/2}
\]

where

\[
\rho_0' = \frac{P_o' \left( MW' \right)}{(18540) T_o'}
\]

\[
a_{T'} = \sqrt{\frac{yRT_o'}{MW'}}
\]

\[
u' = a_{T'} \sqrt{\frac{M^2}{1 + 0.2M^2}}
\]

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:

\[
N_E = 1 + \sum_{n=1}^{N_A} 6n
\]

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

\[
\int_0^\psi \psi d\psi = \int_0^\psi \rho u y dy
\]

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 \((10 \times N_A)\) 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:

\[
\rho vr^N = \psi^N \psi_x \tag{26}
\]

\[
- \rho vr^N = \psi^N \psi_x
\]
SPARK BREAKDOWN VOLTAGE CRITERIA

Breakdown Potential (Volts)

Pressure X Spark Gap (PSI INCHES)
MINIMUM IGNITION ENERGY

ENERGY (Millijoules)

MIXTURE RATIO
EFFECT OF MIXTURE RATIO ON FLAME QUENCHING PARAMETER*

\[
(PD)_{\text{Limit}} = R T_o \left[ \frac{\Delta h (T_F - T_W)}{K \Delta H_r \exp(-E/R T_F)} \right]
\]

Contraction Ratio Range 1-6
Temperature 540°F

\[
Q_{\text{gen}} \geq Q_{\text{Loss}}
\]

\[
H_2/O_2 \rightarrow Q_{\text{Loss}}
\]

Quenching Parameter
\( P \times D \) (lb/in\(^2\)) x (in)

* "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**

\[
\frac{\partial \tilde{\alpha}_i}{\partial t} = \left( \frac{1}{\Psi^N} \right) \frac{\partial}{\partial \Psi} \left[ \left( \frac{Le}{Pr} \right) \left( \rho \mu \Psi^N \right)_r \frac{2N}{\Psi} \frac{\partial \tilde{\alpha}_i}{\partial \Psi} \right]
\]

where

\[
\tilde{\alpha}_j = \sum_i \nu_{ji} \left( \frac{w_j}{w_i} \right) \alpha_i
\]

and \( \nu_{ji} \) is the amount of element \( j \) in specie \( i \) and the \( w_i \)'s are the molecular weights.

**Specie Conservation**

\[
\frac{\partial \alpha_i}{\partial t} = \left( \frac{1}{\Psi^N} \right) \frac{\partial}{\partial \Psi} \left[ \left( \frac{Le}{Pr} \right) \left( \rho \mu \Psi^N \right)_r \frac{2N}{\Psi} \frac{\partial \alpha_i}{\partial \Psi} \right] + \dot{\omega}_i / \rho \Psi
\]

where \( \dot{\omega}_i \) is the volumetric rate of production of specie \( i \). Note also that:

\[
\sum_i \nu_{ji} \dot{\omega}_i \left( \frac{w_j}{w_i} \right) = 0
\]

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 indeterminacy approached by the production term, \( \dot{\omega}_i \), 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**

\[
\frac{\partial u}{\partial x} = \left( \frac{1}{\Psi^N} \right) \frac{\partial}{\partial \Psi} \left[ \left( \rho \mu \Psi^N \right)_r \frac{2N}{\Psi} \frac{\partial u}{\partial \Psi} \right] - \left( \frac{1}{\rho u} \right) \frac{\partial p}{\partial x}
\]
Energy

\[
\frac{\partial H}{\partial x} = \left(1/\Psi^N\right) \frac{\partial}{\partial \Psi} \left( (\rho u_\mu/Pr \Psi)^N \right) 2N \frac{\partial H}{\partial \Psi} + (Pr-1) \frac{\partial u^2}{2}{\partial \Psi} + \\
+ \sum\limits_i h_i (Le-1) \frac{\partial^2 h_i}{\partial \Psi^2}
\]

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

State:

\[
\rho = \frac{P}{RT \sum_i (a_i / W_i)}
\]

also

\[
H = h + \frac{u^2}{2}
\]

where

\[
h = \sum_i a_i h_i
\]

with*

\[
h_i = h_i (T)
\]

In addition, representations for the turbulent transport coefficients \(\mu\), \(Pr\) 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 \( \mu \) - (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

\[
\mu = \kappa \frac{\rho u D}{1 + \kappa / S}
\]

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 \( w_i \) are given by

\[
\dot{w}_i = w_i \sum_{p=1}^{R} (\nu''_{ip} - \nu'_{ip}) f_{p, p}^{0} \prod_{i=1}^{N_c, p} \left[ 1 - \frac{\rho}{k_{c, p}} \prod_{i=1}^{N} \left( \frac{\alpha_i}{w_i} \right)^{\nu''_{ip} - \nu'_{ip}} \right]
\]

\[
\dot{m}_p = \sum_{i=1}^{N} \nu'_{ip} ; \quad \dot{N}_p = i=1^{P} (\nu''_{ip} - \nu'_{ip})
\]

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

\[
\sum_{i=1}^{N} \nu'_{ip} M_i = \frac{k_{f, p}}{k_{b, p}} \sum_{i=1}^{N} \nu''_{ip} x_i \quad p = 1, 2, \ldots, R
\]

The present study employs the reaction mechanism given in Table II which involves 8 species entering into a total of 17 reactions.
# TABLE II. H₂/O₂ CHEMICAL SYSTEM

<table>
<thead>
<tr>
<th>Reaction No.</th>
<th>Reaction</th>
<th>A</th>
<th>B</th>
<th>E/R (10⁻⁰K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H₂O₂ + H = O₂ + H₂</td>
<td>6 x 10⁻¹³</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>H₂O₂ + H = OH + OH</td>
<td>6 x 10⁻¹³</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>H₂O₂ + O₂ = O₂ + OH</td>
<td>1 x 10⁻¹³</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>H₂O₂ + OH = O₂ + H₂O</td>
<td>1 x 10⁻¹³</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>H₂ + OH = H + H₂O</td>
<td>2.19 x 10⁻¹³</td>
<td>0</td>
<td>2,593</td>
</tr>
<tr>
<td>6</td>
<td>OH + OH = O₂ + H₂O</td>
<td>5.75 x 10⁻¹²</td>
<td>0</td>
<td>392.7</td>
</tr>
<tr>
<td>7</td>
<td>H₂ + O = H + OH</td>
<td>1.74 x 10⁻¹³</td>
<td>0</td>
<td>4,758</td>
</tr>
<tr>
<td>8</td>
<td>O₂ + H = OH + O</td>
<td>2.29 x 10⁻¹⁴</td>
<td>0</td>
<td>8,459</td>
</tr>
<tr>
<td>9</td>
<td>H₂O₂ + OH = H₂O + HO₂</td>
<td>1 x 10⁻¹³</td>
<td>0</td>
<td>906.3</td>
</tr>
<tr>
<td>10</td>
<td>H₂O₂ + H = H₂ + HO₂</td>
<td>2.34 x 10⁻¹³</td>
<td>0</td>
<td>4,632</td>
</tr>
<tr>
<td>11</td>
<td>H₂O₂ + H = OH + H₂O</td>
<td>3.18 x 10⁻¹⁴</td>
<td>0</td>
<td>4,532</td>
</tr>
<tr>
<td>12</td>
<td>O₂ + H + M = OH + M</td>
<td>1 x 10⁻¹⁶</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>O + O + M = O₂ + M</td>
<td>9.38 x 10⁻¹⁴</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>H + H + M = H₂ + M</td>
<td>5 x 10⁻¹⁵</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>H + OH + M = H₂O + M</td>
<td>1 x 10⁻¹⁷</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>O₂ + H + M = HO₂ + M</td>
<td>1.59 x 10⁻¹⁵</td>
<td>0</td>
<td>-503.5</td>
</tr>
<tr>
<td>17</td>
<td>OH + OH + M = H₂O₂ + M</td>
<td>8.4 x 10⁻¹⁴</td>
<td>0</td>
<td>-2,669</td>
</tr>
</tbody>
</table>

Notes:

\[
k_f = AT^B e^{-E/RT}
\]

\[
k_b = k_f / k_c
\]

\(k_f\) 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:

$$\begin{align*}
\text{at } x = 0 \text{ (starting station) and } 0 \leq r \leq r_w(0) & \quad \{ \\
0 \leq r \leq r_1 & \quad \{ u = u_1(r) \\
& \quad \{ T = T_1(r) \\
& \quad \{ \alpha_i = \alpha_{i,1}(r) \\
\quad \vdots & \quad \{ u = u_2(r) \\
\quad \vdots & \quad \{ T = T_2(r) \\
\quad \vdots & \quad \{ \alpha_i = \alpha_{i,2}(r) \\
\quad \vdots & \quad \{ u = u_k(r) \\
\quad \vdots & \quad \{ T = T_k(r) \\
\quad \vdots & \quad \{ \alpha_i = \alpha_{i,k}(r)
\end{align*}$$

- **Boundary conditions - thrust chamber geometry** - To provide the versatility of either predicting behavior of existing hardware or 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.
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 \( r = 0 \)

  \[
  \frac{\partial \alpha}{\partial \psi} = \frac{\partial \tilde{\alpha}}{\partial \psi} = \frac{\partial T}{\partial \psi} = \frac{\partial H}{\partial \psi} = \frac{\partial u}{\partial \psi} = 0
  \]

- **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 \(x-\psi\) 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.

\[
\frac{\partial P}{\partial x} = \frac{1}{\Delta x}(P_{n+1,m} - P_{n,m})
\]

\[
\frac{\partial P}{\partial \psi} = \frac{1}{2\Delta \psi}(P_{n,m+1} - P_{n,m-1})
\]

\[
\frac{\partial}{\partial \psi} \left[ \frac{\partial \psi}{\partial \psi} \right] \left[ \frac{\partial}{\partial \psi} \left[ \frac{\partial \psi}{\partial \psi} \right] \right] = \frac{1}{\Delta \psi^2} \left[ b_{n,m,k}(P_{n,m+1} - P_{n,m}) - b_{n,m,k}(P_{n,m} - P_{n,m-1}) \right]
\]
### TABLE III. WALL BOUNDARY CONDITIONS

\(( x \geq 0 \; ; \; r = r_w(x) \) )

<table>
<thead>
<tr>
<th>Cooling Model Variable</th>
<th>Film</th>
<th>External Regenerative</th>
<th>Transpiration</th>
</tr>
</thead>
<tbody>
<tr>
<td>H (energy transfer)</td>
<td>( \frac{\partial H}{\partial \psi} = 0 )</td>
<td>( \frac{\partial H}{\partial \psi} = - q_b \left( \frac{\mu}{\rho u \nu} \right) b \frac{N}{N_\nu} )</td>
<td>( H = H(a, T) )</td>
</tr>
<tr>
<td>( a_i ) (mass transfer)</td>
<td>( \frac{\partial a_i}{\partial \psi} = 0 )</td>
<td>( \frac{\partial a_i}{\partial \psi} = \dot{m}<em>c \left( \frac{P}{V} \right) \left( \frac{N}{N</em>\nu} \right) ); coolant</td>
<td>( \frac{\partial a_i}{\partial \psi} = \dot{m}<em>c \left( \frac{P}{V} \right) \left( \frac{N}{N</em>\nu} \right) ); tracer</td>
</tr>
<tr>
<td>( u ) (shear)</td>
<td></td>
<td>( \frac{\partial u}{\partial \psi} = - \frac{\mu a}{\mu - \mu^*} b \frac{N}{N_\nu} \frac{c_f}{2} )</td>
<td></td>
</tr>
</tbody>
</table>

**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
- \( T_w(x) \) is specified and \( q_L(x) \) computed
FINITE DIFFERENCE GRID NETWORK
where

\[ b = \left( \frac{\rho \mu}{\psi} \right)^2 \]

(43)

and

\[ b_{n,m+1} = \frac{1}{2} (b_{n,m} + b_{n,m+1}) \]

(44)

The conservation equations in difference form are:

Elements:

\[ \mathbf{m} = 0; \]

(46A)

\[ \begin{align*}
(\tilde{\alpha}_j)_{n+1,0} &= (\tilde{\alpha}_j)_{n,0} \\
&\quad + \frac{2\Delta x (N+1)}{\Delta \psi^2} \left[ (Le \mu / Pr) (\rho u) \right]_{n,0} \left[ (\tilde{\alpha}_j)_{n,1} - (\tilde{\alpha}_j)_{n,0} \right]
\end{align*} \]

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

\[ m = 0 \]

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

\[ \quad \cdots - \frac{\Delta x}{(\rho u)_{n,0}} \left( \frac{\partial p}{\partial x} \right)_{n+1} \tag{47} \]

\[ m \neq 0 \]

\[ u_{n+1,m} = u_{n,m} + \frac{\Delta x}{m_N (\Delta \Psi)^{2+N}} \left\{ b_{n,m+\frac{1}{2}} u_{n,m+1} \right\} \]

\[ \quad \cdots - \frac{u_{n,m} \left( b_{n,m+\frac{1}{2}} + b_{n,m-\frac{1}{2}} \right)}{b_{n,m-\frac{1}{2}} u_{n,m-1}} \]

\[ \quad \cdots - \frac{(\Delta x)}{\rho u \ n,m} \left( \frac{\partial p}{\partial x} \right)_{n+1} \tag{48} \]

Energy

\[ m = 0 \]

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

\[ \quad \cdots \frac{1}{2} \left( 1 - \frac{1}{Pr} \right)_{n,0} (u_{n,1} - u_{n,0}) \]

\[ \quad \cdots \frac{\mu}{\rho} \left( \alpha_i \right)_{n,0} \left( \alpha_i \right)_{n,0} \left( \frac{Le-1}{Pr} \right)_{n,0} \tag{49} \]
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:

\[
\frac{\Delta \psi^2}{6(1+N)} \left[ \frac{Pr/ \Sigma i}{(\rho u)^{1-N}} \right] \geq \Delta x \leq \frac{1}{3} \left[ \frac{N \Delta \psi^{2+N}}{m_{n,m+1}^2 + (Le \; \Sigma i)} \right] \]

\[
H_{n+1,m} = H_{n,m} + \frac{\Delta \psi}{m} \left( \frac{b}{Pr} \right)_{n,m+1} + \frac{1}{2} \left( \frac{b-b/Pr}{Pr} \right)_{n,m} u^2_{n,m+1} \]

\[- \left( \frac{b}{Pr} \right)_{n,m+1} + \left( \frac{b}{Pr} \right)_{n,m} H_{n,m} \]

\[+ \left( \frac{b}{Pr} \right)_{n,m} H_{n,m-1} + \frac{1}{2} \left( \frac{b-b/Pr}{Pr} \right)_{n,m} u^2_{n,m} \]

\[+ \frac{1}{2} \left( \frac{b-b/Pr}{Pr} \right)_{n,m+1} + \frac{1}{2} \left( \frac{b-b/Pr}{Pr} \right)_{n,m-1} u^2_{n,m} \]

\[+ \frac{1}{2} \left( \frac{b-b/Pr}{Pr} \right)_{n,m} u^2_{n,m} \]

\[+ \sum_i (a_i)_{n,m} \left[ bh_i \left( \frac{Le-1}{Pr} \right)_{n,m+1} + bh_i \left( \frac{Le-1}{Pr} \right)_{n,m-1} \right] \]

\[+ \sum_i (a_i)_{n,m-1} \left[ bh_i \left( \frac{Le-1}{Pr} \right)_{m,m-1} \right] \]
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*

\[ \ln \eta^I = \left( \text{Pr} \frac{R^I}{c_c} \right)^{1/8} \phi^I [0.04 - \frac{(x-x_s^I)}{G^I s^I}] \]

where the effectiveness parameter is defined by

\[ \eta^I = \frac{(T^I - T^I_{w1})}{(T^I_{w2} - T^I_{c})} \]  

and

\[ G^I = \frac{(\rho u c_p)^I}{h^I} \]

\[ \phi^I = \begin{cases} (U^I)^{1/8} [1 + 0.4 \arctan(U^I - 1)] & ; U^I \geq 1.0 \\
(U^I)^{13/8} \left[ (U^I) [1 - 1/U^I] \right] & ; U^I \leq 1.0 \end{cases} \]

*More precisely, these equations apply in the region bounded by \((x-x_s^I)/G^I s^I \geq 0.04\) 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 \(T^I_{c}\) as has been indicated in Figure 9.
Following Bartz, the heat transfer coefficient $h_g^I$, appearing in Equation (53), is calculated from

$$h_g^I = \sigma \left( \frac{R_b^*}{A^*} \right)^{0.9} \left[ \frac{0.025}{(D^*)^{0.2}} \left( \frac{c_p^*}{\mu_b^*} \right) \left( \frac{D^*}{c_b^*} \right) \left( \frac{R_b^*}{D^*} \right)^{0.1} \right]$$  \hspace{1cm} (55)

where

$$\sigma^I = \left[ \frac{1}{2} \left( 1 + \frac{\rho_w^I}{\rho_b^I} \right) \right]^{8} \left[ \frac{1}{2} \left( 1 + \frac{\mu_w^I}{\mu_b^I} \right) \right]$$  \hspace{1cm} (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 $T_w^0$. Consistent with the manner in which the Hatch-Pappel model was developed for single slots, we take $T_w^0$ 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 $(H_{aw})_b$.

$$\frac{(H_{aw})_b}{(H_{aw})_b} = h_b + (Pr)_b^{1/3} \frac{u_b^2}{2}$$  \hspace{1cm} (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 $T_c^b$, 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 $x_S^I < x_n < x_S^{I+1}$ 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 $(T_w)^{I+1}_{n+1}$. 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

$$H_b = \frac{2\pi \rho uHr dx}{\dot{m}}$$

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

$$h_g^I = \sigma \left( \frac{R_b^*}{A^*} \right)^{0.9} \left[ \frac{0.025}{(D^*)^{0.2}} \left( \frac{c_p^*}{\mu_b^*} \right) \left( \frac{D^*}{c_b^*} \right) \left( \frac{R_b^*}{D^*} \right)^{0.1} \right]$$  \hspace{1cm} (55)

where

$$\sigma^I = \left[ \frac{1}{2} \left( 1 + \frac{\rho_w^I}{\rho_b^I} \right) \right]^{8} \left[ \frac{1}{2} \left( 1 + \frac{\mu_w^I}{\mu_b^I} \right) \right]$$  \hspace{1cm} (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 $T_w^0$. Consistent with the manner in which the Hatch-Pappel model was developed for single slots, we take $T_w^0$ 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 $(H_{aw})_b$.

$$\frac{(H_{aw})_b}{(H_{aw})_b} = h_b + (Pr)_b^{1/3} \frac{u_b^2}{2}$$  \hspace{1cm} (57)
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 \( \mu_b, \rho_b, (T_{aw})_{b}, \) etc. from appropriate auxiliary thermodynamic relations, e.g.:

\[
(T_{aw})_b = \text{func} \left[ (H_{aw})_b, (\alpha_{i,b})_b \right]
\]

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

\[
\rho_w = \frac{W_w}{W_b} \frac{T_b}{T_w}
\]

and

\[
\mu_w = \text{func} (T_w)
\]

the only unknowns appearing in Equation (51) are \( (T_w)_{n+1} \) and \( (T_w)_{n+1}^{-1} \)

*It is important to note here that \( \mu_b \) 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 \( \mu \) from the Sutherland formula for air; viz:

\[
\mu_b = 2.27 \frac{T_b^{3/2}}{T_b + 158.6} \times 10^{-8} \frac{\text{lb}. \text{sec.}}{\text{ft}^2}
\]

where \( T_b \) is in degrees Rankine.
But

\[ \ln \eta^{I-1} = \left( \frac{Pr_c R_c}{\mathcal{G}} \right)^{I-1/8} \left[ 0.04 - (x-x_s) \right] / G^{I-1} \]

\[ \eta^{I-1} = \left( T_w^{I-2} - T_w^{I-1} \right) / \left( T_w^{I-2} - T_c^{I-1} \right) \]

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

\[ (T_w)^{I+1}_{n+1}, (T_w)^{I+2}_{n+1}, \ldots, (T_w)^{I+1}_n, (T_w)^{I+1}_{n+1}' \]

by iterative solution of the transcendental Equation (51).

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

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:

\[ \frac{(T_{aw}) - T_w}{(T_{aw}) - T_c} = 1 - \left( 1 + \frac{G}{3} \right)^{-3} \]

where

\[ G = \frac{1}{S_t} \left( \frac{\dot{m}c_p}{\rho u c_p} \right)_e \]

\[ \frac{(\dot{m}c_p)}{\rho u c_p} \]
\[ (T_{aw})_o = \text{fcn} \left[ (H_{aw})_o , \alpha_{ie} \right] \]
\[ (H_{aw})_o = h_e + (Pr)^{1/3} \frac{u_e^2}{2} \]

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:

\[
S_{to} = 0.0592 (Pr)^{-2/3} \left( \frac{\rho^* u^*}{\mu^*} \right) \left( \frac{\alpha^*}{\rho^*} \right) \left( \frac{p^* u^* x}{\mu^*} \right)^{-1/5} \tag{60}
\]

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

\[
h^* = \frac{1}{2} (H_w + h_e) + 0.22 (H_{aw} - h_e) \tag{61}
\]

with Knuth's representation for the reference composition:

\[
\alpha_i^* = \alpha_i \left( \frac{\rho_i}{\rho_e} \right) \quad i \neq \text{c} \]
\[
\alpha_c^* = 1 - \alpha_e^* \tag{62}
\]

where

\[
\alpha_c^* = \left( \frac{\overline{W}_e}{\overline{W}_c} \right) \frac{\ln (W/W')}{\ln \left[ \frac{(1-\alpha_c^*) W_e}{(1-\alpha_c^*) W_c} \right]} \tag{63}
\]

\[
\overline{W}_c = (1-\alpha_c^*) \left( \sum_i \alpha_i W_i \right)^{-1}
\]
\[
\overline{W}_w = (\sum_i \alpha_i W_i)^{-1}
\]

We note here that the parameter \( \overline{W}_c \) 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 \( \alpha_i \) 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 \( x = x_{n+1} \), \( (\alpha_{i\text{w}})_{n+1} \), 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 \( \alpha_{i\text{w}} \) 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 \( \rho* \) and \( \mu* \) 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 \( \alpha_{T,*} \leq 0.01 \alpha_{T_w} \).*
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

$$I_{sp} = \frac{\sum (I_{spK_i} \dot{w}_i)}{\sum (\dot{w}_i - \dot{w}_c)} - (I_{sp})_s (1 - \eta_D) - \left( \frac{\Delta F_{BL}}{\dot{w}_T} + \frac{I_{spc}}{\dot{w}_c} \right)$$  

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 Laboratories 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 O₂ and H₂. In order to make the data adaptable to propellant at any temperature, the data was modified to the form of percent recombination of

\[ \eta_R = \frac{I_{spK} - I_{spF}}{I_{spS} - I_{spF}} \]

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, \( \gamma \), 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 terms of thrust is

\[
\Delta F = 2\pi r_e P_o \left( \rho_e u_e^2 / P_o \right) \theta \cos \alpha \left( 1 - \left( P/P_o \right) \left( \delta / \theta \right) \right)
\]
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

\( r_c/r_t = 1.0 \)

![Diagram showing nozzle area ratio gradients with specific heat ratio 1.25 and ideal ratio of 1.0.](image)
NORMALIZED GRAPHICAL SOLUTION FOR FREEZING AREA RATIO USING MODIFIED BRAY ANALYSIS (REF 12)

\[ \text{H}_2(\text{g}) + \text{O}_2(\text{g}) \]

\[ P_c = 500 \text{ PSIA} (3.448 \times 10^6 \text{ N/m}^2) \]

\[ O/F = 6.00 \]

NOTE: REACTION RATE CONSTANTS LISTED IN TABLE H-1

TABLE OF REACTIONS

<table>
<thead>
<tr>
<th>No.</th>
<th>Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H + H + H → H(_2) + H</td>
</tr>
<tr>
<td>2</td>
<td>H + H + H(_2) → H(_2) + H(_2)</td>
</tr>
<tr>
<td>3</td>
<td>H + H + H(_2)O → H(_2) + H(_2)O</td>
</tr>
<tr>
<td>4</td>
<td>H + OH + H(_2)O → H(_2)O + H(_2)O</td>
</tr>
<tr>
<td>5</td>
<td>H + OH + H(_2) → H(_2)O + H</td>
</tr>
<tr>
<td>6</td>
<td>H + OH + H → H(_2)O + H</td>
</tr>
</tbody>
</table>

COMPOSITE KINETIC (TRANSITION FACTOR = 1)

EQUILIBRIUM
(15 DEG CONE, \( r_1 = 1 \text{ FT} (0.3048 \text{ m}), r_1/r_2 = 1 \))

EQUILIBRIUM
\[ (1/r_1) \left[ \frac{6 (A/A_{MIN})}{4 (x/r_2)} \right] = 1 \text{ FT}^{-1} \]

\( r_1 = 1.0 \text{ FT} (0.3048 \text{ m}) \)

NOTE: REACTION RATE CONSTANTS LISTED IN TABLE H-1
EFFECT OF FREEZING AREA RATIO ON NONEQUILIBRIUM PERFORMANCE FOR HYDROGEN-OXYGEN PROPELLANT SYSTEM (REF 12)

$H_2(g) - O_2(g)$

$P_c = 500$ PSIA $\left(3.448 \times 10^6 \text{N/m}^2\right)$

$O/F = 6.00$

FREEZING AREA RATIO, $(A/A_{MIN})_{FR}$

Vacuum Specific Impulse, $I_{SP, vac} - \text{SEC}$
EFFECTIVE NOZZLE WALL ANGLE FOR CONTOURED NOZZLE

\[ \gamma = 1.2 \]
BOUNDARY LAYER SHAPE FACTOR

\[ \gamma = 1.2 \]

\[ \frac{\delta^*}{\theta} \]

Mach Number
GROWTH OF THE BOUNDARY LAYER MOMENTUM THICKNESS

\[ \gamma = 1.2 \]

\[ \frac{\theta}{r} \left[ \frac{C}{Re} \right]^{1/5} \left| \sin \alpha \right|^{1/2} \]

Mach Number
NOZZLE 1 - DIMENSIONAL EXIT MACH NUMBER VS, AREA RATIO

\[ \gamma = 1.2 \]
where \( \frac{P}{\rho u^2} = \left( \frac{1}{\gamma M^2} \right) \) and

\[
\frac{\rho e u^2}{P} = \left( \gamma M^2 \right) \frac{1}{1 + \frac{\gamma - 1}{2} M^2} \gamma^{-1}
\]

(66)

\[
\theta_e = \left[ \frac{1}{1 + e \left( \frac{\gamma}{\gamma - 1} \right) \frac{1}{Re} \left( \frac{r_e}{(\sin \alpha)} \right)^2} \right]^{\frac{1}{2}}
\]

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 assumed 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

<table>
<thead>
<tr>
<th>Streamline</th>
<th>Radius</th>
<th>Area (( i )) of Streamtube</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( r_1 = 0 )</td>
<td>( \pi \left( \frac{r_2}{2} \right) )</td>
</tr>
<tr>
<td>2</td>
<td>( r_2 )</td>
<td>( \pi \left[ \left( \frac{r_3 + r_2}{2} \right)^2 - \left( \frac{r_1 + r_2}{2} \right)^2 \right] )</td>
</tr>
<tr>
<td>i</td>
<td>( r_i )</td>
<td>( \pi \left[ \left( \frac{r_{i+1} + r_i}{2} \right)^2 - \left( \frac{r_{i-1} + r_i}{2} \right)^2 \right] )</td>
</tr>
<tr>
<td>n</td>
<td>( r_n )</td>
<td>( \pi \left( \frac{r_n + r_{n-1}}{2} \right) )</td>
</tr>
</tbody>
</table>
DIVERGENCE EFFICIENCY FOR CONTOURED NOZZLES

\[ \frac{p_e}{p_o} = 0 \]
\[ \gamma = 1.2 \]

from Ref. 1
where \( r = \) wall radius, will, when combined with the individual \( \dot{w}/A \) 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 \( \text{CO}_2/\text{GH}_2 \) 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

\[
\sum_{j \text{ connected to } i} \left[ A_{D_{i,j}} \left( T_j - T_i \right) \right]_t = \frac{C_i}{\Delta t} \left( \frac{T_i}{\Delta t} + \Delta t - T_i \right)
\]  

which can be rearranged to give \( T_i + \Delta t \) explicitly.

\[
T_i + \Delta t = \frac{\Delta t}{C_i} \sum_{j \text{ connected to } i} \left( A_{D_{i,j}} T_j \right) + \left( 1 - \frac{\sum A_{D_{i,j}} \Delta t}{C_i} \right) T_i
\]  

(68)
HEAT TRANSFER MODEL

COOLING METHODS

FIG. 19
HEAT TRANSFER MODELING

Conductive Admittance \[ = \frac{KA}{l} \]

Convective Admittance \[ = h_s A \]

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 \[ = \alpha C_p V \] (Mass Nodes)
\[ = 0 \] (Boundary Nodes)
In other words, the temperature of node $i$ at time $(t + \Delta t)$ depends upon the weighted average of the temperatures of nearby nodes and the temperature of $i$ at time $t$.

Except for the coefficient of $T_i$, the weights are inherently positive. In order to assure solution stability, all $T_i$ coefficients are kept non-negative by selecting time increments, $\Delta t$, such that

$$\Delta t \leq \frac{C_i}{\sum_{j \text{ connected to } i} A_{D_{i,j}}}$$

(69)

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

$$T_i = \frac{\sum_{j \text{ connected to } i} A_{D_{i,j}}}{\sum A_{D_{i,j}}} T_j$$

(70)

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

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

Prandtl number is calculated using the approximate result

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

(71)

Specific heat, $C_p$, 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 $15^\circ$ half angle cone.
The local view factor is given by a polynomial curve fit of previously calculated view factor data for 15° cones.

\[ \mathcal{J} = \mathcal{J} \left( \frac{L}{D^*}, \frac{X}{L} \right) \]

**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

\[ h_g = 0.048 \frac{k}{D_H} \frac{Re_b^{0.8}}{Pr_b^{0.4}} \left( \frac{T_w}{T_b} \right)^{-0.55} \]

where:

\[ Re_b = \frac{\omega D_H}{\mu} \]

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 \( \text{Res} = 1/\dot{w}C_p \). 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

\[ \frac{A_x}{A} = \frac{1}{M} \left[ \left( \frac{2}{\gamma + 1} \right) \left( 1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{\frac{\gamma + 1}{2(\gamma - 1)}} \]

for Mach number by Newton's method. The ratio of specific heats, \( \gamma \), 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:

$$\epsilon = \frac{1}{\frac{1}{\epsilon_{\text{liner}}} + \frac{1}{\epsilon_{\text{combustor}}}} - 1$$

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 Performance

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.
IMPINGING ELEMENT INJECTOR MANIFOLDING AND FLOW SCHEMATIC

(REF 2)
HEAT SINK CHAMBER
L* = 15 INCHES
(REF 2)
GO$_2$/GH$_2$ ENGINE START SEQUENCE AND PRESSURE HISTORY

(REF 2)
DYNAMIC RESPONSE - AEROJET GO\textsubscript{2}/GH\textsubscript{2} ENGINE
(COMPUTER PROGRAM RESULTS)

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

FIG. 27

COOLANT SIDE TEMP (T_w)

GAS-SIDE WALL TEMP (T_w)

COOLANT RISE TEMP

HEAT FLUX

AXIAL DISTANCE (IN)

AXIAL DISTANCE (IN)
TEMPERATURE VS LOCATION
AEROJET REGENERATIVELY COOLED THRUST CHAMBER
COMPARISON OF ANALYSIS
(REF 2)

AXIAL DISTANCE FROM INJECTOR FACE - INCHES
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


SYMBOLS

a  speed of sound (total conditions)
A  flow area (geometric)
A* thrust chamber throat area
AD mass flow admittance or heat transfer admittance
B  function of (4fL/D)
cf  skin friction coefficient
CF  thrust coefficient
cp  specific heat
Ci  wc of Ith node
C* pA*/m
Cd  orifice coefficient
D* thrust chamber throat diameter
D  diameter
f  Fanning friction coefficient
F  thrust
GI mass transfer parameter for Ith slot
G  mass transfer parameter for transpiration
h  static enthalpy of mixture
hi static enthalpy of Ith species
hi heat transfer coefficient for Ith slot
h* reference state enthalpy
H  total enthalpy
h  empirical constant
SYMBOLS (Continued)

\( \text{i} \)  
impulse

\( \text{I}_{\text{sp}} \)  
specific impulse

\( k \)  
thermal conductivity

\( k_{b,p} \)  
backward, equilibrium, and forward rate constants

\( k_{c,p} \)  
for the pth reaction

\( k_{t,p} \)  

\( L \)  
length

\( L_e \)  
Lewis number

\( M \)  
Mach number

\( \dot{m}_{\text{c}} \)  
coolant mass flow rate

\( \dot{m} \)  
thrust chamber mass flow rate

\( \text{MW} \)  
molecular weight

\( m \)  
percent oxygen

\( N_E \)  
number of injection elements

\( N_A \)  
number of annuli

\( p \)  
pressure

\( P_o \)  
stagnation pressure

\( \text{Pr} \)  
Prandtl number

\( q_L \)  
heat transfer rate from thrust chamber to wall

\( r \)  
radial coordinate

\( r_w \)  
local thrust chamber radius
SYMBOLS (Continued)

\[ R_c^I \] 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

\[ U^I \] \( \left( \frac{u_b}{u_c} \right)^I \)

\[ \dot{w} \] 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)

\[ x_I^S \]  \hspace{2cm} \text{axial coordinate of Ith slot}

\[ \alpha_i \]  \hspace{2cm} \text{mass fraction of ith species}

\[ \alpha_j \]  \hspace{2cm} \text{mass fraction of jth element}

\[ \alpha_i^* \]  \hspace{2cm} \text{reference state composition}

\[ \Delta F_{BL} \]  \hspace{2cm} \text{boundary layer loss}

\[ \eta^I \]  \hspace{2cm} \text{effectiveness parameter for Ith slot}

\[ \eta_D \]  \hspace{2cm} \text{divergence effectiveness}

\[ \mu \]  \hspace{2cm} \text{turbulent viscosity}

\[ \mu^* \]  \hspace{2cm} \text{molecular viscosity based on reference state properties}

\[ \overline{\mu} \]  \hspace{2cm} \text{configuration factor}

\[ \mu_b \]  \hspace{2cm} \text{molecular viscosity based on bulk properties}

\[ \mu_w \]  \hspace{2cm} \text{molecular viscosity based on wall conditions}

\[ \rho \]  \hspace{2cm} \text{density}

\[ \rho_0 \]  \hspace{2cm} \text{stagnation density}

\[ \rho^* \]  \hspace{2cm} \text{density based on reference state properties}

\[ \sigma^I \]  \hspace{2cm} \text{thermodynamic parameter}

\[ \Psi \]  \hspace{2cm} \text{stream function}

\[ \gamma \]  \hspace{2cm} \text{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)
\[ \Delta t \]    time increment
K     kinetic
S     shifting equilibrium,
F     frozen equilibrium

Superscripts

'     fuel rich region
''    oxidizer rich region
—     average
*     refers to throat
APPENDIX A

PROGRAM INPUT
<table>
<thead>
<tr>
<th>JOB NO</th>
<th>JOB NAME</th>
<th>NAME</th>
<th>EXTENSION</th>
<th>DEPT.</th>
<th>PAGE</th>
</tr>
</thead>
</table>

**CHARACTERIZATION INPUT**

<table>
<thead>
<tr>
<th>NDyna</th>
<th>NCom</th>
<th>NNe</th>
<th>NHeat</th>
<th>NCAS</th>
<th>NBLF</th>
<th>RUN CONTROL CARD (REQUIRED)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NNODE</th>
<th>NCOS</th>
<th>NITER</th>
<th>LPITF</th>
<th>NCREK</th>
<th>PCHER</th>
<th>PST</th>
<th>CD</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NCOMCU</th>
<th>RPOMCU</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NVOL</th>
<th>ICOMB</th>
<th>VOL</th>
<th>PRES</th>
<th>TEMP</th>
<th>RMDX</th>
</tr>
</thead>
</table>

OMIT IF NDyna = 0

"NNODE" OF THESE REQ'D

OMIT THESE TWO CARDS - IF NO COMBUSTION CALCULATIONS (NCOMBU = 0)
<table>
<thead>
<tr>
<th>UNIT #</th>
<th>TYPE</th>
<th>CAREA</th>
<th>DCOFF or DLEN</th>
<th>TIMON</th>
<th>TIMOFF</th>
<th>TIMOPN</th>
<th>TIMCLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SRTIME</th>
<th>SKTL</th>
<th>SPKF</th>
<th>SPKAP</th>
<th>SPARKP</th>
<th>SPARKE</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMBUSTION PROGRAM TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MPSI</th>
<th>IPRESS</th>
<th>ITURB</th>
<th>LW</th>
<th>NTYPE</th>
<th>ISOBAT</th>
<th>MB</th>
<th>ICHEM</th>
<th>MC</th>
<th>MG</th>
<th>LZ</th>
<th>MA</th>
<th>MY</th>
<th>MH</th>
<th>DBATY</th>
<th>NSLOT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRNT</th>
<th>XMAX</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### IBM 360 INPUT FORM - 80 COLUMN

**Form Type:** IBM 360 Input Form - 80 Column

**Page Dimensions:** 609.0 x 791.0

**Job No.:**

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

**Job Name:**

222

**Name:**

333

**Extension:**

444

**Dept.:**

555

**Page:**

666

#### TWX(1, J) - TWX(7, J)

<table>
<thead>
<tr>
<th>TWX(1, J)</th>
<th>TWX(2, J)</th>
<th>TWX(3, J)</th>
<th>TWX(4, J)</th>
<th>TWX(5, J)</th>
<th>TWX(6, J)</th>
<th>TWX(7, J)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Cards:** 4

**Omit If:**

- MB = 0
- ISOBAT ≠ 3
- or
- BBATY ≠ 3

**NSLOT**

- Omit if ISOBAT ≠ 4

**XS(1, K) - XS(12, K)**

<table>
<thead>
<tr>
<th>XS(1, K)</th>
<th>SH(1, K)</th>
<th>UC(1, K)</th>
<th>RUCF(1, K)</th>
<th>TCS(1, K)</th>
</tr>
</thead>
</table>

**Omit If ISOBAT ≠ 4**

**NSLOT Cards**

**RSTAR**

- Omit if ISOBAT ≠ 4

**IST**

**PRC**

**OMIT IF NINE ≠ 0**

**NELEM**

**DCAMB**

**TO**

**TH**

**EMR**

**Omit If NINE ≠ 0**

**WT**

**PO**

**Omit If NINE ≠ 0 OR NDYNA ≠ 1**

**T(1)**

**T(2)**

**T(3)**

**T(4)**

**Omit this card if NINE ≠ 1**

**MPSL Values**
INPUT VARIABLES
### $O_2/H_2$ Characterization Program Input Parameters

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDYNA</td>
<td>Driver</td>
<td>I 3</td>
<td>1 if dynamics calculation, 0 otherwise</td>
</tr>
<tr>
<td>NCOMBU</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if combustion calculation, 0 otherwise</td>
</tr>
<tr>
<td>NNJNE</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if injection model used, 0 otherwise</td>
</tr>
<tr>
<td>NHEAT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if heat transfer calculation, 0 otherwise</td>
</tr>
<tr>
<td>NNODE</td>
<td>Dynamics</td>
<td>I 4</td>
<td>No. of nodes (volumes)</td>
</tr>
<tr>
<td>NCONN</td>
<td>&quot;</td>
<td>&quot;</td>
<td>No. of connectors (orifices or ducts)</td>
</tr>
<tr>
<td>NITER</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Maximum no. of iterations</td>
</tr>
<tr>
<td>LPRTF</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Print frequency (print results every LPRTF iterations)</td>
</tr>
<tr>
<td>NCHEK</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Time is advanced to the first valve closing time when:</td>
</tr>
<tr>
<td>PCHEK</td>
<td>&quot;</td>
<td>E10.2</td>
<td>[ \frac{d(PRESS(NCHEK))}{d(TIME)} &lt; PCHEK \text{ (psia/ms)} ]</td>
</tr>
<tr>
<td>PSST</td>
<td>&quot;</td>
<td>&quot;</td>
<td>once (PRES(NCHEK) \geq PSST \text{ (psia)})</td>
</tr>
<tr>
<td>NCOMCL</td>
<td>&quot;</td>
<td>I 4</td>
<td>Call combustion every NCOMCL dynamics iterations</td>
</tr>
<tr>
<td>DPCOMC</td>
<td>&quot;</td>
<td>F10.2</td>
<td>Call combustion every time combustion pressure increases by DPCOMC (psia) * can use either or both</td>
</tr>
<tr>
<td>CD</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Nozzle coefficient</td>
</tr>
<tr>
<td>NOINJ(I)</td>
<td>&quot;</td>
<td>24 I 3</td>
<td>Volume numbers feeding injector face (used by injection model)</td>
</tr>
<tr>
<td>NVOL(I)</td>
<td>&quot;</td>
<td>I 3</td>
<td>Volume number (consecutive integers, (I = 1, \ldots, NNODE))</td>
</tr>
<tr>
<td>ICOMB(I)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>ICOMB(I) = 0 Neither pilot nor combustor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 Unlit combustor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 Lit combustor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 Unlit pilot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 Lit pilot</td>
</tr>
<tr>
<td>VOL(I)</td>
<td>&quot;</td>
<td>F10.0</td>
<td>Node volume (in$^3$)</td>
</tr>
<tr>
<td>PRES(I)</td>
<td>&quot;</td>
<td>F10.0</td>
<td>Initial pressure in Node I (psia)</td>
</tr>
</tbody>
</table>
### \( \text{O}_2/\text{H}_2 \) Characterization Program Input Parameters (Continued)

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP(I)</td>
<td>Dynamics</td>
<td>F10.0</td>
<td>Initial temperature in Node I (°R)</td>
</tr>
<tr>
<td>RMIX(I)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Initial Massfraction ( \text{O}_2 ) in Node I</td>
</tr>
<tr>
<td>IADMIT(J,I)</td>
<td>&quot;</td>
<td>I 3</td>
<td>Volume no. connected to admittance ( J ) ((I = 1, 2))</td>
</tr>
<tr>
<td>IRTYPE(J)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>IRTYPE(J) = 0: orifice ( I = 1 ) = 1: duct ( I = 0 ) = 2: throat</td>
</tr>
<tr>
<td>CAREA(J)</td>
<td>&quot;</td>
<td>F10.0</td>
<td>Full open cross sectional area (in²)</td>
</tr>
<tr>
<td>CCOEFF(J)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Discharge coefficient for ( I = 0, 2 )</td>
</tr>
<tr>
<td>DLEN(J)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Length (inches) for ( I = 1 )</td>
</tr>
<tr>
<td>TIMON(J)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Time valve starts to open (sec)</td>
</tr>
<tr>
<td>TIMOFF(J)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Time valve starts to close (sec)</td>
</tr>
<tr>
<td>TIMOPN(J)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Valve opening response time</td>
</tr>
<tr>
<td>TIMCLS(J)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Valve closing response time</td>
</tr>
<tr>
<td>SPTIME</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Time of initial spark (sec)</td>
</tr>
<tr>
<td>SPKTL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Time of last spark (sec)</td>
</tr>
<tr>
<td>SPKF</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Time between sparks (sec)</td>
</tr>
<tr>
<td>SPGAP</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Spark plug gap (inches)</td>
</tr>
<tr>
<td>SPARKP</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Spark plug potential (volts)</td>
</tr>
<tr>
<td>SPARKE</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Spark energy (millijoules)</td>
</tr>
<tr>
<td>DC</td>
<td>Combustion</td>
<td>12A6</td>
<td>Chamber diameter (inches)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Title card – any statement. Will be printed on each radial profile output.</td>
</tr>
<tr>
<td>MPSI</td>
<td>&quot;</td>
<td>I 5</td>
<td>Number of grid points at ( x = x ) initial.</td>
</tr>
<tr>
<td>IPRESS</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Number of grid points at ( x = 0 ), used in halving the grid, and program restarts at ( x &gt; 0 ).</td>
</tr>
<tr>
<td>ITURB</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 - Use Hersch viscosity model. 2 - Viscosity is Input</td>
</tr>
<tr>
<td>LW</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Specifies the laminar viscosity model used for the wall cooling modes. ( LW = 1 ) employs the Sutherland air viscosity model.</td>
</tr>
</tbody>
</table>

91
<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTYPE</td>
<td>Combustion</td>
<td>I 5</td>
<td>0 - Axisymmetric flow field</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Plane two-dimensional flow</td>
</tr>
<tr>
<td>ISOBAT</td>
<td></td>
<td></td>
<td>1 - Isoenergetic wall</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Regeneratively cooled wall</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Transpiration cooling</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 - Slot cooling</td>
</tr>
<tr>
<td>MB</td>
<td></td>
<td></td>
<td>0 - Free jet; P (x) prescribed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Ducted flow; P (x) prescribed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Ducted flow; Wall radius (x) prescribed</td>
</tr>
<tr>
<td>ICHEM</td>
<td></td>
<td></td>
<td>1 - Chemically frozen flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Equilibrium (&quot;complete combustion&quot;) chemistry</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Finite rate chemical kinetics</td>
</tr>
<tr>
<td>MC</td>
<td></td>
<td></td>
<td>Printout of the flow field radial profiles is made</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>every MC finite difference steps. The default is 10.</td>
</tr>
<tr>
<td>MG</td>
<td></td>
<td></td>
<td>Specifies the diluent specie used as a tracer in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>transpiration model. The diluent may also be present initially in the main stream flows, when other wall cooling models are used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Diluent is Nitrogen</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Diluent is Helium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Diluent is Argon</td>
</tr>
<tr>
<td>LZ</td>
<td></td>
<td></td>
<td>Printout dump controls for various portions of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>program. In general, nn = 0 means no dump; nn = 1 yields moderate dump; nn ≥ 2 yields overwhelming printout dump.</td>
</tr>
<tr>
<td>MA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISBATY</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Regenerative Cooling (ISOBAT = 2)**

1. Wall temperature, $T_w(x)$, (°R), specified. The wall heat transfer, $q_w(x)$ Btu/in² sec, is computed.

2. $q_w(x)$ specified. $T_w$ is computed.
<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISBATY (Cont.)</td>
<td>Combustion</td>
<td>I 5</td>
<td>Transpiration Cooling (ISOBAT = 3)</td>
</tr>
<tr>
<td>NSLOT</td>
<td></td>
<td></td>
<td>The number of slots in the slot wall cooling model.</td>
</tr>
<tr>
<td>PRNT</td>
<td></td>
<td>E10.8</td>
<td>Printout interval Δx (inches).</td>
</tr>
<tr>
<td>XMAX</td>
<td></td>
<td></td>
<td>Axial distance to which calculation is carried out, X_max (inches).</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td>Axial station at which calculation is begun, x_init (inches).</td>
</tr>
<tr>
<td>XLE</td>
<td></td>
<td></td>
<td>Turbulent Lewis number.</td>
</tr>
<tr>
<td>SIGMA</td>
<td></td>
<td></td>
<td>Turbulent Prandtl number.</td>
</tr>
<tr>
<td>DELPSI</td>
<td></td>
<td></td>
<td>The spacing between adjacent flow field grid points.</td>
</tr>
<tr>
<td>XMPS</td>
<td></td>
<td></td>
<td>Available for use in reducing the diffusion step size, Δx = Δx/XMPS. Let MPS = 1.</td>
</tr>
<tr>
<td>XK2</td>
<td></td>
<td></td>
<td>Constant employed in the turbulent viscosity models.</td>
</tr>
<tr>
<td>PSI(1)</td>
<td></td>
<td></td>
<td>Value of the flow field grid point, φ1, nearest the chamber centerline, with dimensions of ( \left( \frac{1 + NTYPE}{2} \right)^{\frac{1}{2}} \text{lb/sec} ).</td>
</tr>
</tbody>
</table>

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

\[ X \]
### \( \text{O}_2/\text{H}_2 \) Characterization Program Input Parameters

(Continued)

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUCX(I,J)</td>
<td>Combustion</td>
<td>7E10.8</td>
<td>Four sets of polynomial coefficients for ( \rho u ) ( \text{C} ) ((\text{lb/in}^2 \text{ sec})) (Transpiration cooling).</td>
</tr>
<tr>
<td>TCX(I,J)</td>
<td></td>
<td></td>
<td>Four sets of polynomial coefficients for ( T ) ( \text{C} ) ( (\text{oR}) ) (Transpiration cooling).</td>
</tr>
<tr>
<td>NSLOT</td>
<td></td>
<td>E10.8</td>
<td>Number of slots in the chamber wall ((\leq 21)).</td>
</tr>
<tr>
<td>XS(K)</td>
<td></td>
<td></td>
<td>Axial location of first slot ( (\text{inches}) ).</td>
</tr>
<tr>
<td>SH(K)</td>
<td></td>
<td></td>
<td>Height of first slot ( (\text{inches}) ).</td>
</tr>
<tr>
<td>UC(K)</td>
<td></td>
<td></td>
<td>Coolant velocity ( (\text{in/sec}) ).</td>
</tr>
<tr>
<td>RUCF(K)</td>
<td></td>
<td></td>
<td>Coolant mass flux ( (\text{lb/in}^2 \text{ sec}) ).</td>
</tr>
<tr>
<td>TCS(K)</td>
<td></td>
<td></td>
<td>Coolant temperature ( (\text{oR}) ).</td>
</tr>
<tr>
<td>RSTAR</td>
<td></td>
<td></td>
<td>Radius of curvature of engine throat ( (\text{inches}) ).</td>
</tr>
<tr>
<td>PST</td>
<td></td>
<td></td>
<td>Throat diameter ( (\text{inches}) ).</td>
</tr>
<tr>
<td>PRC</td>
<td></td>
<td></td>
<td>Coolant Prandtl number.</td>
</tr>
<tr>
<td>NELEM</td>
<td>Injection</td>
<td>I3</td>
<td>No. of injector elements</td>
</tr>
<tr>
<td>DCHAMB</td>
<td></td>
<td>F10.0</td>
<td>Chamber diameter at injector plane ( (\text{inches}) ).</td>
</tr>
<tr>
<td>TO</td>
<td></td>
<td></td>
<td>Oxygen total temperature ( (\text{oR}) ).</td>
</tr>
<tr>
<td>TH</td>
<td></td>
<td></td>
<td>Hydrogen total temperature ( (\text{oR}) ).</td>
</tr>
<tr>
<td>EMR</td>
<td></td>
<td></td>
<td>Rupe mixing factor</td>
</tr>
<tr>
<td>WT</td>
<td></td>
<td></td>
<td>Mass flow through injector ( (\text{lb/sec}) ).</td>
</tr>
<tr>
<td>PO</td>
<td></td>
<td></td>
<td>Chamber total pressure ( (\text{psia}) ).</td>
</tr>
<tr>
<td>FM</td>
<td></td>
<td></td>
<td>Mass fraction oxygen thru injector</td>
</tr>
<tr>
<td>T(I)</td>
<td>Combustion</td>
<td>E10.8</td>
<td>Static temperature ( (\text{oR}) ) at grid point 1</td>
</tr>
<tr>
<td>U(I)</td>
<td></td>
<td></td>
<td>Axial velocity ( (\text{in/sec}) ) at each grid point</td>
</tr>
<tr>
<td>FIX(I)</td>
<td></td>
<td></td>
<td>Chemistry time step ( (\text{seconds}) ) at each grid point</td>
</tr>
<tr>
<td>TELAP(I)</td>
<td></td>
<td></td>
<td>Integrated streamline residence times ( (\text{seconds}) ) for each grid point</td>
</tr>
<tr>
<td>YSPEC(1,I)</td>
<td></td>
<td></td>
<td>H mass fraction at grid point 1</td>
</tr>
<tr>
<td>YSPEC(2,I)</td>
<td></td>
<td></td>
<td>O mass fraction at grid point 1</td>
</tr>
<tr>
<td>YSPEC(3,I)</td>
<td></td>
<td></td>
<td>( \text{H}_2\text{O} ) mass fraction at grid point 1</td>
</tr>
</tbody>
</table>
### O$_2$/H$_2$ Characterization Program Input Parameters

(Continued)

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>YSPEC(4,I)</td>
<td>Combustion</td>
<td>E10.8</td>
<td>H$_2$ mass fraction at grid point I</td>
</tr>
<tr>
<td>YSPEC(5,I)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>O$_2$ mass fraction at grid point I</td>
</tr>
<tr>
<td>YSPEC(6,I)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>OH mass fraction at grid point I</td>
</tr>
<tr>
<td>YSPEC(7,I)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>HO$_2$ mass fraction at grid point I</td>
</tr>
<tr>
<td>YSPEC(8,I)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>H$_2$O$_2$ mass fraction at grid point I</td>
</tr>
<tr>
<td>YSPEC(9,I)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Diluent mass fraction at grid point I</td>
</tr>
<tr>
<td>RT</td>
<td>Performance</td>
<td>F10.0</td>
<td>Throat radius (inches)</td>
</tr>
<tr>
<td>RE</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Exit radius (inches)</td>
</tr>
<tr>
<td>XN</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Nozzle length (inches)</td>
</tr>
<tr>
<td>PERBEL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Percent bell</td>
</tr>
<tr>
<td>ENTHO</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Oxygen enthalpy (Btu/lb)</td>
</tr>
<tr>
<td>ENTHH</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Hydrogen enthalpy (Btu/lb)</td>
</tr>
<tr>
<td>OFINPT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Mixture ratio</td>
</tr>
<tr>
<td>PER</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Percent fuel injected in supersonic region</td>
</tr>
<tr>
<td>TTT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Temperature of fuel injected in supersonic region</td>
</tr>
<tr>
<td>NNODE</td>
<td>Heat Transfer</td>
<td>I 3</td>
<td>Number of nodes</td>
</tr>
<tr>
<td>NTYPFL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if transient, 0 if steady state</td>
</tr>
<tr>
<td>NFLMFL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if film cooling, 0 if not</td>
</tr>
<tr>
<td>NRGNFL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if regen cooling, 0 if not</td>
</tr>
<tr>
<td>NLFL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if liner, 0 if not</td>
</tr>
<tr>
<td>INJFL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1 if injector, 0 if not</td>
</tr>
<tr>
<td>TSTOP</td>
<td>&quot;</td>
<td>F12.0</td>
<td>Cut-off-time for transient case (sec)</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Number of iterations for steady state case (input as a real number)</td>
</tr>
<tr>
<td>PRINT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Print interval in seconds for transient and in number of iterations</td>
</tr>
<tr>
<td>INTERVAL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>for steady state</td>
</tr>
<tr>
<td>TWA LLF</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Initial wall temperature (°F)</td>
</tr>
</tbody>
</table>
### CHARACTERIZATION PROGRAM INPUT PARAMETERS

#### (Continued)

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSINKF</td>
<td>Heat Transfer</td>
<td>F12.0</td>
<td>Sink temperature (°F)</td>
</tr>
<tr>
<td>CAPPA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Conductivity (Btu/in sec °R)</td>
</tr>
<tr>
<td>EPS</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Wall emissivity (none)</td>
</tr>
<tr>
<td>PO</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Combustion chamber pressure (lb/in²)</td>
</tr>
<tr>
<td>OF</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Mixture ratio (none)</td>
</tr>
<tr>
<td>RC</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Radius of curvature at throat (inches)</td>
</tr>
<tr>
<td>RHO</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Density of wall material (lbs/in³)</td>
</tr>
<tr>
<td>CP</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Specific heat capacity of wall material (Btu/lb °R)</td>
</tr>
<tr>
<td>TINJH2</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Hydrogen injection temperature (°R)</td>
</tr>
<tr>
<td>TINJO2</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Oxygen injection temperature (°R)</td>
</tr>
<tr>
<td>X</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Axial distance of node (in)</td>
</tr>
<tr>
<td>DI</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Wall inner diameter (in)</td>
</tr>
<tr>
<td>DO</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Wall outer diameter (in)</td>
</tr>
<tr>
<td>NINJ</td>
<td>&quot;</td>
<td>I 3</td>
<td>Number of film injection stations</td>
</tr>
<tr>
<td>SLOT</td>
<td>&quot;</td>
<td>F12.0</td>
<td>Axial distance of film injection station (in)</td>
</tr>
<tr>
<td>HLSLOT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Slot height (in)</td>
</tr>
<tr>
<td>WCOOL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Coolant weight flow (lb/sec)</td>
</tr>
<tr>
<td>HRWD</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Hydrogen regen weight flow (lb/sec)</td>
</tr>
<tr>
<td>NPASS</td>
<td>&quot;</td>
<td>I 12</td>
<td>Number of regen passages</td>
</tr>
<tr>
<td>XREGEN</td>
<td>&quot;</td>
<td>F12.0</td>
<td>Regen injection station (in)</td>
</tr>
<tr>
<td>X</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Axial distance of node (in) (must correspond to nodes)</td>
</tr>
<tr>
<td>HPASS</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Regen passage height (in)</td>
</tr>
<tr>
<td>WPASS</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Regen passage width (in)</td>
</tr>
<tr>
<td>FIN EFFICIENCY</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Fin efficiency (none)</td>
</tr>
<tr>
<td>MASS FLOW</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Mass flow between liner and wall (lb/sec)</td>
</tr>
<tr>
<td>EMISSIVITY</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Liner emissivity (none)</td>
</tr>
<tr>
<td>LINER LENGTH</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Maximum axial distance of liner (in)</td>
</tr>
</tbody>
</table>
### O₂/H₂ CHARACTERIZATION PROGRAM INPUT PARAMETERS

(Continued)

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODEL</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Heat Transfer</td>
<td>F12.0</td>
<td>Axial distance of node (in) (must correspond to a node)</td>
</tr>
<tr>
<td>LINER ID</td>
<td></td>
<td></td>
<td>Liner inner diameter (in)</td>
</tr>
<tr>
<td>LINER OD</td>
<td></td>
<td></td>
<td>Liner outer diameter (in)</td>
</tr>
<tr>
<td>AINJS</td>
<td></td>
<td></td>
<td>Injector surface area (in²)</td>
</tr>
<tr>
<td>EMINJ</td>
<td></td>
<td></td>
<td>Injector emissivity</td>
</tr>
<tr>
<td>ARINJH</td>
<td></td>
<td></td>
<td>Injector H₂ Convection Area (in²)</td>
</tr>
<tr>
<td>HGINJH</td>
<td></td>
<td></td>
<td>Injector H₂ Convective Coefficient (Btu/in²/sec °R)</td>
</tr>
<tr>
<td>ARINJO</td>
<td></td>
<td></td>
<td>Injector O₂ Convection Area (in²)</td>
</tr>
<tr>
<td>HGINJO</td>
<td></td>
<td></td>
<td>Injector O₂ Convective Coefficient (Btu/in²/sec °R)</td>
</tr>
<tr>
<td>RESINJ</td>
<td></td>
<td></td>
<td>Injector - Combustor thermal resistance (sec°R/Btu)</td>
</tr>
<tr>
<td>OFINJ</td>
<td></td>
<td></td>
<td>Injector face heat flux (Btu/in² sec)</td>
</tr>
<tr>
<td>WTINJ</td>
<td></td>
<td></td>
<td>Injector weight (lbs)</td>
</tr>
<tr>
<td>CPINJ</td>
<td></td>
<td></td>
<td>Injector specific heat (Btu/lb °R)</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Aerojet H2/02 1500 LB Thrust Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 50 1 1 2</td>
<td>2 50 1 0 0 0 0 0</td>
</tr>
<tr>
<td>(5.900 \times 10^{4} )</td>
<td>( \frac{5.200}{10^{4} \text{C-01}} )</td>
</tr>
<tr>
<td>(1.000 \times 10^{5} \times 01)</td>
<td>( \frac{1.000}{10^{5} \text{C-01}} )</td>
</tr>
<tr>
<td>(1.000 \times 10^{5} \times 01)</td>
<td>( \frac{1.000}{10^{5} \text{C-01}} )</td>
</tr>
<tr>
<td>(3.000 \times 10^{5} \times 03)</td>
<td>( \frac{3.275}{10^{5} \text{C-01}} )</td>
</tr>
<tr>
<td>(1.825 \times 10^{5} \times 01)</td>
<td>( \frac{1.823}{10^{5} \text{C-01}} )</td>
</tr>
<tr>
<td>(3.244 \times 10^{5} \times 01)</td>
<td>( \frac{1.4267}{10^{5} \text{C-01}} )</td>
</tr>
<tr>
<td>(9.634 \times 10^{5} \times 00)</td>
<td>( \frac{5.5418}{10^{5} \text{C-01}} )</td>
</tr>
<tr>
<td>(5.297 \times 10^{2} \times 03)</td>
<td>( \frac{5.297}{10^{2} \text{C-01}} )</td>
</tr>
<tr>
<td>(5.297 \times 10^{2} \times 03)</td>
<td>( \frac{5.297}{10^{2} \text{C-01}} )</td>
</tr>
<tr>
<td>(5.297 \times 10^{2} \times 03)</td>
<td>( \frac{5.297}{10^{2} \text{C-01}} )</td>
</tr>
<tr>
<td>(5.297 \times 10^{2} \times 03)</td>
<td>( \frac{5.297}{10^{2} \text{C-01}} )</td>
</tr>
<tr>
<td>(5.297 \times 10^{2} \times 03)</td>
<td>( \frac{5.297}{10^{2} \text{C-01}} )</td>
</tr>
<tr>
<td>(5.297 \times 10^{2} \times 03)</td>
<td>( \frac{5.297}{10^{2} \text{C-01}} )</td>
</tr>
</tbody>
</table>

### Temperature (°C) Array

| \(1.5844788 \times 04\) | \( \frac{1.5844788}{10^{4} \text{C-01}} \) | \(1.5844788 \times 04\) | \(1.5844788 \times 04\) | \(1.5844788 \times 04\) | \(1.1029094 \times 04\) | \(1.1029094 \times 04\) | \(1.1029094 \times 04\) | \(1.1029094 \times 04\) |

### Velocity Profiles (in/s)

| \(0.002 \times 00\) | \( \frac{0.002}{10^{0} \text{C-01}} \) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) |

### Species Mass Fraction Profiles

| \(0.000 \times 00\) | \( \frac{0.000}{10^{0} \text{C-01}} \) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) | \(0.00000000\) |

### Other

<p>| (0.000 \times 00) | ( \frac{0.000}{10^{0} \text{C-01}} ) | (0.00000000) | (0.00000000) | (0.00000000) | (0.00000000) | (0.00000000) | (0.00000000) | (0.00000000) |</p>
<table>
<thead>
<tr>
<th>T</th>
<th>BAR</th>
<th>U</th>
<th>BAR</th>
<th>W</th>
<th>BAR</th>
<th>PH</th>
<th>BAR</th>
<th>GAMMA</th>
<th>BAR</th>
<th>ELEM</th>
<th>W</th>
<th>CPBAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.16010801</td>
<td>1.10446198</td>
<td>1.0494094040</td>
<td>1.2177211000</td>
<td>6.6717135001</td>
<td>1.9320636001</td>
<td>4.4524520203</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.4455535601</td>
<td>.1037915079</td>
<td>.1426806097</td>
<td>.3339999680</td>
<td>.7351637003</td>
<td>.2096872907</td>
<td>.3427666404</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELEM</td>
<td>1.16010801</td>
<td>1.0494094040</td>
<td>1.2177211000</td>
<td>6.6717135001</td>
<td>1.9320636001</td>
<td>4.4524520203</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMP</td>
<td>FNC</td>
<td>.5282228004</td>
<td>NET THRUST</td>
<td>.8732670004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**HULK MASS MEAN AVERAGED QUANTITIES**

<table>
<thead>
<tr>
<th>PT</th>
<th>H</th>
<th>0</th>
<th>W</th>
<th>H2O</th>
<th>PT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>2</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>3</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>4</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>5</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>6</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>7</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>8</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>9</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>10</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>11</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>12</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>13</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>14</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>15</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>16</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>17</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>18</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>19</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>20</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>21</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>22</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>23</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>24</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>25</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>26</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>27</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>28</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>29</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>30</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>31</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>32</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>33</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>34</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>35</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>36</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>37</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>38</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>39</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>40</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>41</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>42</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>43</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>44</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>45</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>46</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>
THE PERFORMANCE OF THE ENGINE AT TIME .000 SEC IS 386.48

THE THRUST IS 1387.96 LBS

THE MIXTURE RATIO IS 4.16
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

<table>
<thead>
<tr>
<th>NODE</th>
<th>NCORN</th>
<th>NITER</th>
<th>NCHEK</th>
<th>PCHEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>8</td>
<td>4000</td>
<td>1</td>
<td>1.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NVEL</th>
<th>ICORB</th>
<th>VOL</th>
<th>PRES</th>
<th>TEMP</th>
<th>RMIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>48.300</td>
<td>0.000</td>
<td>530.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>23.800</td>
<td>0.000</td>
<td>530.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>12.800</td>
<td>0.000</td>
<td>530.000</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.467</td>
<td>0.000</td>
<td>530.000</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>375.000</td>
<td>375.000</td>
<td>375.000</td>
<td>375.000</td>
</tr>
</tbody>
</table>

### DYNAMICS RESULTS

<table>
<thead>
<tr>
<th>TIME</th>
<th>0.026193</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ITERATION NO</th>
<th>460</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL NO</td>
<td>460</td>
</tr>
<tr>
<td>PRESSURE (PSIA)</td>
<td>5226.4</td>
</tr>
<tr>
<td>TEMP (R)</td>
<td>92.806</td>
</tr>
<tr>
<td>(CX)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DELT USED</th>
<th>0.00025</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLET</td>
<td>0.00</td>
</tr>
<tr>
<td>IMPULSE</td>
<td>0.00000</td>
</tr>
<tr>
<td>ISP</td>
<td>1.00</td>
</tr>
<tr>
<td>ACCUM VT FLOW</td>
<td>60300</td>
</tr>
<tr>
<td>VOL</td>
<td>1</td>
</tr>
<tr>
<td>PRES CHANGE</td>
<td>4.956PSI PER MS</td>
</tr>
</tbody>
</table>

---

**S-1220**
**DYNAMICS RESULTS**

TIME = 0.028671

<table>
<thead>
<tr>
<th>ITERATION NO</th>
<th>620</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL NC</td>
<td></td>
</tr>
<tr>
<td>PRESSURE</td>
<td></td>
</tr>
<tr>
<td>TEMP (R)</td>
<td></td>
</tr>
<tr>
<td>(PSIA)</td>
<td></td>
</tr>
<tr>
<td>1 (CCHBUSTOR)</td>
<td>17.16</td>
</tr>
<tr>
<td>2</td>
<td>.61</td>
</tr>
<tr>
<td>3</td>
<td>18.43</td>
</tr>
<tr>
<td>4</td>
<td>59.91</td>
</tr>
<tr>
<td>5 (PILOT)</td>
<td>295.66</td>
</tr>
<tr>
<td>6</td>
<td>374.59</td>
</tr>
<tr>
<td>7</td>
<td>374.59</td>
</tr>
</tbody>
</table>

DELTA LSED = 0.00025

THrust = .00

IMPULSE = .00000

ISP = .00

VOL 1 PRES CHANGE = 8.043 PSI PER MS

**DYNAMICS RESULTS**

TIME = 0.029144

<table>
<thead>
<tr>
<th>ITERATION NO</th>
<th>640</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL NC</td>
<td></td>
</tr>
<tr>
<td>PRESSURE</td>
<td></td>
</tr>
<tr>
<td>TEMP (R)</td>
<td></td>
</tr>
<tr>
<td>(PSIA)</td>
<td></td>
</tr>
<tr>
<td>1 (CCHBUSTOR)</td>
<td>22.08</td>
</tr>
<tr>
<td>2</td>
<td>.61</td>
</tr>
<tr>
<td>3</td>
<td>28.41</td>
</tr>
<tr>
<td>4</td>
<td>59.66</td>
</tr>
<tr>
<td>5 (PILOT)</td>
<td>295.65</td>
</tr>
<tr>
<td>6</td>
<td>374.49</td>
</tr>
<tr>
<td>7</td>
<td>374.49</td>
</tr>
</tbody>
</table>

DELTA LSED = 0.00025

THrust = .00

IMPULSE = .00000

ISP = .00

VOL 1 PRES CHANGE = 11.550 PSI PER MS

**DYNAMICS RESULTS**

TIME = 0.029656

<table>
<thead>
<tr>
<th>ITERATION NO</th>
<th>660</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL NC</td>
<td></td>
</tr>
<tr>
<td>PRESSURE</td>
<td></td>
</tr>
<tr>
<td>TEMP (R)</td>
<td></td>
</tr>
<tr>
<td>(PSIA)</td>
<td></td>
</tr>
<tr>
<td>1 (CCHBUSTOR)</td>
<td>29.64</td>
</tr>
<tr>
<td>2</td>
<td>.61</td>
</tr>
<tr>
<td>3</td>
<td>36.97</td>
</tr>
<tr>
<td>4</td>
<td>65.48</td>
</tr>
<tr>
<td>5 (PILOT)</td>
<td>295.95</td>
</tr>
<tr>
<td>6</td>
<td>374.98</td>
</tr>
<tr>
<td>7</td>
<td>374.59</td>
</tr>
</tbody>
</table>

DELTA LSED = 0.00025

THrust = .00

IMPULSE = .00000

ISP = .00

VOL 1 PRES CHANGE = 26.784 PSI 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.
<table>
<thead>
<tr>
<th>NODE</th>
<th>STATION</th>
<th>TD</th>
<th>OD</th>
<th>HEIGHT</th>
<th>WDOT</th>
<th>HEIGHT</th>
<th>WIDTH</th>
<th>ETA</th>
<th>F</th>
<th>ID</th>
<th>OD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>3.58</td>
<td>4.66</td>
<td>0.011</td>
<td>0.008</td>
<td>0.400</td>
<td>0.050</td>
<td>0.900</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>3.58</td>
<td>4.66</td>
<td>0.0</td>
<td>0.005</td>
<td>0.400</td>
<td>0.050</td>
<td>0.900</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.05</td>
<td>3.58</td>
<td>4.66</td>
<td>0.0</td>
<td>0.005</td>
<td>0.400</td>
<td>0.050</td>
<td>0.900</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1.30</td>
<td>3.58</td>
<td>4.66</td>
<td>0.0</td>
<td>0.005</td>
<td>0.400</td>
<td>0.050</td>
<td>0.900</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1.95</td>
<td>3.58</td>
<td>4.66</td>
<td>0.0</td>
<td>0.005</td>
<td>0.400</td>
<td>0.050</td>
<td>0.900</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2.60</td>
<td>3.58</td>
<td>4.66</td>
<td>0.0</td>
<td>0.005</td>
<td>0.400</td>
<td>0.050</td>
<td>0.900</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>3.25</td>
<td>3.46</td>
<td>4.49</td>
<td>0.0</td>
<td>0.005</td>
<td>0.330</td>
<td>0.050</td>
<td>0.870</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>3.90</td>
<td>3.11</td>
<td>3.97</td>
<td>0.0</td>
<td>0.005</td>
<td>0.280</td>
<td>0.050</td>
<td>0.850</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>4.55</td>
<td>2.95</td>
<td>3.17</td>
<td>0.0</td>
<td>0.005</td>
<td>0.240</td>
<td>0.050</td>
<td>0.830</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>5.20</td>
<td>2.12</td>
<td>2.81</td>
<td>0.0</td>
<td>0.005</td>
<td>0.200</td>
<td>0.050</td>
<td>0.800</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>5.80</td>
<td>1.86</td>
<td>2.99</td>
<td>0.0</td>
<td>0.005</td>
<td>0.170</td>
<td>0.050</td>
<td>0.800</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>6.88</td>
<td>3.02</td>
<td>3.85</td>
<td>0.0</td>
<td>0.005</td>
<td>0.250</td>
<td>0.050</td>
<td>0.820</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>7.66</td>
<td>4.40</td>
<td>5.59</td>
<td>0.0</td>
<td>0.005</td>
<td>0.390</td>
<td>0.050</td>
<td>0.880</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>9.04</td>
<td>5.53</td>
<td>6.65</td>
<td>0.001</td>
<td>0.094</td>
<td>0.460</td>
<td>0.050</td>
<td>0.960</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>9.65</td>
<td>5.54</td>
<td>6.65</td>
<td>0.0</td>
<td>0.005</td>
<td>0.420</td>
<td>0.050</td>
<td>0.960</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>10.12</td>
<td>6.57</td>
<td>6.67</td>
<td>0.0</td>
<td>0.005</td>
<td>0.380</td>
<td>0.050</td>
<td>0.960</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>11.20</td>
<td>7.43</td>
<td>7.53</td>
<td>0.0</td>
<td>0.005</td>
<td>0.340</td>
<td>0.050</td>
<td>0.960</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>12.27</td>
<td>8.28</td>
<td>8.38</td>
<td>0.0</td>
<td>0.005</td>
<td>0.300</td>
<td>0.050</td>
<td>0.960</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>13.28</td>
<td>8.29</td>
<td>8.39</td>
<td>0.0</td>
<td>0.005</td>
<td>0.260</td>
<td>0.050</td>
<td>0.960</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
## --ROCKET HEAT TRANSFER MODEL-- AEROJET REGENERATIVELY COOLED C2/H2 ENGINE

### Iteration No. Core O/F Core Temp WT. Flow

<table>
<thead>
<tr>
<th>NODE</th>
<th>STATION</th>
<th>T WALL</th>
<th>T FILM</th>
<th>T BULK</th>
<th>T LINER</th>
<th>T H2 LINER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>141.5</td>
<td>141.4</td>
<td>141.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>141.5</td>
<td>144.3</td>
<td>141.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.65</td>
<td>144.3</td>
<td>308.2</td>
<td>141.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>1.30</td>
<td>147.1</td>
<td>473.9</td>
<td>140.8</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>1.95</td>
<td>149.5</td>
<td>633.9</td>
<td>139.6</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>2.60</td>
<td>151.6</td>
<td>791.0</td>
<td>137.7</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>3.25</td>
<td>155.4</td>
<td>445.5</td>
<td>135.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>3.90</td>
<td>162.3</td>
<td>1133.5</td>
<td>132.8</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>4.55</td>
<td>174.9</td>
<td>1207.5</td>
<td>129.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>5.20</td>
<td>193.2</td>
<td>1503.0</td>
<td>125.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### --ROCKET HEAT TRANSFER MODEL-- AEROJET REGENERATIVELY COOLED C2/H2 ENGINE

### Iteration No. Core O/F Core Temp WT. Flow

<table>
<thead>
<tr>
<th>NODE</th>
<th>STATION</th>
<th>T WALL</th>
<th>T FILM</th>
<th>T BULK</th>
<th>T LINER</th>
<th>T H2 LINER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>140.5</td>
<td>140.5</td>
<td>140.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>140.5</td>
<td>143.5</td>
<td>141.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.65</td>
<td>143.8</td>
<td>307.2</td>
<td>140.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>1.30</td>
<td>146.7</td>
<td>473.6</td>
<td>140.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>1.95</td>
<td>149.1</td>
<td>633.6</td>
<td>139.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>2.60</td>
<td>151.2</td>
<td>791.8</td>
<td>137.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>3.25</td>
<td>155.0</td>
<td>448.5</td>
<td>135.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>3.90</td>
<td>161.5</td>
<td>1110.4</td>
<td>132.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>4.55</td>
<td>174.9</td>
<td>1207.5</td>
<td>129.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>5.20</td>
<td>193.2</td>
<td>1503.0</td>
<td>125.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

---

116
---ROCKET HEAT TRANSFER MODEL---  AEROJET REGENERATIVELY CCELED C2/H2 ENGINE

**ITERATION NO.**  CORE ON/OFF  CORE TEMP  WT. FLUX

<table>
<thead>
<tr>
<th>NODE</th>
<th>STATION</th>
<th>T WALL</th>
<th>T FILM</th>
<th>T BULK</th>
<th>T LINER</th>
<th>T H2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.0</td>
<td>141.1</td>
<td>141.1</td>
<td>141.2</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.01</td>
<td>141.1</td>
<td>143.2</td>
<td>141.2</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.65</td>
<td>144.1</td>
<td>308.0</td>
<td>141.2</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>1.30</td>
<td>146.9</td>
<td>472.9</td>
<td>140.6</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>1.95</td>
<td>149.3</td>
<td>633.8</td>
<td>139.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>2.60</td>
<td>151.4</td>
<td>750.9</td>
<td>137.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>3.25</td>
<td>155.2</td>
<td>646.6</td>
<td>135.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>3.90</td>
<td>162.1</td>
<td>1110.5</td>
<td>132.6</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>4.55</td>
<td>174.7</td>
<td>1297.5</td>
<td>129.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>5.20</td>
<td>193.0</td>
<td>1500.0</td>
<td>125.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11</td>
<td>5.80</td>
<td>157.8</td>
<td>1699.8</td>
<td>118.4</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>12</td>
<td>6.58</td>
<td>192.6</td>
<td>2017.1</td>
<td>197.7</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>13</td>
<td>7.96</td>
<td>148.1</td>
<td>2276.4</td>
<td>97.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>14</td>
<td>9.04</td>
<td>157.8</td>
<td>2462.1</td>
<td>70.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>15</td>
<td>9.05</td>
<td>158.0</td>
<td>143.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>16</td>
<td>10.12</td>
<td>207.7</td>
<td>327.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>17</td>
<td>11.20</td>
<td>327.4</td>
<td>500.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>18</td>
<td>12.27</td>
<td>262.3</td>
<td>661.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>19</td>
<td>12.28</td>
<td>262.3</td>
<td>696.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
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), \( \frac{dP}{dx} \) 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 \( \alpha^j \) or \( \bar{\alpha}^k \)) 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 (\( h, T, C_p, \rho, w, \) 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 α^j (x, ψ).

SUBROUTINE DIV, BL, SIDEI

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 ($\hat{\alpha}(x, \psi)$) for $H$ and $O$.

**SUBROUTINE FFCF**

Subroutine FFCF is used to compute duct friction coefficient given the flow Reynolds'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 \( \text{GO}_2/\text{GH}_2 \) 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 \( \alpha^j \) or \( \alpha^k \)). 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 \( \text{KOPT} = 1 \), \( C_p(x, \psi), h(x, \psi), \) and \( H(x, \psi) \) are determined from \( T(x, \psi), u(x, \psi) \) and \( \alpha^j(x, \psi) \). For \( \text{KOPT} = 2 \), \( T(x, \psi), C_p(x, \psi), \) 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 \( \text{O}_2 \) and \( \text{H}_2 \).
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. Isp, Thrust, and CF 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, 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 CO_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

\[ Δx = \min \left( \frac{(\Delta ψ)^2}{4\mu Sc} \right) \left( \frac{\psi_{m,n}(\Delta ψ)^2}{(b)} \left( \frac{1}{Sc_{n,m+1/2}} + \frac{1}{Sc_{n,m-1/2}} \right) \right) \left( r_{n,m+1} - r_{n,m} \right) \]

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:

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

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
SUBROUTINE AREA

Is this the first axial step in the computation?

YES: NUMBER = 0
LT = 40
LU = 20
LX = 20

NO: Initialize parameters for the computation of \( \frac{dY}{dx} \) from the chamber mass flow, area, and flow field properties.

Call "PRESS" to evaluate the wall radius, \( r_w \), \( (n+1) \), from the inputted polynomials.

(Do Loop)
Compute \( \frac{dP}{dx} \).

DO LOOP
M = 1 to LT

PNEW is too large; readjust the iteration procedure

DO LOOP
M = 1 to LT

Is PNEW > .0 ?

YES: Use first guess for \( \frac{dP}{dx} \)

\[ \begin{align*}
\frac{dP}{dx} &= \frac{P_{NEW} - P_n}{dx_{n+1}} \\
HDPDX &= \frac{dP}{dx_{n+1}}
\end{align*} \]

NO: WRITE
YW, DPDX, PNEW, P, DX, X

DO LOOP
M = 1 to LT

Is LZ ≠ 0 ?

YES: Write
IOUT, I, LR, MPSI, KAT, PNEW, RU(I), RUR(I), DX, DPDX, RUT(I)

NO: WRITE PFRD = PNEW

Is PNEW < PFRD ?

YES: PNEW = .9 *PFRD

NO: Is UFAC < u n

YES: Is still large enough to cause numerical errors in momentum computation - readjust:

\[ \begin{align*}
UFAC &= \frac{dx \frac{dP}{dx_{n+1}}}{(\rho u)_n} \\
PFAC &= .1 \times u_n \frac{dP}{dx_{n+1}}
\end{align*} \]

NO: \( \frac{dP}{dx_{n+1}} < .0 \)

YES: PFAC = -PFAC

NO: PFAC = \( -PFAC \)

KAT = KAT + 1

132
SUBROUTINE AREA (Cont.)

Does the wall still lie above the grid? 

YES

NO

WRITE PNEW, RU(I), U(I), FX DPDX, RUT(I), UFAC, PFAC

WRITE PNEW, RU(I), U(I), PX DPDX, RUT(I), UFAC, PFAC

KAT = KAT + 1

Is KAT > 10?

NO

YES

SMALLH(I) = H(I) - (U(I)/2)

RU(I) = u(I) dx (dp1(I)/dx)(n+1)

RHOZ(I) = P1(I)/R(i) - [n(I)]/W

MPSI = MPSI - wMPsi

RNMP = RNMP - (dn+1(I)dx)

FALPA = FALPA + FNEW

DO LOOP

I = 1 to LR

INCREMENT I

Is the loop complete?

YES

NO

DO Loop

Set AP(I) = A(I,n)

DO Loop

CALL "INVERT" to determine

1 = f (h(I), a(I))

Is Tj < 0?

YES

NO

RHOZ(I) = P1(I)/R(i) - [n(I)]/W

Does LT = 0?

YES

NO

Does the wall still lie above the grid?

YES

NO

WRITE RT, RHOZ, YZ, RU for top point of grid

YWP = YZ(NPSI)

YWP = MPSI + (MPSI - MPSI)

YWP = MPSI - MPSI

ALPHA = 1

FALPA = ALPA + FNEW

FYWP = yW

Is ALPA > ALPA?

YES

NO

PAY(I) = p(I) - 1

PAY(2) = yW - 1

PAY(3) = yW - 1

PAY(4) = PAY(1)*PAY(2)/PAY(3)

Is this the first iteration on yW?

YES

NO

IND = 9

Is ALPA < 0.001?

YES

NO

δ = 1 x δ

δ = 0.1 x δ
SUBROUTINE BEGIN

CNFSMS and CNKCJL are units conversion factors for converting \( \text{ft}^2 \to \text{m}^2 \)
\( \text{sec}^2 \to \text{sec}^2 \)
and kilocalories to joules
\( \text{gm} \cdot \text{mole} \to \text{kgm} \cdot \text{mole} \)

Information is stored in DATA Statements for
\( \gamma_j \) (GAM), 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 (HF) for each species

Set diluent element and specie molecular weight, thermodynamic fits and \( \gamma_j \) to inputted choice of \( \text{N}_2, \text{H}_2 \), or \( \text{A} \)

(DO Loop)
Compute \( \Gamma_{ij} \) (GAM) from \( \gamma_j \) 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 (\( = \text{m}^7/\text{sec}^2 \))

Is \( \text{MA} < 1 \) ?
Yes WRITE HF
No RETURN
SUBROUTINE BLOW

- Calculate blowdown pseudo-volume
- Calculate mixed gas temperature
- Determine total mass, mixture, mol wt, heat content of gas remaining in engine
- Calculate pressure, thrust, accumulated impulse
- Time = TIME + 0.0005
- Calculate Isp for entire pulse
- Pressure < 0.5 psi?
- Print Results
- Return to MAIN

From MAIN
SUBROUTINE BULK

Is the flow field axisymmetric?

YES

EMD = \frac{\psi}{w}

(DO Loop) EMDY(J) = \frac{\psi}{w}

NO

EMD = \psi

(DO Loop) EMDT(I) = \psi

Initialize parameters to prepare for integration across the chamber width to compute bulk values for the independent variables u, H, and \sigma^k or \sigma^k^.

BARA = \frac{m_i}{m_w}

BARB = 1 - \frac{m_i}{m_w}

IM = I - 1

BARC = \Delta m

(\text{DO Loop}) I = 2 to LR

(IM = I - 1)

Barc = \Delta m

(\text{DO Loop}) I = 2 to LR

UBL = \frac{1}{I} \left( U_1 + U_{IM} \right) \Delta m

HBL = \frac{1}{I} \left( H_1 + H_{IM} \right) \Delta m

Is equilibrium chemistry being used?

NO

(\text{DO Loop}) SBL(J) = \sum I (\sigma^j_1 + \sigma^j_{IM} \Delta m)

NO

(\text{DO Loop}) EBL(K) = \sum I (\sigma^k_1 + \sigma^k_{IM} \Delta m)

(\text{DO Loop}) SBL(J)

Is the loop on I complete?

YES

Increment I

YES

NO

(\text{DO Loop}) \sigma^j = \sum I \frac{U_j}{U}

(\text{DO Loop}) \sigma^j = \sum I \frac{H_j}{H}

Is equilibrium chemistry being used?

NO

Call "DENSE" to evaluate \( \sigma^j \), \( \sigma^j \), and \( W \) from \( P, T, \) and \( \sigma^j \).

YES

Call "EQUILC" To compute bulk specie mass fractions, \( \sigma^j \), from the bulk element mass fractions \( \sigma^k \)
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 ($DF$)

PRINT RESULTS

Subtract Loss DP, DVL from Theoretical F & Isp

Call Subroutine DIV to Find Percent Isp Loss due to Divergence (DVL)

IS PER > .01

YES

Calculate Additional $F$ and Change in ISP due to Mass Addition

NO

RETURN

PRINT OUTPUT
SUBROUTINE COMB

CALL "DENSE" to Compute \( p, p_u, \) and \( W \) for all Grid Points

CALL "HEAT" to Compute the Enthalpy, \( h = h(T,\alpha) \) for all Grid Points

Is there only One Grid Point?

Compute \( \psi \) and \( y \) for 1-D Flow

(DO Loop)

Store Edge Values of \( \tau_1, \Delta t, \dot{h}, \alpha^1, \tau_1, T_1, \) in Unused Streamlines above the Computational Grid, to Provide for Future Expansion of the Grid

(Do Loop)

Initialize \( \omega^1, h^1 \)

Compute \( \phi_w \), the Wall Streamline Coordinate

WRITE \( \psi_w, P, \frac{dP}{dx}, y_w, \frac{dy_w}{dx} \)

CALL "WALL" to determine Initial Boundary Conditions

CALL "VISC" to find \( \mu_t \)

Does the Wall lie above the Grid?

YES

Initialize Coolant Composition:

\[
\begin{align*}
\alpha_c^0 &= 1 - \alpha_c^t \\
\alpha_c &= 1 - \alpha_c^t \\
\alpha_c^t &= 0 \\
\end{align*}
\]

CALL "MARCH" to find Computation to \( \psi \)_{max}

Is \( \psi_w \leq \psi_{NPS1-1} \)?

YES

TO MAIN

NO

2

TO MAIN
SUBROUTINE CONSRV

Is there more than one grid point?

NO  DLS = Δ υ^2
SCH = Pr/Lc
EX1 = Δ υ / Δ x

DO Loop
I = 2 to NPSI

a
IP = I + 1
IM = I - 1

YES

Is the flow field axisymmetric?

NO

YES

Solve Conservation of Momentum equation for μ_n+1,1 (RU (0))

EX11 = b_n,1 + 1/2
EX12 = b_n,1 - 1/2

EX1 = Δ υ / Δ x

EX8 = b_n,i / Pr
EX9 = b_n,i+1/2 / Pr
EX10 = b_n,i-1/2 / Pr
EX14 = 0.5 Δ υ^2 / Pr

EX3 = b_n,i / Sc
EX5 = b_n,i+1/2 / Sc
EX7 = b_n,i-1/2 / Sc
EX13 = 0.5 Δ υ^2 - b_n,i+1/2 - b_n,i-1/2

Δ x / Sc
Δ x / Sc

EX20 = b_n,i + 1/2 (1 - 1/Pr)^2
EX21 = b_n,i - 1/2 (1 - 1/Pr)^2
EX22 = b_n,i + 1/2 (1 - 1/Pr)^2
EX23 = b_n,i - 1/2 (1 - 1/Pr)^2

EX24 = (b_n,i + 1/2 + b_n,i - 1/2)(1 - 1/Pr)^2
EX25 = EX22 - EX24 + EX23

EX25 = 0

EX30 = b_n,i+1/2 (1 - 1/Pr)
EX31 = b_n,i-1/2 (1 - 1/Pr)

(DO Loop)
EX32 = P b_n+1,i
EX33 = P b_n,i+1
EX34 = P b_n,i-1
EX35 = P b_n,i

EX36 = 1/2 EX30 [b_n+1,i + EX32]
EX37 = 1/2 [EX30 b_n,i + EX33]
EX38 = 1/2 EX31 [b_n,i-1 + EX35]
EX39 = EX36 - EX37 + EX38

YES

Le < .001

NO

EX39 = 0

YES

EX39 = 0
SUBROUTINE CONSRV (Cont.)

Solve Conservation of Energy Equation for \( H_{n+1,1} \) (RH(1))

Is Equilibrium Chemistry being used?

YES

Solve Diffusion Conservation Equation in terms of species, \( \alpha_j^n \) \( RYSPEC(j, l) \), including chemical kinetics effects, if any.

DO Loop

NO

Is the Loop on I complete?

YES

INCREMENT I

EX51 = 0.0

NO

EX50 = \( \frac{h_j}{Pr} \) \( \alpha_j \)

EX51 = \( \frac{Le - 1}{Pr} \) (EX50 - h_j)

Solve Energy Conservation Equation on the Centerline for \( H_{n+1,1} \) (RH(1))

Solve Momentum Conservation Equation on the Centerline for \( H_{n+1,1} \) (RH(1))

EX16 = \( \mu_1 \Delta x \frac{\Delta u}{(\Delta \phi)^2} \)

EX16 = \( \mu_1 \Delta x \frac{\Delta u}{(\Delta \phi)^2} \)

Solve Momentum Conservation Equation on the Centerline for \( H_{n+1,1} \) (RH(1))

Pr - 1, | < 0.01

EX40 = \( (1 - \frac{1}{Pr}) \frac{2 \frac{u_1^2}{2} - \frac{u_1^2}{2}}{2} \)

EX40 = 0

Is the Flow field axisymmetric?

YES

NO

EX50 = \( \frac{h_j}{Pr} \) \( \alpha_j \)

EX51 = \( \frac{Le - 1}{Pr} \) (EX50 - h_j)

EX51 = 0.0

Is \( u_j \geq \Delta \phi \)?

YES

NO

Solve Diffusion Conservation Equation in terms of elements, \( \alpha_j^n = 1,1 \) (RALHA(j, l))

EX50 = \( \frac{h_j}{Pr} \) \( \alpha_j \)

EX51 = \( \frac{Le - 1}{Pr} \) (EX50 - h_j)

NO

EX51 = 0.0

Is \( u_j \geq \Delta \phi \)?

YES

NO

EX50 = \( \frac{h_j}{Pr} \) \( \alpha_j \)

EX51 = \( \frac{Le - 1}{Pr} \) (EX50 - h_j)
SUBROUTINE CONSRV (Cont.)

(DO Loop)
Solve Diffusion Conservation Equation on the Centerline in Terms of species, \( \dot{\alpha}^j \), including Chemical kinetic effects, if any

(YES)
Specie Diffusion Equation
(DO Loop)
\[ \dot{\alpha}^j_{n+1,1} = \dot{\alpha}^j_{n,1} + \dot{\psi}^j_{n,1} \]

(YES)
Element Diffusion Equation
(DO Loop)
\[ \frac{\alpha^k}{\alpha^k_{n+1,1}} = \frac{\alpha^k}{\alpha^k_{n,1}} \]

(Momentum Equation)
\[ u_{n+1,1} = u_{n,1} \]

(Energy Equation)
\[ H_{n+1,1} = H_{n,1} \]

(Conservation Equations for Outer Edge of Flow Field):
\[ u_{n+1,1,\text{MPsi}} = u_{n,\text{MPsi}} \]
\[ H_{n+1,1,\text{MPsi}} = H_{n,\text{MPsi}} \]

(Element Diffusion Equation)
(DO Loop)
\[ \frac{-k}{\alpha^k_{n+1,1,\text{MPsi}}} = \frac{j_k}{\alpha^k_{n,1,\text{MPsi}}} \]

(YES)
Specie Diffusion Equation
(DO Loop)
\[ \dot{\alpha}^j_{n+1,1,\text{MPsi}} = \dot{\alpha}^j_{n,1,\text{MPsi}} + \dot{\psi}^j_{n,1,\text{MPsi}} \]

(YES)
Element Diffusion Equation
(DO Loop)
\[ \frac{\alpha^k}{\alpha^k_{n+1,1,\text{MPsi}}} = \frac{\alpha^k}{\alpha^k_{n,1,\text{MPsi}}} \]

(DO Loop)
Write
\[ RU(0), U(0)R(0), H(0), TAUT(0), T(0), RT(0), SMALLH(0) \]

(YES)
RETURN

(NO)
RETURN
SUBROUTINE CRSPLT

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

CONTINUE

NO

Intersection ?

YES

Set AFRZ = 5.0

CONTINUE

DO LOOP

RETURN
SUBROUTINE DENSE (JR, JS)

JRR = JR
JSS = JS

DO Loop
I = JRR to JSS

DUM = 0

(Do Loop)
Compute
DUM = \sum_{j=1}^{all\ gases} \left( \frac{\alpha_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(P₁) and ISPK(P₂) to find SIK(I) (Percent Recombination for I\textsuperscript{th} Streamtube)

Convert SIK(I) to Specific Impulse

RETURN
SUBROUTINE DYNAM

FROM MAIN

TIME = 0.0 ?

Artificial time advance flag = 1 ?

YES

NO

TIME = TIME + .00005

YES

NO

Zero Parameters

Determine Combustor & Pilot Nodes

Calculate Mass and Heat Content of Volumes

Is TIMON(J) > TIME ?

Are TIMON(J), TIMOFF(J), TIMOPN(J), TIMCLS(J) all 0 ?

Loop thru Connectors J = 1

ADD = 0.0

ADD = 1.0

1002

1001

500

146
SUBROUTINE DYNAM (Cont.)

1001

Is
TIME > TIMON(J) + TIMOPN(J)
YES
NO

ADD = TIME - TIMON(J)
TIMOPN

NO

Is
TIME < TIMOFF(J)
YES
NO

ADD = 1.0

ADD = 0.0

Is
TIME > TIMCLS(J) + TIMOFF(J)
YES
NO

ADD = 1.0 - (TIME - TIMOFF(J))
TIMCLS(J)

1002
SUBROUTINE DYNAM (Cont.)

Area of orifice J is ADD times input cross sectional area, CAREA

Calculate pressure ratio and determine driving volume, NDX3, and receiving volume, NDX4

Set flow and admittance equal 0

Is \((PRES(NDX3) - PRES(NDX4)) < 0.001\)?

Calculate unchoked admittance, ADMIT, and Mass Flow Rate, WDOT

Call FMOLWT & SPHEAT to determine hot gas mol. wt. and \(\gamma\)

Calculate choked admittance, ADMIT, and Mass Flow Rate, WDOT

Is orifice choked?

Is \(J\) an orifice?

Calculate cold gas molecular wt. and \(\gamma\) and gas constant

1002

375

350
SUBROUTINE DYNAM (Cont.)

Calculate Reynolds No., REN

Estimate (4FL/D), Set K = 1

CALL CPRF Is duct choked?

YES

Calculate unehked admittance, mass
flow (CALL FLOW)

Calculate choked admittance, mass
flow (CALL CFLOW)

REn < 2100 (laminar flow)?

YES

Calculate laminar friction factor

Determine turbulent friction factor
(CALL FFCF)

K = 3?

YES

Use New Friction Factor K = K + 1

NO

K = 3

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

Set flow rate from the combustor and accumulated flow

Is NDX3 the Combustor?

YES NO
SUBROUTINE DYNAM (Cont.)

Set heat flow rate, mass flow rate, and sum of admittances for NDX3 & NDX4

J = number of connectors?

YES

Set new mass content, new mixture, new heat content for Volume I

Loop thru volumes I = 1

Is volume Ignited?

YES

Determine new temperature (CALL TGR) and new mol wt (CALL FMOLWT)

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?
SUBROUTINE DYNAM (Cont.)

1004

Is time constant, DELT less than stable time, CHECK

YES

NO

Set DELT = .9 CHECK

600

DELT < .8 CHECK

YES

NO

Is DELT > $10^{-7}$ or is this a spark arrival time?

YES

Set artificial time advance flag to 1

NO

Set TIME = TIME + DELT

Is this a spark arrival time?

YES

NO

Set DELT = .8 CHECK

600

1005

Set spark arrival flag, reset DELT = (SPTIME - TIME), save old DELT

Is next spark arrival time, SPTIME, > TIME?

YES

NO
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 there a pilot?

NO

Is the chamber already been lit?

YES

Has the chamber already been lit?

NO

Is this a spark arrival time?

YES

Is this a spark arrival time?

NO

Is pilot lit but chamber not?

YES

Set plug pressure & mixture to combustor conditions

Set plug pressure & mixture to pilot conditions

Write ignition failure message

Write spark failure message

Is there sufficient potential for a spark to occur?

YES

Write spark failure message

Is the spark energy sufficient for ignition?

YES

1006

1007
SUBROUTINE DYNAM (Cont.)

1007

Calculate new pressure & temp. for pilot or combustor & write results

Is there a pilot?

Yes

1006

Does the pilot flame quench in combustor?

Yes

NO

Set next spark time and set spark flag, NSPK, to 0

Set combustor flag to lite, (ICOMB(ICVOL)) = 2

NO

Calculate pressure change per millisecond in volume NCHEK

RETURN TO MAIN

NSPK = 1
SUBROUTINE EQUILC (LAA, LBB)

Compute the element concentration array EW(K)

Set the specie concentration array, AW(J), equal to zero

Write printout dump for the subroutines

Is the loop on I complete?

Increment I

Return

Are there any other species besides the diluent? NO

Is there a surplus of $H_2$?

YES

Compute the concentrations of the specie $H_2O$ and $O_2$, $AW(NHTO)$ and $AW(NOT)$

Compute the concentrations of the specie $H_2O$ and $H_2$, $AW(NHTO)$ and $AW(NHT)$

Compute specie mass fractions, YSPEC(J,I) from concentrations and molecular weights

Is $MA > 0$?

YES

NO

DO Loop

LCC = LAA
LDD = LBB

DO Loop

1 - LCC to LDD

(RDO Loop)

Compute the element concentration array EW(K)

Compute the Concentration for the diluent specie AW(NNT)

Compute the Concentration for the diluent specie AW(NNT)
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

Is the Loop on I complete?

YES

Is MH > 0?

NO

WRITE FRT

NO

YES

Compute the Gibbs Free Energy, FRT(I) for each specie at Temperature, T

(DO Loop) Compute X = \( \sum_{i=1}^{n} \frac{\alpha_i}{W_i} \)

Compute the forward and backward reaction rates, QK(I,1) and QK(I,2) for all reactions.

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

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

In this subroutine \( A_{ij} \) and \( B_i \) are evaluated for \((T, \rho, t_o, y_o)\)

\[
PL(j, \ell) = \frac{\partial}{\partial y_j} \left[ k_j(T) \rho^{-1} \prod_{\ell=1}^{n} (\rho y_\ell) \nu'_j \right]
\]

\[
A(i, \ell) = \sum_{j=1}^{m} (\nu''_{ij} - \nu'_i) PL(j, \ell)
\]

\[
B(i) = \sum_{j=1}^{m} (\nu''_{ij} - \nu'_j) k_j(T) \rho^{-1} \prod_{\ell=1}^{n} (\rho y_\ell) \nu'_j
\]

RETURN
SUBROUTINE FLUX

Are adding a grid point to the flow field, whose physical properties are those of the wall point. Set TELAPI, U(I), H(I), T(I) equal to wall values.

DO Loop
I = MPSI to MP

Is DEER < \Delta

NO

YES

DEER = PSIWA - PSI(LR)

Set ALPHA (J,I) equal to wall values

Is equilibrium chemistry being used?

YES

(NO Loop) Set YSPEC(I,J) equal to wall values

INCREMENT I

NO

Is the loop complete?

YES

NPSI = NPSI + 1

NNS = NSLCH + 1

Is NNS > NSLOT?

NO

Call "HOOT" to obtain coolant enthalphy for the NNSth slot

Is X < XS (NNS)?

YES

200

NO

200

DPS = RUCF (NNS) x SH(NNS)

Is NTYPE = 0?

NO

Yes

120

DPS = \sqrt{\frac{(PSIWA)^2 \times RUCF(NNS) \times SH(NNS))^2 + 2 \times YW}{PSIWA}}

NPSI = NPSI + 1

NO

Set FIX(I), IDA(I), TRLD(I), RHLI(I) equal to wall values.

Is finite rate chemistry being used?

NO

YES

150

PSIWA = PSIWA + RUC x DX

Is wall slot cooling being used?

NO

YES

PSIWA = \sqrt{PSIWA^2 + 2 \times RUC x YW x DX}

Is NTYPE = 0?

NO

YES

Call "HOOT" to obtain coolant enthalphy for the NNSth slot

Is X < XS (NNS)?

YES

200

NO

200

DPS = RUCF (NNS) x SH(NNS)

Is NTYPE = 0?

NO

Yes

120

DPS = \sqrt{\frac{(PSIWA)^2 \times RUCF(NNS) \times SH(NNS))^2 + 2 \times YW}{PSIWA}}
SUBROUTINE FLUX (Cont.)

120  PSIWA = PSIWA + DPS

150

200

150  PSIWA = PSIWA + DELPS

DO Loop
I = MPSI to NAMP
Set TELAP(I), U(I), H(I), T(I)
of the new grid point(s) equal
to the slot flow conditions

DO Loop
Set the element mass
fractions, ALPHA(J,I)
to the slot flow composition.

DO Loop
Set the specie mass
fractions, YSPEC(J,1)
equal to the slot flow
composition.

180  NPSI = NPSI + NADD

180  LR = NPSI
MP = MPSI + 1
MQ = MP + 1
MR = 2 * MPSI

WRITE
IOUT, NADD, NAD, ADD, HAD,
PSIWA, PSIW, DEER, DPS
SUBROUTINE GRAD, NOZ

From PERF

For I = 1, 22
A(I) = A Log (AR(I))

RETURN

For J = 1, 22
I = 1, 6 (corresponding to \( \phi_F = 4, 6, 8 \) 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) \( \frac{\Delta DAX(J)}{RT} \)
which is \( \frac{\Delta H}{\Delta Q} \) from NASA CR 72601

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

RETURN

I = 1, 22
Calculate DADXI(T) = f (AR(I))
For $\psi > \Delta \psi$, 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:

(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 $\psi_1 = \psi - \Delta \psi$

Is the wall still beyond the computation grid?

Yes

999

No

2000
SUBROUTINE GRID (Cont.)

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:

\[ \frac{F_{\text{NPSI}} - F_{\text{MPSI}}}{F_{\text{MPSI}}} \leq 0.01 \]

NO

\[ \begin{align*}
\text{MPSI} &= \text{MPSI} + 1 \\
\text{NPSI} &= \text{MPSI} + 1 \\
\text{LR} &= \text{LR} + 1 \\
\text{MP} &= \text{MPSI} + 1 \\
\text{Duplicate information stored in previous top grid point in new top grid point. This includes values of TELAP, U, T, and ALPHA, or YSPEC, HHLD, IDA, THLD and FIX.}
\end{align*} \]

YES

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, T, ALPHA, or YSPEC, HHLD, IDA, THLD, and FIX are preserved. \( \phi_i \) is recomputed.

DO Loop

Fill up abandoned grid points (to MR) with values from new top point (MPSI) of grid for TELAP, U, T, ALPHA, or YSPEC, HHLD, IDA, THLD, and FIX. Recompute \( \phi_i \).

Compute MP, MQ and MR from MPSI and MF.

RETURN
SUBROUTINE HEAT (LLA, LLB)

DO Loop I = LA to LB

Compute GA, GB, GC - the coefficients of the temperature enthalpy polynomials in the temperature range of TT

DO Loop

Compute the mixture Cp (CPGAS) and, if needed (KDPT = 1), h (HE), and the enthalpy of each specie, h_i (W).

TM = T(i)
TS = TM x TM
TW = 2 x TM

HE(i) = 0

KOPT > 1

[T(i) - TT] > 100

KNT > 2

Does KOPT = 1?

H(i) = HE(i) + TAUT(i)

Is the Loop Complete?

RETURN
SUBROUTINE HEATT

From MAIN:
  - Initialize some parameters, mostly for output purposes.
  - Set up x, y coordinates for wall.
  - Read number of nodes, flags for type of problem.
  - Read data common to all types of problems.

Combustor geometric calculations:
  - 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.

  - Write the input.
  - Read injector input.
  - Do all regen stations match nodes?
    - Yes: Write error message. Call exit.
    - No: Injector.

  - Do all liner stations match nodes?
    - Yes: Write error message. Call exit.
    - No: Match film cooling stations with nodes for output purposes.

  - Make sure liner flow is injected as film cooling flow whether input as such or not.

  - Match x, y coordinates for liner. Treat them same as wall coordinates.

  - Calculate for each wall element, inner surface areas, outer surface areas, volumes, average cross-section area, accumulated inner surface areas.

  - Set up x, y coordinates for liner.

  - Calculate for each liner element, inner surface area, outer surface area, accumulated inner surface area.

  - Find the throat. Calculate A*.

  - Combustor heat transfer characteristics.

  - Conduction admittance.

  - Find the end of the liner.
SUBROUTINE HEATT (Cont.)
SUBROUTINE HEATT (Cont.)

200

Steady State

YES
Calculate wall temperatures
RETURN

NO

Find maximum delta time for stability

YES

Injector?

Find injector temperature at this time step

NO

Find wall temperatures at this time step
SUBROUTINE HONC

**KTEST = 1**

**IDA(KPSI) = IDA(KPSI) + 1**

**NO**

**Initialize Chemistry Computation parameters:** N, NDUM, NP, NALL, PHM, PLT, PUH, PLT, TMAX, FUDGE, TESTY

**YES**

Test to see if the initial Chemistry step size, DT, Should be set to the suggested step size, FIX(KPSI), or reduced

**YES**

**Initialize Physical Properties for Chemistry Computation = T, PRESS, ALPHA(J), AW(J), RHG, RW, TW**

**222**

**NUT = 0**

**NO**

**WRITE**

**IOUT, KPSI, RHG, T, H, FIX(KPSI), DT, ANN, TIMEF, DUM, TMAX, ALPHA, AW**

**4**

**YES**

Is DT > the Remainder of the Chemistry Time Step?

**NO**

Set DT = the Remainder of the Diffusion Time Step

Write:

<table>
<thead>
<tr>
<th>MY = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITE A(l, J) and C(J), the matrix and forcing sector Coefficients</td>
</tr>
<tr>
<td>Call &quot;PADE&quot; to solve the matrix</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MY &gt; 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set NUT = NUT + 2</td>
</tr>
</tbody>
</table>

Chemistry step size will not be increased for rest of Diffusion Step

<table>
<thead>
<tr>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call &quot;FG&quot; to Evaluate the Coefficient for the Matrix Solution of the linearized Elemental Chemical Reactions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>(DO Loop) Store the Existing ( \left[ \frac{\partial}{\partial t} \right] ) in KXY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MY &gt; 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITE C(J) = ( \Delta \left[ \frac{\partial}{\partial t} \right] )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta \left[ \frac{\partial}{\partial t} \right] ) &lt; .0 (but less than ( \epsilon ))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta \left[ \frac{\partial}{\partial t} \right] = -\left[ \frac{\partial}{\partial t} \right] )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta \left[ \frac{\partial}{\partial t} \right] ) &lt; .0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increment J</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta \left[ \frac{\partial}{\partial t} \right] ) Complete?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
</tr>
</tbody>
</table>

167
SUBROUTINE HONC (Cont.)

(DO Loop)
Set \( \frac{\alpha_j}{W_j} \) (active specie)
Compute \( DUM = \sum_{j=1}^{n} \frac{\alpha_j}{W_j} \)
all specie
\( GUM = \sum_{j=1}^{n} \left( \frac{\alpha_j}{W_j} \right) \)

For the revised Chemical Composition, the enthalpy, \( H \), the quadratic enthalpy Temperature curve fits, compute the New Temperature, \( T \), by inversion,

\[
RHO = \frac{P}{RT} \sum_{j=1}^{n} \frac{\alpha_j}{W_j}
\]

Is \( MT > 0 \) ?

WRITE
\( AWW, RHO, T, TT, TMEF, FIX(KPSI), DT, TIME, TMEF \)

DO Loop
\( j = 1 \) to \( N \)

Is \( HXY(j) \leq \text{TESTY} \) or \( AW(j) \leq \text{TESTY} \) ?

TESTW = \( 1 + 0.00464 \left[ HXY(j) \right]^{1/3} \)

TESTX = \( 1/\text{TESTW} \)

TEST = \( AW(j)/HXY(j) \)

Is \( \text{TEST} \leq \text{TESTW} \) or \( \text{TEST} > \text{TESTX} \) ?

INCREMENT J

Has the time step directly been increased?

Is the DO Loop complete?

Is the temperature changing too rapidly?

Is the density changing too rapidly?

Is \( \text{FIX(KPSI)} \leq \text{TMAX} \)?

Double Chemistry step size
\( DT = 2 \times DT \)
\( \text{FIX(KPSI)} = 2 \times \text{FIX(KPSI)} \)
\( \text{NUT} = \text{NUT} + 1 \)

Is \( \text{FIX(KPSI)} = \text{TMAX} \) ?

Is \( DT \leq \text{TMAX} \)?

\( DT = \text{TMAX} \)

YES

YES

NO

YES

NO

81
SUBROUTINE HONC (Cont.)

Is Density Changing too Rapidly?

Is Temperature Changing too Rapidly?

NO

NO

DT = DT/2

FIX(KPSI) = DT

Write Failure Message, IOUT, KPSI

Write ALPH, DT, T, X, ALPH, DT, T, X

Success: Set ALPHA(J) to composition before failure, and ALP to composition after obtaining negative species

Failure: Set TEMP = 5000 as a failure flag. Computation will be terminated

Repeat computation with printout dumps

Return
SUBROUTINE INVERT (IT, H, AL, CPP)

ENTRY HOOT

Set T = TT and AP array = AL array

CP = .0
HE = .0
TS = T x T
TWT = 2. x T

(Do Loop)
Compute Cp (CP) and h (HE)
for the given temperature (T)
and specie mass fractions (AP)

RETURN

TM = T
SA = .0
SB = .0
SC = -HE

(DO Loop)
Compute SA, SB, SC - the coefficients
of the temperature - enthalpy polynomials
in the temperature range of TM

WRITE
TM, T, HE, SA
SB, SC, SD, SE

RETURN

TT = T

Does LZ = 777?

NO

WRITE
X, DX, P, DPDX

Is SA = .0?

YES

NO

SD = -SB/SA
SE = SD x SD - SC/SA

T = SD / SE

Is SE < .0?

YES

NO

T = -SC/SB

RETURN

NUT = NUT + 1

SET T = TT, HE = H
and AP array = AL array

NUT = 0

LZ = 777

NO

WRITE
X, DX, P, DPDX

RETURN

TT = T
H = HE
CPP = CP
SUBROUTINE MARCH

From MAIN

Is Equilibrium Chemistry Being Used?

YES

CALL "HEAT" to compute Temperature, $C_p$, and specie enthalpies for all Grid Points

KOPT = 2 Specifies Enthalpy known and $T$ to be found

CALL "EQUILC" to compute the specie Chemical Composition for all Grid Points

Is Equilibrium Chemistry Being Used?

YES

CALL "HEAT" to compute Temperature, $C_p$, and specie enthalpies for all Grid Points

KOPT = 2 Specifies Enthalpy known and $T$ to be found

CALL "EQUILC" to compute the specie Chemical Composition for all Grid Points

NO

CALL "DENSE" to Compute Density, Molecular Weight and Mass Flux for all Grid Points

Is there more than One Grid Point?

YES

CALL "BULK" to Compute the bulk Properties across the Duct

NO

Is the 1-D Flow Area Prescribed?

YES

$y_w$ is an input

NO

$y_w = y_w (\omega, \rho u)$

DO Loop

Compute the mixing parameter, $A(I)$, for all Grid Points, for either Axisymmetric or 2-D flow

CALL "STEP..." to compute the Field Axial Step Size for the next Diffusion Step

CALL "WALL" to determine the Wall Boundary conditions as a function of the Cooling Model Used

Is there more than One Grid Point?

YES

CALL "VISC" to compute the Value of the Turbulent Viscosity Coefficient

DO Loop

Compute the Radial Location of each streamline, $Y(I)$

DO Loop

Compute the Vertical Location of each streamline, $Y(I)$

Is Flow Field Axisymmetric?

YES

NO

DO Loop

Compute the O/F = Ratio and the Stoichiometric Ratio from the ALPHA $(K,I)$

DO Loop

Compute the Element Mass Fractions, ALPHA $(K,I)$, from Specie Mass Fractions YSPEC $(J,I)$

DO Loop

Compute the Radial Location of each streamline, $Y(I)$

DO Loop

Compute the Vertical Location of each streamline, $Y(I)$

Is there more than One Grid Point?

YES

CALL "BULK" to Compute the bulk Properties across the Duct

NO

$y_w = y_w (\omega, \rho u)$

DONE

172
SUBROUTINE MARCH (Cont.)

Is the Computation Beginning or Ending?

NO: KKK = KKK + 1
    PCNT = PCNT + DX

YES: Has the Maximum Axial Distance of the Flow Field been reached?

NO: Continue

YES: IFINIS = 2 to terminate Calculation

The KKK > MC?

NO: Is PRNT <= PCNT?

YES: Call "PRINT" to write Flow Field Radial Profiles, Bulk Quantities, and General Information at the Present Axial Station

NO: KKK = 0
    PCNT = 0

YES: Does IFINIS = 2?

YES: RETURN

NO: Is Equilibrium Chemistry Being Used?

YES: RX = X + DX

NO: Call "PREPAR" to begin the finite rate chemistry computation for all Grid Points

YES: Is Finite Rate Chemistry Being Used?

NO: (GO Loop)
    Set all WDT = .6

YES: Call "PRESS" to calculate P, and PH+1 over the diffusion step from the inputted P(x) Polynomials

NO: Call "AREA" to calculate dP/dx over the diffusion step from the inputted f_m(x) Polynomials and the flow field conditions

a
SUBROUTINE MARCH (Cont.)

Are any of the Velocities or Mass Fractions, across the Grid Negative (DO Loop)

WRITE Failure Message, RX, DX, IOUT, YSPEC, RU, RT, RALPHA

IF INIS = 2

(IFINIS = 0 ?)

IFINIS = 1
MINIT = IPRESS
MHALF = 2*MINT-1

Are the transpiration or slot wall cooling models in use ?

Are there only one grid point ?

IFINIS = 0
IFINIS = 1
MINIT = IPRESS
MHALF = 2*MINT-1

Call "FLUX" To record the amount of mass added to the flow field through the wall over the last diffusion step

Call "GRID" to see if additional grid points should be added to the computation, or \( \Delta \) \$ doubled
SUBROUTINE PADE

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

\[
y(t_0 + \Delta t) - y(t_0) = \left[I - \bar{A} \left(\frac{\Delta t}{2}\right) + \bar{A}^2 \left(\frac{\Delta t}{12}\right)^2\right]^{-1} \Delta t \bar{B}
\]

\[
\begin{align*}
C_1 &= \Delta t^2/12 \\
C_2 &= \Delta t/2 \\
(\text{DO Loop}) &\quad AS = \bar{A} \times \bar{A} \\
(\text{DO Loop}) &\quad B = \bar{B} \Delta t \\
&\quad \bar{C} = I - \bar{A} \left(\frac{\Delta t}{2}\right) + \bar{A}^2 \left(\frac{\Delta t}{12}\right)^2 \\
\end{align*}
\]

CALL PICRT to evaluate

\[
y(t_0 + \Delta t) - y(t_0) = \bar{C}^{-1}\bar{B}
\]

using the Crout Matrix reduction technique.
SUBROUTINE PERF

From MAIN

Convert EY, R, W, PT, CStar, and T obtained 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 Grad Subroutine To Calculate
\[ \frac{\partial H}{\partial Q} \] 
\[ \frac{\partial (A/A_{\text{Min}})}{\partial X} = 1 \]
\[ \frac{\text{AREA RATIO GRADIENT}}{\text{RADIUS THROAT}} \]

Call SI Subroutine to calculate Frozen & Shifting Isp for each streamtube

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 SIDEI To Calculate Isp Kinetic for each Streamtube

Call Subroutine DIV To Find % Loss in Isp Due to Nozzle Configuration

Call Subroutine BL To Calculate Boundary Layer Loss in Nozzle

PERF Cont. I
SUBROUTINE PERF (Cont.)

1. Calculate Streamtube Areas A(I) and find Mass Averaged Kinetic Isp (SDK)
2. Specific Impulse Loss Due to Boundary Layer (SDD)
3. Average Mixture Ratio (OF) and Call Subroutine SI To Find Overall Shifting Isp (Theor) 100% \( \eta_c \) (ST)
4. Theoretical Mass Averaged Isp (SSCOMB) Based on Combustion Efficiency
5. Determine Delivered Isp and Thrust SDEL AND FDEL
6. PRINT RESULTS
7. IS FKM Dumped in Supersonic Region
   a. NO
      i. Calculate \( C_F \) Based on Dynamic Pressure RETURN
   b. YES
      i. Determine Hydrogen Isp = f(expansion (Area Ratio) and correct for Injection Temp.
      ii. Multiply by Mass Injected to Find Additional Thrust
      iii. Add to Previous Thrust (FDEL) and Divide by Total Flow to Find Isp (DS)
8. PRINT RESULTS
SUBROUTINE PERSHF

From PERF

Calculate Percent Recombination (ISPK(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

SET ISPK(I) = 1.

YES

ISPK(I) > 1.0

NO

RETURN

NO

ISPK(I) < 0

YES

Set ISPK(I) = 0
SUBROUTINE PREPAR

1. **Initialization**
   - **CMT = 0**
   - **KTM = 0**
   - Create array HDX
   - Create array XMPS
   - **IOUT = 1**

2. **Loop**
   - **DO Loop**
     - Create array IMM, I = 1 to MPSI
     - **HC = SMALLH(I)**
     - **Set AL(I) array = YSPEC (J, I) Array**
     - **DT = DXU(I)**
     - **TMP = T(I)**

3. **Conditions**
   - **Are the Temperature, T(I), and the specie, YSPEC (J, I), the same as for the previous grid point?**
     - **YES**
       - **Set ΔTChem, ΔCChem, WDT(I, B), equal to the values for the IMth grid point**
     - **NO**

4. **Conditions**
   - **Are there other species besides the diluent present?**
     - **NO**
       - **Set ΔTChem, ΔCChem, WDT(I, B), equal to the values for the IMth grid point**
     - **YES**
       - **Is TMP < 500?**
         - **YES**
           - **Call "HONC" to perform the finite rate chemistry computation for the Ith grid point**
         - **NO**

5. **Conditions**
   - **Is the Loop on I complete?**
     - **NO**
       - **Set TCHEM equal to the Change in Temperature due to Chemistry and WIT(I, I) to the change in species**
     - **YES**

6. **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**
   - **RETURN**

7. **Set IFINIS = 2 to terminate computation**
   - **WRITE Failure Message**
SUBROUTINE PRESS (PARR, DPARR, X, CGPR, XP, LYY)

LY = LYY

(DO Loop)
Set CGP Array = CGPR Array

Is X \leq XP(LY) ?

YES

LY = LYY

NO

LY = LY + 1

(DO Loop)
Compute PAR and DPAR from the LY\text{th} CGP polynomial

PARR = PAR
DPARR = DPAR

RETURN

Initialize DPAR, XT, PAR
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 \( A_{ij}^{-1} \) less than \( \epsilon \)?
  - Yes: WRITE Failure message -have singular matrix
  - No: Solve \( A_{ij} B_i \) for \( y_i \) using Crout's method

RETURN
SUBROUTINE PRINT

1. Is this the first axial step?
   - Yes: Compute sensible total enthalpy $H_i = \sum_{j=1}^{n} \omega_j$
   - No: Increment $I$

2. (DO Loop) Compute mole fractions, $X_i = \frac{Z_i}{\sum_{j=1}^{n} Z_j}$ and $SUM(i) = \sum_{j=1}^{n} \omega_j$

3. Is there more than one grid point?
   - Yes: Continue
   - No: Compute the bulk quantities

4. Is $P(X)$ inputted for the duct?
   - Yes: Compute the wall location $r_w$
   - No: Compute bulk properties Radially across the chamber: $\bar{H}_r, \bar{Y}, \bar{C}, \bar{M}, \bar{T}, \bar{\tau}

5. Is the flow field axisymmetric?
   - Yes: Compute $\bar{n} = n_{w} \delta
   - No: $I = PA + \bar{n} \bar{U}$

6. Is this the first axial step?
   - Yes: $F_{i} = F_{i-1} - F_{0}$
   - No: $F = F + F_{0}$

WRITE
Title information, $X, IPAGE, P, Lx, Pr, \mu, \delta x$, OUT

IPAGE = IPAGE + 1

$P$ (psia) = $P(N/M^2)$

101325.

$XD = X/XK$
SUBROUTINE SI

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 \( \bar{OF}(I) \) and Set \( II = 1 \)

Set \( \bar{OF}(I) = 9 \)

\( \bar{OF}(I) > 9 \) ?

\( \bar{OF}(I) \) Integer from 0 to 9

Go to Equation for SISD(II), SIFD(II) for \( \bar{OF}(I) \) Integer from 0 to 9

Interpolate for \( II = 1, 2, SISD(II) \) SIFD(II) to find SIS(I) and SIF(I) at specified Input \( \bar{OF}(I) \)

Add 1. to \( \bar{OF}(I) \) and \( II = 2 \)

\( II = 2 \) ?

\( \bar{OF}(I) \) ?

RETURN

I = NS ?

YES

NO

YES

NO
SUBROUTINE SLOT

Initialize Parameters:
NSLCR = 1
HTCG = A* 0.9 (0.025 Ds/Dc)^0.5 Pr^-0.6 (Ds/Rc)^0.1
FPZ = .001
WTLM = 10
TWS(K) = T(LR); K = 1 to NSLOT

Call "INVERT" to Obtain Adiabatic Wall Temperature, TAW, from HAW and ALAW

WW is Wall Molecular Weight

Compute Adiabatic Wall Enthalpy, HAW from Bulk Conditions

(DO Loop)
Set ALS(I) Equal to Chemical Specie at the Wall, and ALAW(J) to Bulk Values of Species Across the Chamber

DO Loop
K = NSLCR to NSLOT

Compute UB(K), and SF(K)

WRITE IOUT, NUSE, NSLCR, HTCG, HAW, TAW, WW, RUE, RRUT, VS

WRITE IOUT, NSLOT, K, UC(K), UB(K), RUCF(K), FFF(K), UGA, UGC, SF(K)

Is MY > 0 ?

WRITE IOUT, NUSE, NSLCR
HTCG, HAW, TAW, WW, RUE, RRUT, VS

WRITE IOUT, NUSE, NSLCR

Increment K

Increment K

Increment K
SUBROUTINE SLOT (Cont.)

1. Compute ANW, EMW and CS (A*, B*, and C*)

2. CALL "VASC" to compute coolant laminar viscosity, VC.

3. WRITE IOUT, NP, IWTLM, ANW, EMW, CS, RRHO, RCP, RVS, TWS (1)

4. CALL "ROOT" to compute coolant enthalpy, HCS(K), from coolant temperature and specie.

5. DO Loop K = 1 to NSLCR

6. CALL "VASC" to compute laminar viscosity, VWS(K), at wall from TWS(K).

7. Compute HTC(K), ST(K), GGG(K), RYNC(K), and ETA(K)

8. WRITE K, IW, KM, HCS(K), VC, VWS(K), TCS(K), TWS(K), RWS(K), SG(K)

9. Is MY > 0?

10. Write K, IW, KM, HTC(K), ST(K), GGG(K), RHC, RYNC(K), ETA, ETA(K)

11. Is ETA(K) > 1?

12. YES ETA(K) = 1.

13. T_w = T_w - \eta^k (T_c - T_w)

14. YES K = 1

15. T_w = T_a + \eta (T_c - T_a)

16. Compute RWS(K) and SG(K) \((\rho_w^k, \sigma^k)\)
SUBROUTINE SLOT (Cont.)

$\text{EPW} = \frac{TWS(K)}{TWB} - 1$

WRITE IOUT, K, IW, EPW, TWB, TWS(K), GRPZ

Is IW > IWTLM?

YES

Is EPW ≤ EPZ?

YES

TWB = TWS(K)

NO

Increment K

NO

TW = TWS(NSLCR)

store information for next step:
RRUT = RUTBL
RUE = UBL
RRHO = RHOB
RCP = CPBL
RVS = VS
RALAW(J) = ALAW(J)

WRITE PROBLEM MESSAGE
IOUT, ISOBAT, K, TWS(K), TWB, ETA(K), RYNC(K), RHC, HTC(K), SG(K)

Is the loop on K complete?

YES

Is NUSE = 0?

NO

RETURN

YES

RETURN

132 140 187
SUBROUTINE STEP

Is there more than one grid point? NO YES

Is Finite Rate Chemistry being used? NO YES

Is this the first flow field step? NO YES

Evaluate DX as the smaller of either the streamtube radius, or a function of the total flowfield length

Evaluate DX as a function of the changes in velocity and mass flux

Evaluate XD as the minimum of all grid point values for XD as a function of Sc, ∆ψ, µ

Evaluate XD as a function of centerline values for Sc, ∆ψ, µ

Evaluate DX as the minimum of all grid point values for XD and the radial distance between adjacent grid points

DO Loop

DO Loop

WRITE

RETURN

RETURN

188
SUBROUTINE TRANSP

Is this the first axial step using this cooling model?

NO

Initialize parameters:
EPT = .01
PRW = 0.02 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 computed below.

Call "PRESS" to evaluate the local coolant mass flux, RUC, from the inputted polynomials. The wall temperature, TW, will be computed below.

Is ISBATY > 2 ?

NO

Call "ROOT" to evaluate the coolant enthalpy, HC, from TC.

Is LZ > 0 ?

NO

WRITE IOUT, NUSE, ISBATY
TC, CPC, HC, TW, RUC

Is ISBATY = 1 ?

NO

Is ISBATY = 2 ?

NO

Is ISBATY = 3 ?

NO

NO

YES

YES

YES

YES

NO

50

50

50

YES

YES

NO

150

22

Is YSPEC(NNT, K) < (YSPEC(NNT, LR) x EPT) ?

K = LR - J + 1

DO Loop J = 2 to LR

YES

NO

Is the loop on J complete?

YES

NO

Set the "edge" conditions for H, a', µ, ρ, C', h, and T, equal to the properties of the NEDth grid point.

NED = LR - 1

NED = K

INCREMENT J

NO

22

22
SUBROUTINE TRANSP (Cont.)

Set the "edge" conditions for \( H_{aw} \), \( \alpha_{aw}, \rho, C_p, h \), and \( T \), equal to the chamber bulk properties.

**WRITE**

NED, ITWL, NCL, RRRT, RUE, RRHO, RCP, RHEE, RHAW, RAWNCL.

**YES**

**WRITE**

IOUT, NED, LR, FF, WW, WEB, RCW.

**NO**

Is \( LZ > 0 \)?

**WRITE**

IOUT, NED, LR, FF, WW, WEB, RCW.

**YES**

Compute \( FF, WW, WE, WEB \) (\( F, W, W^*, W_e \)).

\[ WM = \frac{WE}{WW}, RCW = 1 - \alpha_{aw} \]

**NO**

Compute \( ALSB, \) (\( ALSB(J), J \neq NCL \)), and WS (\( \alpha^*, \alpha^*_j, \) and \( W^* \)).

**WRITE**

IOUT, NED, LR, FF, WW, WEB, ALSB, ALSD, WSPEC(NCL, LR).

**IS**

\( 1 \geq ALSB \geq 0 \)

**ALSB = 0**

**NO**

Compute \( ALSB (G^*) \).

Is \( LZ < 1 \)?

**Yes**

**LZ = 1**

Compute ALS(NCL), (ALS(J), \( J \neq NCL \)), and WS (\( \alpha^*_e, \alpha^*_j, \) and \( W^* \)).

**IS**

\( ISBATT > 2 \)

**T(LR) = TW**

**ITW = ITW + 1**

**KOPT = 1**

**WRITE**

FF, WW, WEB, ALSB, ALSD, WS, TWB, TSTAR.

**IS**

\( LZ > 0 \)

**ITW = 0**

**TWB = T(LR)**

**TSTAR = TAW**

**ITW**

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 \( h^* \).
SUBROUTINE TRANSP (Cont.)

Call "VASC" to Compute VSTAR (\( \mu^* \)) from \( T^* \) → Compute RHST and RYNS (\( \phi^* \) and \( \phi_{ox}^* \)) → Is \( LZ < 1 \) and \( \phi_{ox}^* < 0 \)? → Yes → \( LZ = 1 \)

Write Error Message

Is \( STAN = 1 \)? → Yes → Compute STAN (\( S_{st} \))

Is \( LZ > 0 \)? → No → Write ITW, LR, MPSI, HSTAR, CPSTAR, VSTAR, RHST, RYNS, STAN, TWB

Is \( ISBATY > 2 \)? → No → Compute GG and TWB (G and \( T_w \))

Is \( ITW < 1 \)? → Yes → \( T(LR) = TWB \)

Write Error Message, IOUT, ISOBAT, ISBATY, TWB, TWA, GG, STAN, RYNS, RHST, TSTAR

Compute GG, FF, RUC (G, F, \( \dot{m}_c/A \)) → NUSE = NUSE + 1

SCW = \( \frac{\dot{m}_c}{\rho_b} \left( \frac{\dot{n}_m}{\rho_c} \right) \) (2 \( \Delta \phi \))

Is Equilibrium Chemistry Being Used? → Yes → Write FF, GG, RUC TWA, TWB, TW, SCN

Compute Wall B.C. for Diffusion Conservation eq. in Terms of Species:

\[ \left( \frac{\partial \phi}{\partial \xi} \right) \left( \frac{\partial T}{\partial \xi} \right) \left( \frac{\partial \phi_1}{\partial \xi} \right) \]

Is \( LZ > 0 \)? → Yes → 250

Repeat Computation With Printout Dump
SUBROUTINE TRANSP (Cont.)

Compute wall B.C. for Diffusion Conservation equation in terms of chemical elements:
\[
\left( \frac{\partial \bar{T}}{\partial \varphi} \right)_w, \left( \frac{\partial \bar{v}}{\partial \varphi} \right)_w, \left( \frac{\partial \bar{\rho}}{\partial \varphi} \right)_w
\]

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)

252

Is this the first axial step using this cooling model?

RETURN
SUBROUTINE VASC (XMU, TT, ALP)

T = TT  

(Do Loop)  
Set AL Array = ALP Array  

CNV = 47.87  

Is LW = 1?  

Compute Local Laminar Viscosity, XMUT, from Sutherland air viscosity model  

XMU = XMUT  

RETURN
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

WRITE PSIW, T(NPSI)

NPSI = MPSI - 1

PCNT = PRNT

PSIW = PSI(NPSI)

PSIW = PSI(NPSI)

NO

NO

NEW = 1

YES

Compute adiabatic wall total enthalpy, HAW, from chamber bulk conditions

Call "INVERT" to obtain adiabatic wall temperature - TAW

Compute adiabatic wall total enthalpy, HAW, from chamber bulk conditions

Call "INVERT" to obtain adiabatic wall temperature - TAW

Impose Wall B.C. on momentum equation:

\[
\frac{\partial \mu}{\partial x} = \frac{C}{2} \times RUSE \times \text{GAMW} \\
\mu_{\text{MPSI}} = \mu_{\text{JR}} - 2 \Delta \phi \frac{2 \phi}{2 \phi_{\text{w}}} \\
\]

Is \( \mu_{\text{MPSI}} < 0 \)?

NO

YES

Is the Transpiration Wall Cooling Model Being Used?

YES

200

NO

TRANSP_500

(DO Loop)

Set element and species mass fraction at MPSI to values at JR

26

27

28

29

195
SUBROUTINE WALL (Cont.)

27

Issenergetic, Impermeable Wall
H (MPSI) = H(J)
R(J) = H(J)
KOPT = 2

Call "HEAT" to obtain
T(MPSI) from H(MPSI)

Is

T(MPSI) ≤ 0 ?

YES

 FAILURE

NUSE = NUSE + 1

NO

Is ISBATY > 1 ?

YES

Call "PRESS" to
Evaluate Wall Temperature Polynomials

28

Is NUSE = 0 ?

YES

RPRB = PRB

NO

NUSE = NUSE + 1

Is ISBATY > 1 ?

YES

Call "PRESS" to
Evaluate Wall Temperature Polynomials

25

Compute wall
Heat transfer Rate, QL

Call "HEAT" to
Evaluate Enthalpy at the Wall, and at
B.C. Grid Point, from $T_W$

T(NPSI) = $T_W$
T(MPSI) = $T_W$
KOPT = 1

Is

TW ≤ 0 ?

YES

Failure

NO

281

Evaluate $H_{MPSI}$ from QL

Call "HEAT" to
Compute $T_{MPSI}$ from $H_{MPSI}$

Is

T(MPSI) ≤ 0 ?

YES

Failure

NO

29

TW = T (NPSI)

200

Call "PRESS" to
Evaluate Local
Coolant Temperature, $T_C$, for the transpired Wall, from the Inputted Polynomials

Is

TC > T(NPSI) ?

NO

Is ISBATY < 3 ?

NO

Call "PRESS" to
Evaluate Wall Temperature from
the Inputted Polynomials

YES

(NO cooling required)

YES

26

T(LR) = TW
T(MPSI) = T(LR)
KOPT = 1

Call "TRANSV" to
Evaluate conditions at the transpiration cooled wall

Is

TW ≥ T(LR) ?

YES

(NO cooling required)

YES

26

(NO cooling required)
SUBROUTINE WALL (Cont.)

- Call "ELJULC" 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
- Call "SLOT" to compute the Conditions at the Slot Cooled Wall
  - If FINIS = 2 to terminate Computation
  - WRITE Failure Message and RETURN
  - Store Information for Next Stop: HRUSE = RUSE, RPRB = PRB, RVB = VB and RETURN
- If LR = MPSI (Wall is still above computation grid) RETURN
- If LR < MPSI (Wall is still above computation grid) RETURN
- If T(LR) < 0 THEN FAILURE
- If T(MPSI) < 0 THEN Call "HEAT" to evaluate the Temperature from the Enthalpy at the B.C. Grid Point
- If the First Slot been Reached yet THEN Call "SLOT" to compute the Conditions at the Slot Cooled Wall
  - H(MPSI) = H(JR)
  - RH(MPSI) = H(JR)
  - KOPT = ?
- If Equilibrium Chemistry in Use THEN Call "EQUILC" to compute the Chemical Species from the Modified Chemical Element Composition at the Wall and B.C. Grid Points
- If NO THEN RETURN
C0750 272  GC TO 716
C0751 273  705 TASTAR=894.757
C0752 274  706 CC 707 J=1.8
C0755 275  XP(1)=X(1),.0254
C0756 276  707 CONTINUE:
C0756 277  C
C0760 278  C READ (5,1)(0)9001
C0760 279  C FOUR CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
C0760 279  C
C0766 280  WRITE (6,12)
C0770 281  152 FORMAT (4,31=WALL RADIUS(INCHES) POLYNOMIALS )
C0770 282  C POLYNOMIALS FOR PRESSURE OR WALL RADIUS
C0771 283  WRITE (6,11)
C0777 284  C
C0780 285  IF (HS=1)708,709,711
C0802 286  708 CC 709 J=1.4
C0802 287  GCP(7,J)=GCP(7,J)+.0254
C0806 288  710 CC 711 J=1.6
C0806 289  GCP(1,J)=GCP(1,J)+.0254/.0254*(1-I))
C0813 290  709 CONTINUE
C0815 291  GC TO 713
C0816 292  711 CC 712 J=1.4
C0821 293  GCP(7,J)=GCP(7,J)+.0254
C0822 294  713 GCP(1,J)=GCP(1,J)+.0254/.0254*(1-I))
C0827 295  712 CONTINUE
C0829 296  714 CONTINUE
C0831 297  C
C0831 298  C READ IN WALL BOUNDARY CONDITION INPUTS
C0832 299  IF (P=0.23,1360 TO 50
C0834 300  GC TO(31.62)+33.54).ISOBAT
C0835 301  51 CONTINUE
C0839 302  C IDENERGETIC WALL - IMPERMEABLE WALL
C0839 303  GC TO 91
C0839 304  52 CONTINUE
C0839 305  C EXTERNAL REGENERATIVE COOLING - IMPERMEABLE WALL
C0840 306  IF (SBEGY.GT.1100 TO 521
C0842 307  WPAD(5,1)=W
C0842 308  C FOUR CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
C0850 309  153 FORMAT (4,44=REGENERATIVE COOLING - WALL TEMP(R) PROFILE )
C0852 310  WRITE (6,41)
C0853 311  WRITE (6,41)
C0856 312  CC 714 J=1.4
C0864 313  T=7,X(7,J)+.0254*TWX(7,J)
C0865 314  715 TWX(J)=TWX(J)+55556/.0254*(1-I))
C0870 315  716 CONTINUE
C0872 316  717 CONTINUE
C0874 317  GC TO 51
C0874 318  521 CONTINUE
C0876 319  WPAD(5,1)=W
C0876 320  C FOUR CARDS OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
C0884 321  WRITE (6,39)
C0886 322  154 FORMAT (4,3=REGENER. WALL COOLING HEAT TRANSFER RATE )
C0890 323  WRITE (6,41)
C0891 324  CC 716 J=1.6
C0906 325  QX(7,J)+.0254*OLX(7,J)
C0915 326  CC 717 J=1.6
C0919 327  718 QX(J)=OLX(J)+1.6353176+6/(.0254*(1-I))
C0924 328  719 CONTINUE
C0930 329  GC TO 91
C0930 330  716 CONTINUE
C0933 331  93 CONTINUE
C0933 331  93 CONTINUE
1132 332  WRITE(6,225)HLCX,TX
1134 334  WRITE(6,155)
1136  C  WRITE CARD OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
1150 355  155  FOR IAT(4),,45*COOLANT MASS FLUX AND TEMPERATURE POLYNOMIALS
1151 357  WRITE(6,155)
1153 355  WRITE(6,155)
1155 355  IC 721 J14,4
1170 355  HLCX(7,J)+.254*RUCX(7,J)
1171 355  TCX(7,J)+.025*TCX(7,J)
1172 355  IC 721 J14,4
1176 355  HLCX(1,J)+RUCX(1,J)+703.07/(/.0254**(1+1))
1177 355  TCX(1,J)+TCX(1,J)+.55556/(/.0254**(1+1))
1200 345  CONTINUE
1202 346  GO TO 51
1203 347  533  CONTINUE
1204 348  WRITE(6,155)
1205 349  WRITE CARD OF INPUT - EACH CARD IS A SEPARATE POLYNOMIAL
1219 355  156  FOR IAT(4),,42*WALL AND COOLANT TEMPERATURE, (R) POLYNOMIALS
1220 357  WRITE(6,155)
1222 355  WRITE(6,155)
1227 355  IC 721 J14,4
1229 355  TCX(7,J)+.0254*TCX(7,J)
1231 355  IC 721 J14,4
1233 355  TCX(7,J)+.0254*TCX(7,J)
1235 355  IC 721 J14,4
1238 355  TX(1,J)=TX(1,J)+.55556/(/.0254**(1+1))
1243 355  722  TX(1,J)+TCX(1,J)+.55556/(/.0254**(1+1))
1250 360  CONTINUE
721 360  CONTINUE
1252 361  4T TO 51
1253 362  CONTINUE
54  CONTINUE
1254 364  WRITE(6,155)
1255 365  IC 721 J14,4
1260 367  WRITE(6,155)
1261 368  157  FOR IAT(4),,49*SLOT X(N),X(N),U(N),S(N),RHO=U(LB)/1X2=S(I,T(R))
1262 369  IC 344 K1*NLNO
1263 370  WRITE(6,155)
1264 371  WRITE(6,155)
1265 372  WRITE(6,155)
1266 373  WRITE(6,155)
1267 374  WRITE(6,155)
1268 375  WRITE(6,155)
1269 376  WRITE(6,155)
1270 377  WRITE(6,155)
1271 378  WRITE(6,155)
1272 379  WRITE(6,155)
1273 380  WRITE(6,155)
1274 381  158  FOR IAT(4),,2*R(M),D(M),PR,AP(M2)
1275 382  WRITE(6,155)
1276 383  WRITE(6,155)
1277 384  WRITE(6,155)
1278 385  WRITE(6,155)
1279 386  WRITE(6,155)
1280 387  CONTINUE
1281 388  WRITE(6,155)
1282 389  WRITE(6,155)
1283 390  WRITE(6,155)
1284 391  CALL PRESS(P,CPDX,X,CGP,X,LY)
1285 392  UNIVITIComputing Company
C DRIVER CALLING SEQUENCE
C IF (NDY=4,ED.0) GO TO 680
C TIME=0.0
C C SET PRESSURE TO TEST FOR COMBUSTION PROGRAM CALL
C IF (NC/HR.EQ.0) GO TO 683
C PTCAL=PFCMC
C NTCT=KCOMCL
C CONTINUE
C CALL DYNA
C IF (NC/HR.EQ.0) GO TO 684
C IF (LKEG) TGT AND PRES(CVOL)+LT.PTCAL) GO TO 684
C IF (ICOCBICVOL).EQ.2) GO TO 685
C CALCULATE COLE FLOW PERFORMANCE
C IF(NPT=1,-FM)/FM
C CALL GCO (PC, WU/TC, CO, RTT, RX, PERBEL, WM, TG, TIME, OFNPT)
C IF (NTCT.EQ.1) GO TO 1685
C PTCAL=PFCAL+PDCMC
C GO TO 2684
C NTCT=NTCT+KCOMCL
C CONTINUE
C CONTINUE
C C CALCULATE HOT FLOW PERFORMANCE
C URETAIN:
C CONTINUE
C GO TO 938
C CONTINUE
C PPSAT=69947572
C CALL CC/5
C OF/P=1,-FM)/FM
C INPUT=1,E=NTHD)+((1,-FM)*E/NTHH)
C CALL P/E[RATTI, RX, PERBEL, PCLT, MP, OF, R, P, RUT, WM, TLB, MP8]
C ICN, TIME, PRES(CVOL) OFNPT.EINPUT)
C IF (NTCT.EQ.1) GO TO 1684
C PTCAL=PFCAL+PDCMC
C GO TO 2684
C NTCT=NTCT+KCOMCL
C CONTINUE
C CONTINUE
C CONTINUE
C IF (L=ER.KETER) GO TO 686
C C ARE VALVES SHUTTIC
C DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
C IF (TFST.T EQ.0,0.OR.TIME.LT.TFST) GO TO 687
C IF (PRES(CVOL)+LT.1.0) GO TO 686
C GO TO 618
C CONTINUE
C CONTINUE
C FAKE SHUTDOW/ TESTS FOR RISING COMBUSTOR PRESSURE
C IF (P/C=V,(PRES.(AND.PRESS(NFRETCH),SE.PBST) GO TO 600
C GO TO 619
C CONTINUE
C CONTINUE
C DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
C IF (TFST.T EQ.0,0) GO TO 686
C IF (TIPE.E=TFST) GO TO 609
C TIME=TFST/0.0005
C RETURN TO DYNAMICS PROGRAM
C GO TO 699
C CONTINUE
C DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL,
C IF (TIME.LE.VACSLT.OR.VACSLT.EQ.0.0) GO TO 602
C IF (NB/F.G.E.1) GO TO 688
C CALL BLOW (MBLOW)
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2246</td>
<td>569*</td>
<td>GO TO 576</td>
</tr>
<tr>
<td>2247</td>
<td>570*</td>
<td>680 CONTINUE</td>
</tr>
<tr>
<td>2250</td>
<td>571*</td>
<td>IF (NGCOMM.EQ.0) GO TO 686</td>
</tr>
<tr>
<td>2250</td>
<td>572*</td>
<td>C COMBINATION CALCULATIONS FOR CASE WITHOUT DYNAMICS</td>
</tr>
<tr>
<td>2252</td>
<td>573*</td>
<td>CALL CCM3</td>
</tr>
<tr>
<td>2253</td>
<td>574*</td>
<td>FPRFINPT/1.0FINPT</td>
</tr>
<tr>
<td>2254</td>
<td>575*</td>
<td>INPUT=(.F.*.FINPT)**(1.0-FINPT)</td>
</tr>
<tr>
<td>2255</td>
<td>576*</td>
<td>CALL PERFIN(ATTI,GEN,PXPELB,PTBL,H,CF,RA,Y,RUT,WMPL,TTBL,MPNL)</td>
</tr>
<tr>
<td>2255</td>
<td>577*</td>
<td>1000,TIME,PC,OFINPT,EINPUT)</td>
</tr>
<tr>
<td>2255</td>
<td>578*</td>
<td>686 CONTINUE</td>
</tr>
<tr>
<td>2257</td>
<td>579*</td>
<td>C DYNAMICS AND PERFORMANCE CALCULATIONS COMPLETE OR BYPASSED</td>
</tr>
<tr>
<td>2257</td>
<td>580*</td>
<td>IF (NHEAT.EQ.0) GO TO 691</td>
</tr>
<tr>
<td>2261</td>
<td>581*</td>
<td>CALL HEATT</td>
</tr>
<tr>
<td>2262</td>
<td>582*</td>
<td>691 CONTINUE</td>
</tr>
<tr>
<td>2263</td>
<td>583*</td>
<td>IF (ICAS.LT.NCAS) GO TO 751</td>
</tr>
<tr>
<td>2265</td>
<td>584*</td>
<td>444 CONTINUE</td>
</tr>
<tr>
<td>2265</td>
<td>585*</td>
<td>STOP</td>
</tr>
<tr>
<td>2267</td>
<td>586*</td>
<td>END</td>
</tr>
</tbody>
</table>

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

PHASE 1 TIME 00101.994 
PHASE 2 TIME 00100.998 
PHASE 3 TIME 00102.695 
PHASE 4 TIME 00100.390 
PHASE 5 TIME 00101.941 
PHASE 6 TIME 00101.957 

TOTAL COMPILATION TIME * 00107.148 

MAIN SYMBOLIC 

MAIN CODE RELOCATALE 

(FNS) 23 JUN 72 09:16:11.11 0 00335226 14 904 (DELETED) 

(FNS) 23 JUN 72 09:16:11.11 1 00413206 64 1 (DELETED) 

0 00413332 14 244
C0059    UFACEAB:7X*DPDX/RUT(1)
C0060    IF(UFACE.LT.(1))GO TO 53
C0061    PFAC,1=U(1)*RUT(1)/DX
C0062    IF(UFACE.DT.0.01PFAC=PFAC
C0063    OXI=PARAC
C0064    PULIT=PCC*DPDX
C0065    KAT=KAT+1
C0066    IF(KAT-10.52,5.54
C0067    54    WRITE(6,20)PNEW,RU(1),U(1),DX,DPDX,RUT(1),UFAC,PFAC
C0068    53    CONTINUE
C0069    KAT=KAT+1
C0070    IF(KAT.EQ.10)GO TO 23
C0071    GO TO 36
C0072    RUT(1)RUT(1)-DX*DPDX/RUT(1)
C0073    IF(RUT(1).LT.0.0)GO TO 50
C0074    SMALLH(1)=RUT(1)-5*RUT(1)+RUT(1)/AKA
C0075    GO TO 234,1,AKA
C0076    236    M(3)=MVEPC(JM)
C0077    CALL INVERT(RT(1),SMALLH(1),AP,CPX)
C0078    IF(RU(1).LT.0.0)GO TO 50
C0079    36    RHOZ(1)=PNEW/RUT(1)
C0080    OTHERWISE
C0081    IF(NRYPE.0.7)GO TO 12
C0082    YZ(1)=PSI(1)**(1)
C0083    GO TO 42
C0084    12    YZ(1)=PSI(1)/(RHOZ(1)*RUT(1))
C0085    42    GO TO 14
C0086    IF(LEDPE.VE.O1GO TO 28
C0087    28    YZ=YZ(1-1)+2*DELP1*PSI(1)/RHOZ(1)^RUT(1)=1-1/1/RHOZ(1)=1/RUT(1)
C0088    11-1))
C0089    IF(YZ.LE.0.0)GO TO 300
C0090    YZ(1)=SCHT(YZ)
C0091    GO TO 14
C0092    28    YZ=YZ(1-1)+DELP1/2*(1/RHOZ(1)/RUT(1)=1/1/RHCZ(1-1)/1/RUT(1))
C0093    14    CONTINUE
C0094    IF(LEL.VE.O1GO TO 13
C0095    WRITE(6,20)KOUNT,RHOUZ(KOUNT),YZ(KOUNT),RU(KOUNT)
C0096    OTHERWISE
C0097    IF(NRYP.GE.10)GO TO 4
C0098    WRITE(6,20)
C0099    30    IF(NRYP.EQ.9)GO TO 4
C0100    IF(NRYP.VE.O1GO TO 29
C0101    29    YP=SNAT(YZ(MPS))*2*(PSI**2-PSI(MPS)**2)/(RHCZ(MPS)**RUT(MPS))
C0102    12    GO TO 5
C0103    29    YP=SNAT(YZ(MPS)+PSI/PSI(MPS))RHOZ(MPS)/RUT(MPS)
C0104    5    ALK=SNAT(YP/YM)
C0105    IF(ALK.EQ.FALPA)GO TO 115
C0106    FALPA=FALPA
C0107    FALPA=FNK
C0108    FDN=FDNF
C0109    FYDPYWN
C0110    IF(ALP**LE.EPS1)GO TO 6
C0111    67    IF(KET.EQ.ALPA.LT..001)GO TO 26
C0112    DELTA=1*DELTA
C0113    GO TO 27
C0114    26    IF(CM.GO.1)GO TO 700
C0115    27    IF(KET.EQ.ALPA.LT..001)GO TO 26
C0116    9    IND=9
C0117    PNEW=PNEW
C0118    P Ya(1)=POLD/PN-1.0
C00119          pay(2)+yw/ywp-1.0
C00120          pay(3)+y/ywp-1.0
C00121          pay(4)+pay(1)+pay(2)/pay(3)
C00122          if(pay(4).gt.1.2)pay(4)=1.2
C00123          if(pay(4).lt.-8)pay(4)=8
C00124                          d pay(4)
C00125          if(pay(4).lt.pfrd)go to 30
C00126                          nd=11
C00127                          pnew=9+pfrd
C00128                          continue
C00129                          delta=ue(ph/emp-1.)
C00130                          if(ph<.gt.0.7)go to 32
C00131                          if(alpa$zt.0.01)go to 33
C00132                          dpdx*3(pnew-p)/dx
C00133                          go to 9
C00134                          pold=p
C00135                          go to 7
C00136                          first time thru
C00137                          700 pold=pnew
C00138                          rh=y=yold+(1.*delta*alpha)
C00139                          700 dpdx=1(pnew-p)/dx
C00140                          if(pnewlt.9)go to 330
C00141                          if(lz.ge.0.1)go to 210
C00142                          write(6,21)nd
C00143                          rewrite(6,20)alpa,pold,ywp,dpdx,pay
C00144                          210 ylpwp
C00145                          continue
C00146                          if(zinf-1214+100,100)
C00147                          write(6,15)lt
C00148                          format1:1,20x,20hwp does not converge=115,11h iterations
C00149                          if(zinf-152)
C00150                          if(lz,nf=0)go to 31
C00151                          dpdx=dpdx
C00152                          lzf=664
C00153                          go to 99
C00154                          write(6,34)
C00155                          format1:40x,24hgetting ready to blow up
C00156                          lzf=664
C00157                          300 dpdx=dpdx
C00158                          delta=delta/10.
C00159                           id=1
C00160                           if(lz,xe=0)write(6,301)delta,pnew,pui,alpa,rt111
C00161                          301 format1:29i'had iteration--delta cut to 1e-15,71
C00162                          num=10
C00163                           k=kat-1
C00164                           if(kat-151,22,23
C00165                           6 continue
C00166                          ywp has converged=return to sbr two
C00167                          number=number+1
C00168                          if(lz,eq=0)go to 150
C00169                          write(6,20)dpdx,pnew,alpa,ywp
C00170                          continue
C00171                          if(lz,le=lx)go to 160
C00172                          if(number-le,lu=1,1go to 160
C00173                          if(lux=3)
C00174                          write(6,21)lx,lu,lt,number,1out
C00175                          format1:150,10x,0(10)
C00176                          write(6,20)dpdx,pnew,alpa,ywp
C00177                          format1:15,71
C00178                          write(6,20)yw,dywx,pnew,p,ox,8
C DO 50 J=1, NSPC
C CONVERT DATA FROM KCAL/MOLE TO Hz/SEC
C DLL(J)*(K*KCAL*DMF(J))/NMOLE(J)
 DO 50 K=1, K
 DO 50 L=2, L
 I=1+5*(K-1)+30*(J-1)
 HF(I+K-2)*HFA(L)
 50 CONTINUE
C CONVERT DATA FROM FTZ/SEC TO Hz/SEC
 HF(L+K-2)*CFKFSM*HFA(L)
50 CONTINUE
 IF(MA.GT.1) WRITE(6,2)HF
 RETURN
END
SUBROUTINE BL(RE,RT,WWW,FW,TG,DF)
C CALCULATE ALCUINARY LAYER LOSS PER CPFA 178
C RE = RADIUS OF EXIT INCHES
C RT = RADIUS OF THROAT INCHES
C WWW = CHAMFER PRESSURE PSI
C TG = CHARACTERISTIC VELOCITY FT/SEC
C WWW = MOLECULAR WEIGHT
C DF = TEMPERATURE GAS TOTAL DEG R
C DF = BOUNDARY LAYER LOSS LB FORCE
REAL F,WWW,RE
WWW=ALCUC(RE/RT)**2
M=1.461-.912*AR+.06832*AR**2+.006901*AR**3
M=1.461
PR=(1.2*M)**2
RCD=(1.2*M)**2/(1.1+(1.2*M)**2)**6)
VIS=6.66*10***(TG+.6)*WWW**.5
R=.66**^(1/RT)*VIS
F(1)=.003+0.03786*WWW-.001786*WWW**2
F(2)=1.18+.09405*WWW+1.1449*WWW**2+.008333*WWW**3
F(3)=57.7+.6714*WWW-5.714*WWW**2
TREF=F(3)+57.7+1/(1/(1.4**R)+1/(SINV(6.283/360.*F(3)+1)**3)
TREF=F(1)+TREF
DF=.283*PR*WWW*F(3)*TREF*WWW(6.283/360.,F(3)+1)**3)
RETURN
END
* ELT BUMP.1.720512. 51657.1

FUNCTION BUMP(L)

C ROUTINE TO SET CONVERGE CRITERIA IN DYNAMICS

DIMENSION A(5)

DATA A/2.,6.,8.,9.,95/

C IF(L.GT. 5)GO TO 5

BUMP=4(L)

RETURN

C 5 BUMP=1.C

RETURN

END
* ELT COLD:1 720512: 5:065 1

000001  SUMCUTINE COLD(PC,W,CF,RTXN,PERBEL,WM,TG,TIME,OF)
000002  COMMON/FC/P,PER
000003  REAL (IF)
000004  CSTAR=PC*(RT**2+3,14159)*32,174/W
000005  A=RE**2/RT**2
000006  B=0,866
000007  CF=3,349
000008  PE=3,755-.197097*4-.003062*A**2-1,589*K-5*A**3
000009  PE=EXP(PE)
000010  DFI=(1.-PE)***(1,286)
000011  CF=(8*C/D)***(5*PE*A)
000012  FC=PC*CF*(RT**2+3,14159)
000013  ISP=CSTAR*CF/32,174
000014  CALL BL(RE,RT,PC,CSTAR,WM,TG,DF)
000015  CALL DIV(RE,RT,DFV)
000016  OFISP=DFY
000017  DDIV=1.-DFV)*ISP
000018  DISP=ISP-OFISP-DDIV
000019  FUEL=ISP
000020  WRITE(6,100) TIME,DISP,FUEL,DDIV,OFISP,ISP
000021  100 FORMAT(1,1,30H THE ISP AND THRUST AT TIME = ,F9,3, 4H [0,F9,2]
000022  11M SEC AND *F7,2, 4H LBS/12H DIV LOSS = ,F6,3,5H SEC/23H BOUND
000023  LAYR LOSS = ,F6,5,5H SEC/12H THEO ISP = ,F9,2)
000024  IF(PE,LT,0,1) GO TO 51
000025  ARAN=1RT)**2
000026  AR Rita A R A A
000027  S=211,75*2,07*A-10,94*A**2,94066*A**3
000028  S=51*F(30,0)**5
000029  WS=(PE-1,0)*((1.0-PE)**1.0)
000030  FF=5+WS
000031  WM=WS+WS
000032  FF=EFFFFLE
000033  NSF=FFFF
000034  WRITE(6,105)SUB,DIS
000035  105 FORMAT(1,5H DUE TO MASS ADDITION IN THE SUPERSONIC REGION THE THR
000036  DST/0H WAS INCREASED TO ,F7,2,4H LBS/26H THE DELIVERED ISP IS N
000037  20W = F7,2,5H SEC)
000038  51 CONTINUE
000039  RETURN
000040  END
00003088
EX35=.0
DO 35 J=1,NSPC
EX32=EX32*X(J,1)+YSPEC(J,1+1)
36 EX33=EX33*X(J,1+1)+YSPEC(J,1)
37 EX34=EX34+W(J,1)+1*YSPEC(J,1)
35 EX35=EX35+W(J,1)+1*YSPEC(J,1)
GO TO 36
39 EX37=5*EX30+SMALL(1)*EX31*(SMALL(1)+EX34)
38 EX39=5*EX31*(SMALL(1)+EX35)
GO TO 39
40 RH(1)=(EX9*N(1)+1)*EX14+*W(1)*EX10*m(1)+EX25*EX30)/EX1
GO TO 30
41 IF(CHEP.EQ.2)Go to 300
42 DIFFUSION (SPECIES)
DO 240 J=1,NSPC
240 YSPEC(J,1)=WC(J,1)*(EX6*YSPEC(J,1)+EX13*YSPEC(J,1))
1*EX7*YSPEC(J,1))EX1
GO TO 160
43 CONTINUE
44 DIFFUSION (ELEMENTS)
DO 40 J=1,NEL
40 HAPHA(J,1)*EX6+ALPHA(J,1)*EX13*ALPHA(J,1)*EX7+ALPHA(J,1+1))EX1
GO TO 40
45 CONTINUE
46 SOLVE EXPLICIT EQUATIONS ON AXIS
IF(PS1(1).GE.DELPSI) Go to 21
47 CONTINUE
48 NORMATUM
IF (U(NORM,VE.<.0) GO TO 133
EX16=6*MU(1)DX/DLS
GO TO 78
133 EX16=6*MU(1)DX/RUTU(1)/DLS
78 CONTINUE
N(V1)EX14(U(2)=U(1))=U(1)
CONTINUE
49 ENERGY
IF (ABS(SIGMA(1)=-1,0).GT.001) 138,39,39
39 EX4=1,0.1.0/SIGMA(1)=TAUT(2)-TOUT(1)
GO TO 41
38 LEX11,9
141 IF(A4G(XE(L1)=1,0)-.01442,43,43
37 EX50=EX50+*W(J,1)*YSPEC(J,2)
36 EX31=EX31*1.0+1.0)*SIGMA(1)*EX30+SMALL(1)
GO TO 44
42 EX51=U
44 KH(1)=XH(1)+1*EX16*(KH(2)=Z(H1))/SIGMA(1)*EX40*EX51
47 CONTINUE
48 IF(CHEP.EQ.2)Go to 400
49 CONTINUE
50 DO 250 J=1,NSPC
51 YSPEC(J,1)*YSPEC(J,1)*WDT(J,1)*EX16+(YSPEC(J,1)-YSPEC(J,1))/SCH
GO TO 22
52
C 403 CONTINUE
C DIFFUSION (ELEMENTS)
DO 20 J=1,NEL
20 RHO(J)=RHO(J-1)+EX16*XLE(J)*(ALPHA(J)+ALPHA(J+1))
GO TO 22
C 21 RHO(J)=U(J)
C ENERGY
ON(1)=ON(1)
IF(INIC.EQ.2)GO TO 500
C DIFFUSION (SPECIES)
DO 25 J=1,NPC
25 CONTINUE
GO TO 22
C 300 CONTINUE
C DIFFUSION (ELEMENTS)
DO 23 J=1,NEL
23 RHO(J)=EXP(J)
C EDGE CONDITIONS
22 CONTINUE
RHO(JPSI)=U(JPSI)
RHO(JPSI)=EXP(JPSI)
DO 27 J=1,NPC
27 CONTINUE
GO TO 33
C 600 CONTINUE
DO 24 J=1,NEL
24 RHO(J)=RHO(J-1)+ALPHA(J)*U(J)
C 303 CONTINUE
DO 28 J=1,NPSI
28 RHO(J)=RHO(J-1)+DX*DPX/RUT(J)
C 50 CONTINUE
IF(LZ.EQ.0)RETURN
C 55 WRITE(6,2)
C 2 WRITE(6,1)S,SW,TR,L,R,M,SM,SH,SW,BR,RT
C 20 WRITE(6,2)RU(I)+U(I),RH(I),H(I),TAU(I),T(I),RT(I),SMALL(H(I))
C 1 FORMAT(8E13.7)
RETURN
END
FUNCTION TO COMPUTE DUCT PRESSURE RATIO
GIVE 4FL/0

FUNCTION CPFR(F4L0)

X = 1.0
X2 = X1
X3 = X2
C = CPFR + \text{X3}
1.0 \times \text{X3}
RETURN
END
PROGRAM TO FIND AFHZ AT OF=4,6,8 AND HIGH AND LOW PT

DC 2C 1=1.22
IF ( (41)+GT. 2.5) GO TO 30
AF(1,1)=1+EXP(130.043-292.214*AR(1)+247.014*AR(1)**2-93.2601*AR(1)**3)
1=5
IF (AF(1)+GT.2.5) AF(1,1)=1
A(2,1)=1+EXP(22.9227-23.3745*AR(1)+9.9097*AR(1)**2-2.00204*AR(1)**3)
IF (AF(3)+GT.16.1877-11.9281*AR(1)+4.0312*AR(1)**2-6.69539*AR(1)**3)
IF (PT(1)+LT. 500.) GO TO 24
DC 22 1=1.22
IF ( (41)+GT. 3.0) GO TO 31
AF(4,1)=1+EXP(40.638-63.057*AR(1)+38.100*AR(1)**2+11.0758*AR(1)**3)
1=5
IF (AF(1)+GT.2.5) AF(4,1)=1
AF(5,1)=1+EXP(17.8703-12.9124*AR(1)+3.9970*AR(1)**2-6.93467*AR(1)**3)
1=5
AF(6,1)=1+EXP(15.1594-4.0014*AR(1)+5.2358*AR(1)**2-3.03787*AR(1)**3)
2=2
CONTINUE
24
DC 21 1=1.22
AF(4,1)=AF(1,1)
AF(5,1)=AF(2,1)
AF(6,1)=AF(3,1)
IF ( (AR(1)+GT. 2.0) GO TO 32
AF(1,1)=1+EXP(32.7219-1018.39*AR(1)+1035.75*AR(1)**2+846.680*AR(1)**3)
1=7
AF(1,1)=EXP(10,1731-9.4258*AR(1)+2.67533*AR(1)**2+1.460734*AR(1)**3)
1=5
DC 110 1=1.6
IF ( (AR(1)+GE. AF(1,1)) GO TO 111
DC 112 J=2,22
1=0
I=0
I=0
AF(1,1)=1+EXP(-1.2-4.3*AR(1)+.8*AR(1)**2)
G(AF(1,1)-AF(1,1)) AR( J-1)-AR( J)
C(AF(1,1)-AF(1,1)) AR( J-1)-AR( J)
D(AF(1,1)-OMN(1,1,1)) AR( J-1)-AR( J)
B'S'MN'(J-1)-DAR( J-1)
AFRZ(1,1)=(B-A)/C=0
CONTINUE
112
GOTO 110
CONTINUE
AFRZ(1,1)=5.0
GOTO 110
CONTINUE
RETURN
END
SUBROUTINE DIVRE,RT,DVL

C DIVERGENCE LOSS PER CPIA 178 GAMMA =1.2 TAN CONICURED NOZZLES
C RE RADIUS EXIT
C RT THROTTLE RADIUS
C DVL DIVERGENCE LOSS PERCENT/100
C ASSUMES VACUUM CONDITION SEE PAGE 6 FIG A=4 CPIA 178

AR=ALOG((RE/RT)**2)
DVL=.930+.02138*AR-.0024279*AR**2+.0001325*AR**3
RETURN

END
SUBROUTINE DYNAM

* * * * * * * DYNAMICS MODEL FLOW CALCULATIONS *

THIS SECTION PERFORMS THE FLOW CALCULATIONS ASSOCIATED WITH
EACH CONNECTOR. THE CALCULATIONS ARE DEPENDENT ON THE GIVEN
INITIAL CONDITIONS FOR THE VOLUMES AND THEIR RESPECTIVE
CONNECTORS.

THE GIVEN INITIAL CONDITIONS FOR THE VOLUMES ARE

1. NODE NUMBER
2. VOLUME SIZE
3. TEMPERATURE
4. PRESSURE
5. MIXTURE (1: OXYGEN BY WEIGHT)
6. COMBUSTOR OR NOT COMBUSTOR
THE GIVEN INITIAL CONDITIONS FOR THE CONNECTORS ARE

1. CONNECTED NODE 1
2. CONNECTED NODE 2
3. ORIFICE DISCHARGE COEFF OR DUCT LENGTH
4. CROSS SECTIONAL AREA
5. RESTRICTION-TYPE (ORIFICE OR DUCT)

DYNAMICS MODEL SUBROUTINE

COMMON/MODF/NOEDC,CONN,.WOUTC,NOINJ(24),LPRTF,
INVL(60),ICMP(60),PRES(60),TEMP(60),XMIX(60),VOL(60),
2INJ(6),ITYPE(60),CAREA(60),CDEFF(60),DLEN(60),TJMON(60),
SINOFF(6),TOP(60),TMC(60),TJMON(60),INCUT,TPSTST
00103 32* 49STIME, SPKLF, SPKAP, SPARKP, SPARKE, DC, NCHECK, TIME, I, EVOL, DPRES
00104 33* COMMON/JFCT/VT, PD, FM, PSTAT
00105 34* COMMON/CDFP/NCOMBU, NLINE
00106 35* COMMON/ICT, DYN, TART, APIMP, ACV
00107 36* DIMENSION SWC(T), SIS(T), SADMT(40), C(40), Q(16), SMDOT(60)
00108 37* DIMENSION WEXT(20), RMINX(20), GNEW(20), TNEW(20), PNEW(20)
00109 38* EQUIVALENCE (CQF1(1), DLEN(1))
00110 39* DIMENSION W(60), OCDEF(60)
00111 40* DATA BLK(3), PRTP(3), PRTC(3), PRY(3)
00112 41* 1 'COM', 'EUBT', 'OR' /
00113 42* DIAGNOSTIC* THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00114 43* IF (TIME-EX=0.0) GO TO 1900
00115 44* 1900 SET COMBUSTOR/PILOT CODES
00116 58* DC 1311 I*1=I, NNODE
00117 59* IF (ICOMB(1)=EQ.1 .OR. ICOMB(2)=EQ.2) ICVAL=1
00118 60* IF (ICOMB(1)=EQ.3 .OR. ICOMB(1)=EQ.4) IPVOL=1
00119 61* 1311 CONTINUE
00120 62* 1978 CONTINUE
00121 63* LOC5 1-PU DYNAMICS MODEL ONCE FOR EACH CONNECTOR AND
00122 64* PERFORM THE FLOW CALCULATIONS BASED UPON RESTRICTION
00123 65* TYPE AND NOT ON COLD GAS CRITERIA.
00124 66* 1900 CONTINUE
00125 67* IF ICOMB(1)=2 THEN VOLUME IS A COMBUSTER
00126 68* OR NOT, IF ICOMB(1)=3 THEN VOLUME IS A PILOT
00127 69* OR NOT, IF ICOMB(1)=4 THEN VOLUME IS AN IGNITED PILOT
00128 70* SET INDEXES WHICH WILL DETERMINE IF VOLUME IS A COMBUSTER OR NOT, IF IGNITION HAS OCCURRED OR NOT, AND THE RESTRICTION TYPE
00129 71* IF ICOMB(1)=2 THEN VOLUME IS A COMBUSTER
00130 72* IF ICOMB(1)=3 THEN VOLUME IS A PILOT
00131 73* IF ICOMB(1)=4 THEN VOLUME IS AN IGNITED PILOT
00132 74* IF (ICOMB(1)=0) THEN VOLUME IS NOT A COMBUSTER
00133 75* IF ICOMB(1)=1 THEN VOLUME IS A COMBUSTER, NOT IGNITED
00134 76* IF ICOMB(1)=2 THEN VOLUME IS AN IGNITED COMBUSTER
00135 77* IF ICOMB(1)=3 THEN VOLUME IS A PILOT, NOT IGNITED
00136 78* IF ICOMB(1)=4 THEN VOLUME IS AN IGNITED PILOT
00137 79* IF ICOMB(1)=0 THEN RESTRICTION TYPE IS PRIMARY
00138 80* IF ICOMB(1)=1 THEN RESTRICTION TYPE IS CIRCULAR
00139 81* IF ICOMB(1)=2 THEN RESTRICTION IS THE THROAT
00140 82* IF ICOMB(1)=3 THEN RESTRICTION TYPE IS CIRCULAR
00141 83* IF ICOMB(1)=4 THEN RESTRICTION IS THE THROAT
00142 84* IF (ICOMB(1)=0) THEN RESTRICTION TYPE IS PRIMARY
00143 85* IF (ICOMB(1)=1) THEN RESTRICTION TYPE IS CIRCULAR
00144 86* IF (ICOMB(1)=2) THEN RESTRICTION IS THE THROAT
00145 87* 1900 CONTINUE
00146 88* DATA P/3.1415926/
STARTING GUESS AT 4F

IF (IADV*EQ.0) GO TO 1369

CONTINUE

IF(1) CONTINUE

DO DP*PRES(A, CHECK)

ZERO SUMMING VECTORS + GET START VOL MASSES

CALCULATE THE PRESSURE RATIO ACROSS CONNECTOR J

AND DETERMINE THE DIRECTION OF FLOW

CALC CROSS SECTIONAL AREA

ADD CONTACT RESISTANCE TO TIMON(J)*TIMOFF(J)*TIMPNI(J)*TIMCLS(J)

THE TEST FOR EQUALITY BETWEEN NON-INTEGER MAY NOT BE meaningful

THE TEST FOR EQUALITY BETWEEN NON-INTEGER MAY NOT BE meaningful

THE TEST FOR EQUALITY BETWEEN NON-INTEGER MAY NOT BE meaningful

CHECK IF COMPLETELY OPEN

ADD L.C

IF TIME + LT+ TIMOFF(J)*GO TO 50

ADD 0.0

IF TIME + LT+ TIMOFF(J)*GO TO 50

ADD 0.0

IF TIME + LT+ TIMCLS(J)*TIMOFF(J)*GO TO 50
FIT FOR CLIPPING

**DIAGNOSTIC**
The test for equality between non-integers may not be meaningful.

IF(TIMCLS(j).EG.0.0)ADD=0.0

AREA=ADD*AREA(j)

DIAM=C0*SGRT(AREA)

**CALC PRESSURE RATIO**

PRST=PRST(IVOL1)/PRST(IVOL2)

MNX3=IVOL1

MNX4=IVOL1

**TEST FOR HIGHER PRESSURE**

IF(1.0-PRST) 101.102.102

101

MNX3=IVOL1

MNX4=IVOL2

PRST=1.0/PRST

102 CONTINUE

PRST=PRST(NDX3)+PRST(NDX4)

MNX=ICOMP(MNX3) + 1

MNX2=1RTYPE(j)*1

GO TO HOT OR COLD GAS CALCULATION

IF(PDIF .LT. 1.0E-3) GO TO 150

GO TO (2.2.200.250.250.250.1000).NDX1

GO TO ORIFICE OR DUCT CALCULATIONS

GO TO (3.3.350.300).NDX2

IF NO PRESSURE DIFFERENCE, SET VALUES TO ZERO

WRT=0.0

RMT=1.0

GO TO 375

COLD GAS CALCULATIONS

USE WA TO AVOID DOUBLE USE OF INDEX

C=RMIX(NDX3)

CALCULATE ORIFICE HOLE W

RMHT=C1/(C2*WA1+C3+(C4=W2))

CALCULATE GAS CONSTANT

RCNST=C5/RMHT

GM=1.4

GO TO 133

HOT GAS CALCULATIONS

GO TO 133

CONTINUE

RMIX=RMIX(NDX3)/(1.0-RMIX(NDX3))

RMHT=SHEAT(CF)

RMHT=FPOLYT(OF)

RCNST=C5/RMHT

GO TO 103

PROCESS AS ORIFICE
CONTINUE

CALCULATE CHOKE PRESSURE RATIO

w1 = GM + 1 - C
w2 = 2.6 / /w1
w3 = GM - 1 - C
CPR = w2 ** (GM/WA3)

IF (PRESR>GPS) 301, 302, 302

CONTINUE

CHOKED

C = SORT(w2 ** (WA1/WA3))
GO TO 303

UNCHOKED

C = SORT(2, WA3)
FFP = SORT((PRESR ** (2.0/GM)) - (PRESR ** (WA1/GM)))

w3 = AREA * PRES(NDX3)

w3 = RCYST * TEMP(NDX3)

w3 = SORT(w2 / WA3)

ADMIT = w3 * WA3 * F * COF / PDIF

WDF = ADMIT * PDIF

GO TO 375

CONTINUE

PROCESS A DUCT

CONTINUE

C = NO 356 = 1, 3

CALCULATE 4PL/D

FFP = FPC * DLEN(J/DIAM

CALCULATE CHOKE PRESSURE RATIO

CPR = CPRF(FPP)

IF (PRESR > CPR) 351, 352, 352

CONTINUE

C = CHOKED

w4 = FLOW(FPP)

GO TO 353

CONTINUE

C = UNCHOKED

A #FLOW (PRESR, FPP)

w4 = AREA * PRES(NDX3)

w4 = SORT(CONST * TEMP(NDX3)) * PDIF

ADMIT = w4 * WA3 * COF / WA7

WDF = ADMIT * PDIF

FPU = CM * SQRT (RFMT) * (TEMP(NDX3) ** 0.6)

REN = NW(NDX3) * DIAM / AREA * FPU

IF (210 < REN) 354, 355, 355

CONTINUE

FPC = FPC * FEN

GO TO 376

CONTINUE

FPC = 44.0 * FEN

CONTINUE

C = CALCULATE ADMITTANCE DEPENDENT VARIABLES

CONTINUE

IF (NIN * E = 0.0) GO TO 385

CONTINUE

IF (NDX * A = 13.0) GO TO 385

GO TO 386

CONTINUE

GO TO 385
02412  266  C WT IS USED AS INPT BY THE INJECTION MODEL
02413  267  3) F = 1/(FM*(WT=1000)) = WDOT*(RMIKX3)/WT
02414  268
02415  269  CONTINUE
02416  270  IF [MTYPE(J),NE,2] GO TO 388
02417  271  C W IS USED AS INPT BY COLD, THE COLD PERFORMANCE MODEL
02418  272  WFLK=WDOT
02419  273  388 CONTINUE
02420  274  C
02421  275  C + ADD = TEMP3AD3X3+DOT(3.5-3.26*RMIKX3))
02422  276  QDOT3AD3X3龊DOT(DX3) = AD
02423  277  QDOT(DX4)=QDOT(DX4) + AD
02424  278  RWDOT(DX3)*SBDOT(DX3) - WDOT
02425  279  RWDOT(DX4)*SBDOT(DX4) + WDOT
02426  280  RADMITIVE(DX3)=SADMITIVE(DX3) + AD
02427  281  RADMITIVE(DX4)=SADMITIVE(DX4) + AD
02428  282  XORRMIKX(DX3)WDOT
02429  283  RWDOT(DX3)*SBDOT(DX3) - ADD
02430  284  RWDOT(DX4)*SBDOT(DX4) + ADD
02431  285  560 CONTINUE
02432  286  C
02433  287  C CALCULATE TEMP CHANGES IN ACCUMULATION VOLUMES
02434  288  C
02435  289  660 CONTINUE
02436  290  WJCNT=WJOUNT
02437  291  IF(WJCOUNT.EQ.0) GO TO 9000
02438  292  C JXK=150.0
02439  293  C N=1100 IN ADD
02440  294  C X=F=M(1)+DEL*SWDOT(I)
02441  295  RMIKXN(I)=RMIKX(I)+M(L) + DELT*SMAPT(DX1)/NVEW(1)
02442  296  QAEW(I)=Q(I) + DELT*QDOT(I)
02443  297  IF([C0MBL])NE.2 AND.1.COMBI.NE.4)GO TO 1090
02444  298  C HOT COMBUSTOR EQUATION
02445  299  C + [WM]F=PM bat(O)F
02446  300  TAEM=TEMPRES(I).OF
02447  301  GC TO 1095
02448  302  GC TO 1095
02449  303  C NON-COMBUSTEUR EQUATION
02450  304  C 1090 TAEM(I)QAEW(I)/NVEW(I)+3.5-3.26*RMIKXN(I))
02451  305  RPMT = 64.5/98/12.016/RMIKX(I)+52.0*(1.0-RMIKX(N(I))
02452  306
02453  307  1095 CONTINUE
02454  308  C [G(I)+V(1)]*RPM(T)=C5*TWELV(E)
02455  309  PEWELV(I)=PRES(I) + DELT*SWDOT(I)/C(1)
02456  310  C[OCK]/SAAMIT(I)
02457  311  C DIAGNOSTIC THE TEST FOR EQUALITY BETWEEN NON-INTAGERS MAY NOT BE MEANINGFUL
02458  312  IF[CHECK=EQ.0.0] CHECK=1.0,E-15
02459  313  CHECK=CHECK*CHECK
02460  314  C CHECK=AMAG(CHECK)*CHECK
02461  315  C GO TO 1250
02462  316  C IF(MDEL .LE. 1.0) CHECK=1.0 .GO TO 1250
02463  317  C NOT STABLE: RESIST AND TRY AGAIN
02464  318  GC TO 687
02465  319  C WAS STABLE: GO ON
02466  320  C 1250 CONTINUE
02467  321  IF(MDEL .GE. 1.0) GO TO 1387
02468  322  IF [INSPL.EQ.1.0.OR.EQ.1.0] GO TO 1387
02469  323  C DELT=CHECK*BDMP(L) .GO TO 1387
02470  324  C GO TO 687
IGNITION MODEL

CHECK IF IGNITION HAS ALREADY OCCURRED

IF (ICOMB(IPVOL).EQ.1.AND.ICOMB(IPVOL).EQ.4) GO TO 890
IF (NSPK.EQ.GO) GO TO 898
IF (ICOMB(IPVOL).EQ.2) GO TO 899

SPARK IGNITION MODEL

USE PILOT CONDITIONS OR COMBUSTION CONDITIONS?

IF(IPVOL.EQ.1) GO TO 810
PLUGP=PRE(IPVOL)
IF (RMI(IPVOL).GT.999) GO TO 1829
PLOOF=RMI(IPVOL)/(1.-RMI(IPVOL))
GO TO 825

810 PLUGP=RMI(IPVOL)
IF (RMI(IPVOL).GT.999) GO TO 1829
PLOOF=RMI(IPVOL)/(1.-RMI(IPVOL))
GO TO 825

1825 PLOOF=PLUG.
CONTINUE

CONTINUE IF SUPPLIED POTENTIAL GREATER THAN BREAKDOWN POTENTIAL??

P(CP)PLUGP=SGAP
REQV=.52*((1:022*(ALOG(IPG)))*1.029*(ALOG(IPG))**2)**
11.0292*(ALOG(IPG)+34)*.0044*(ALOG(IPG))**4)
IF (REGV.GT.174) REGV=20.
REGV=EXP(REGV)
IF (REGV-80.830,899
CONTINUE

REGV=EXP(REGV)
IF (REGV.GT.830) 830,899
CONTINUE

IF SPARK ENERGY SUFFICIENT FOR IGNITION?

830 IF (PLOOF.LT.0.4 OR PLOOF.GT.590.) GO TO 895
IF (REGE-EXP(0.499+(8:7104*(ALOG(PLOOF))))*1.9418*(ALOG(PLOOF))
1**2)**(1.512*(ALOG(PLOOF))**3))
IF (REGE-EXP(400,840,891
840 IF (IPVOL.EQ.0) GO TO 893
890 WHITE (.450)
IF (ICOMB(IPVOL)=4
TEMP(IPVOL)=TR(PRES(IPVOL).*PLOOF)
RWT=IPVOL-(PLOOF)
1**164.**2*(2.01*RMI(IPVOL))**(32.4**(1.-RMI(IPVOL))))
PRES(IPVOL)=W(IPVOL)+C5*TEMP(IPVOL)/(IPVOL-IPVOL)*RWT
890 WHITE (.450) TEMF(IPVOL) PREO(IPVOL)
CONTINUE

895 IF (RMI(IPVOL).GT.999) GO TO 899
IF (ICOMB(IPVOL).EQ.1.-RMI(IPVOL))
IF (ICOMB(IPVOL).LT.0.4 OR ICOMB(IPVOL).GT.590.) GO TO 894
30741 445+ \[\text{AREG} = 17.427 - (14.5603 * (\text{ALOG(OFCOMB)}) + (5.8133 * (\text{AI} \text{OG(OFCOMB)})) \]
30742 446+ \[\text{I} = \text{I} + 2 \times \{ (1 - 1.1581 * (\text{ALOG(OFCOMB)}) - 3) \times \{0.991 * (\text{ALOG(OFCOMB)}) - 441 \}
30743 447+ \text{IF}(\text{AREG} < \text{PRESICVAL}) \text{GO TO 893} \]
30744 448+ \text{GOTO 899} \]
30745 449+ \text{GO TO 899} \]
30746 450+ \text{IF}(\text{IPVCL} < \text{F0} \text{G0}) \text{GO TO 892} \]
30747 451+ \text{GOTO 899} \]
30748 452+ \text{GO TO 899} \]
30749 453+ \text{GO TO 899} \]
30750 454+ \text{GO TO 899} \]
30751 455+ \text{WRITE (6.504)} \]
30752 456+ \text{WRITE (6.504)} \]
30753 457+ \text{WRITE (6.504)} \]
30754 458+ \text{WRITE (6.504)} \]
30755 459+ \text{WRITE (6.504)} \]
30756 460+ \text{WRITE (6.504)} \]
30757 461+ \text{WRITE (6.504)} \]
30758 462+ \text{WRITE (6.504)} \]
30759 463+ \text{WRITE (6.504)} \]
30760 464+ \text{WRITE (6.504)} \]
30761 465+ \text{WRITE (6.504)} \]
30762 466+ \text{WRITE (6.504)} \]
30763 467+ \text{WRITE (6.504)} \]
30764 468+ \text{WRITE (6.504)} \]
30765 469+ \text{WRITE (6.504)} \]
30766 470+ \text{WRITE (6.504)} \]
30767 471+ \text{WRITE (6.504)} \]
30768 472+ \text{WRITE (6.504)} \]
30769 473+ \text{WRITE (6.504)} \]
30770 474+ \text{WRITE (6.504)} \]
30771 475+ \text{WRITE (6.504)} \]
30772 476+ \text{WRITE (6.504)} \]
30773 477+ \text{WRITE (6.504)} \]
30774 478+ \text{WRITE (6.504)} \]
30775 479+ \text{WRITE (6.504)} \]
30776 480+ \text{WRITE (6.504)} \]
30777 481+ \text{WRITE (6.504)} \]
30778 482+ \text{WRITE (6.504)} \]
30779 483+ \text{WRITE (6.504)} \]
30780 484+ \text{WRITE (6.504)} \]
30781 485+ \text{WRITE (6.504)} \]
30782 486+ \text{WRITE (6.504)} \]
30783 487+ \text{WRITE (6.504)} \]
30784 488+ \text{WRITE (6.504)} \]
30785 489+ \text{WRITE (6.504)} \]
30786 490+ \text{WRITE (6.504)} \]
30787 491+ \text{WRITE (6.504)} \]
30788 492+ \text{WRITE (6.504)} \]
30789 493+ \text{WRITE (6.504)} \]
30790 494+ \text{WRITE (6.504)} \]
30791 495+ \text{WRITE (6.504)} \]
30792 496+ \text{WRITE (6.504)} \]
30793 497+ \text{WRITE (6.504)} \]
30794 498+ \text{WRITE (6.504)} \]
30795 499+ \text{WRITE (6.504)} \]
30796 500+ \text{WRITE (6.504)} \]
30797 501+ \text{WRITE (6.504)} \]
30798 502+ \text{WRITE (6.504)} \]
30799 503+ \text{WRITE (6.504)} \]
30800 504+ \text{WRITE (6.504)} 

**LCC**

*BYOLING COMPANY*
01053  503*  907 FORMAT (' NEW PILOT TEMP IS ',F8.1,' NEW PRESSURE IS ',F8.1)
01054  504*  908 FORMAT ('C',16X,'THE FLAME HAS QUENCHED IN THE COMBUSTOR!')
01055  507*  909 FORMAT ('D',6X,'SPARKING HAS BEEN COMPLETED')
01056  508*  950 FORMAT ('VOL NO.',1X,'PRESSURE',6X,'TEMP',6X,'OK',/)
01057  509*  1  25X,('PSI'),7X,('R')/
01058  510*  951 FORMAT('X',14X,'F11.2;F10.3')
01059  511*  952 FORMAT('X',14X,** DYNAMICS RESULTS **;F10.6;/)
01060  512*  953 FORMAT ('C',6X,'VOL',114.2X,'PRES CHANGES',F10.3, 'PSI PER MB')
01061  513*  END

END OF LCC 1106 FORTRAN V Compilation. 7 *Diagnostics* Messages

Phase 1 Time 00:00:094
Phase 2 Time 00:00:194
Phase 3 Time 00:00:443
Phase 4 Time 00:00:168
Phase 5 Time 00:01:261
Phase 6 Time 00:01:166

Total Compilation Time = 00:15:231
Dynam Symbolic Dynam Code Relocatable

(FNS) 12 JUN 72 19102107 0 00347872 14 $14 (DELETED)
(FNS) 12 JUN 72 19102107 1 00345926 40 $1 (DELETED)

IV Computing Company
OC0059 1=HF(2,J,1)/T+HF(3,J,1)* (ALOG(T)=1.1)
OC0060 51 CONTINUE
OC0061 FRT(IG)X.G
OC0062 IF (IP.GT.1) WRITE(6,3) (FRT(J),J=1,NGAS)
OC0063 3 FOM &FRT(16X,7)-FOM FROM FC/(8E15,7))
OC0064 DO 14 I=1,NHEAC
OC0065 JPA=JRD(J4)
OC0066 JRB=JRD(J41)
OC0067 JRA=JRD(J2)
OC0068 NLFHT(J)=FRT(JPA)+FRT(JPB)-FRT(JRA)-FRT(JRB)
OC0069 OKC(J)=FXP(DLFHT(J))/QK+OK(J)**.5XOBDY(J)
OC0070 O3=OK(J).EATES(J,J)*T**.0*EATES(J,J)**.0*EXP(RATES(J,J)/T)
OC0071 1=OKHMP(J)
OC0072 OK(J)=OK(J)**.0*QK(J)**.0*OK(J)
OC0073 1F (IP.GT.0) WRITE(6,1) OK(J,J),OK(J,2),OK(J)
OC0074 1 FORMAT(2E15.7)
OC0075 C0076 1B CONTINUE
OC0077 19 DO 28 I=1,N
OC0078 I(J)=0
OC0079 20 DO 25 J=1,N
OC0080 25 K(J,J)=C.
OC0081 CALL FGR
OC0082 RETURN
OC0083 END
\[ a \text{ (6, 5)} = AA - BB \\
\text{C2239} \]

\[ \text{BG} = \text{PL} (2, 1) = \text{PL} (3, 6) = \text{PL} (4, 2) = \text{PL} (5, 2) \\
\text{C2240} \]

\[ \text{BGX} = \text{PL} (6, 1) = \text{PL} (7, 1) = \text{PL} (8, 7) = \text{PL} (9, 2) \\
\text{C2241} \]

\[ \text{BH} = \text{PL} (11, 7) = \text{PL} (12, 6) = \text{PL} (12, 10) = \text{PL} (15, 2) \\
\text{C2242} \]

\[ \text{AB} = \text{PL} (12, 5) = \text{PL} (19, 10) = \text{PL} (17, 10) \\
\text{C2243} \]

\[ \text{BB} = \text{PL} (15, 5) = \text{PL} (17, 1) = \text{PL} (17, 9) \\
\text{C2244} \]

\[ \text{AB} = \text{PL} (6, 6) = AA - BB \\
\text{C2245} \]

\[ \text{AB} = \text{PL} (2, 1) = \text{PL} (3, 1) = \text{PL} (9, 6) = \text{PL} (12, 5) \\
\text{C2246} \]

\[ \text{AA} = \text{PL} (15, 10) = \text{PL} (4, 1) = \text{PL} (12, 10) = \text{PL} (15, 5) = \text{PL} (17, 5) \\
\text{C2247} \]

\[ \text{AB} = \text{PL} (6, 7) = AA - BB \\
\text{C2248} \]

\[ \text{AA} = \text{PL} (11, 1) = \text{PL} (12, 5) = \text{PL} (15, 10) = \text{PL} (15, 5) = \text{PL} (17, 9) \\
\text{C2249} \]

\[ \text{AB} = \text{PL} (16, 2) = \text{PL} (16, 2) = \text{PL} (16, 10) \\
\text{C2250} \]

\[ \text{BB} = \text{PL} (16, 10) = \text{PL} (16, 5) = \text{PL} (16, 5) \\
\text{C2251} \]

\[ \text{AB} = \text{PL} (16, 10) = \text{PL} (16, 5) = \text{PL} (16, 5) \\
\text{C2252} \]

\[ \text{AB} = \text{PL} (1, 7) = \text{PL} (3, 7) = \text{PL} (4, 7) = \text{PL} (16, 1) \\
\text{C2253} \]

\[ \text{AA} = \text{PL} (16, 5) = \text{PL} (16, 10) = \text{PL} (16, 10) \\
\text{C2254} \]

\[ \text{AB} = \text{PL} (2, 6) = \text{PL} (3, 6) = \text{PL} (9, 2) = \text{PL} (10, 9) \\
\text{C2255} \]

\[ \text{AA} = \text{PL} (4, 6) = \text{PL} (16, 5) = \text{PL} (16, 5) \\
\text{C2256} \]

\[ \text{AB} = \text{PL} (9, 7) = \text{PL} (16, 5) = \text{PL} (16, 10) \\
\text{C2257} \]

\[ \text{AB} = \text{PL} (11, 1) = \text{PL} (2, 1) = \text{PL} (3, 1) = \text{PL} (4, 1) \\
\text{C2258} \]

\[ \text{BB} = \text{PL} (9, 6) = \text{PL} (10, 6) = \text{PL} (16, 6) = \text{PL} (16, 10) \\
\text{C2259} \]

\[ \text{AB} = \text{PL} (16, 5) = \text{PL} (16, 5) = \text{PL} (16, 5) \\
\text{C2260} \]

\[ \text{BB} = \text{PL} (16, 10) = \text{PL} (16, 10) = \text{PL} (16, 10) \\
\text{C2261} \]

\[ \text{AA} = \text{PL} (17, 5) = \text{PL} (17, 5) = \text{PL} (17, 5) \\
\text{C2262} \]

\[ \text{AB} = \text{PL} (17, 5) = \text{PL} (17, 5) = \text{PL} (17, 5) \\
\text{C2263} \]

\[ \text{BB} = \text{PL} (17, 5) = \text{PL} (17, 5) = \text{PL} (17, 5) \\
\text{C2264} \]

\[ \text{AB} = \text{PL} (17, 5) = \text{PL} (17, 5) = \text{PL} (17, 5) \\
\text{C2265} \]

\[ \text{BB} = \text{PL} (17, 5) = \text{PL} (17, 5) = \text{PL} (17, 5) \\
\text{C2266} \]

\[ \text{AB} = \text{PL} (9, 6) = \text{PL} (10, 6) = \text{PL} (17, 5) \\
\text{C2267} \]

\[ \text{BB} = \text{PL} (17, 5) = \text{PL} (17, 5) = \text{PL} (17, 5) \\
\text{C2268} \]

\[ \text{AB} = \text{PL} (9, 6) = \text{PL} (10, 6) = \text{PL} (17, 5) \\
\text{C2269} \]

\[ \text{BB} = \text{PL} (17, 5) = \text{PL} (17, 5) = \text{PL} (17, 5) \\
\text{C2270} \]
FUNCTION FLW(PR,F4LD)
FUNCTION TO COMPUTE UNCHOKED DUCT FLOW
PARAMETER GIVEN PRESSURE RATIO * 4FL/D

FLOW= .566764044 - PR**.13283345 - x2*1.2179672 + x3*3.64026368
1
x4+2.650439756 + y1*18299555 + y2*841392345E-4
2
y3*1.141099274E-2 + Y4*.5151692509E-4 + PR*Y5*3.602877874
3
PR*Y2*1.191869795E-1 - x2*Y6*72519993 + PR*Y3*148729387E-3
4
x3*Y7*.5108571395 + X2*Y8*1293878649E-1
0
IF(FLOW.LT.0.0)FLOW=0.0
0
RETURN
0
END
SUBROUTINE FVEL(X,F,DF)
SUBROUTINE TO COMPUTE TOTAL MASS FLOW VALUES FOR NEWTON ITERATION.

COMMON /NJEC2/ B
DATA P/-2.5/

C     A=1.0 + 0.2*X*X
C     CALC DERIVATIVE VALUE
C     F=SQRT(X*X/A) * (A+P) - B
C     RETURN
C     END

********************************************************************************
SUBROUTINE RTNI
********************************************************************************

PURPOSE
TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM F(X)=0
BY MEANS OF NEWTON'S ITERATION METHOD.

USAGE
CALL RTNI(X,F,DERF,FCT,XST,EPAN,IER)
PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.

DESCRIPTION OF PARAMETERS

C     X - RESULTANT ROOT OF EQUATION F(X)=0.
C     F - RESULTANT FUNCTION VALUE AT ROOT X.
C     DERF - RESULTANT VALUE OF DERIVATIVE AT ROOT X.
C     FCT - NAME OF THE EXTERNAL SUBROUTINE USED, IT COMPUTES
C            TO GIVEN ARGUMENT X FUNCTION VALUE F AND DERIVATIVE
C            DERF. ITS PARAMETER LIST MUST BE X,F,DERF.
C     XST - INPUT VALUE WHICH SPECIFIES THE INITIAL GUESS OF
C            THE ROOT X.
C     EPS - INPUT VALUE WHICH SPECIFIES THE UPPER ROUND OF THE
C            ERROR OF RESULT X.
C     IEND - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.
C     IER - RESULTANT ERROR PARAMETER CODED AS FOLLOWS
C            IER=0 - NO ERROR,
C            IER=1 - NO CONVERGENCE AFTER IEND ITERATION STEPS,
C            IER=2 - AT ANY ITERATION STEP DERIVATIVE DERF WAS
C            EQUAL TO ZERO.

REMARKS
THE PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER=2
IF AT ANY ITERATION STEP DERIVATIVE OF F(X) IS EQUAL TO 0,
POSSIBLY THE PROCEDURE WOULD BE SUCCESSFUL IF IT IS STARTED
ONCE MORE WITH ANOTHER INITIAL GUESS XST.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
THE EXTERNAL SUBROUTINE FCT(X,F,DERF) MUST BE FURNISHED
BY THE USER.

METHOD
SOLUTION OF EQUATION F(X)=0 IS DONE BY MEANS OF NEWTON'S
ITERATION METHOD. WHICH STARTS AT THE INITIAL GUESS XST OF
A ROOT X, CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF...
**ELT GRAD:1,729512,51636**

- **SUMMELIKI GRAD(DA/DX,RT,AR)**, **NS,PT,DMIN**
- **DIMENSION DA(DX)2,FR(622)**
- **PROGRAM TO CALCULATE PARTIAL DW/PARTIAL DT PER NASAGR=7240**
- **CURVES AVAILABLE FOR FC 100,500,1000 FRA, OF**
- **ISP WILL BE ASSUMED TO BE FROZEN BELOW 6F2, OF SP WILL BE BASED**
- **ON INTERPOLATION BETWEEN 2 AND 4** (NO KINETIC + VALLE AT 6-+LINEAR)
- **DETERMIN PC'S AT WHICH D(A/DMIN)/(DX/RT)=IF IT IS EVALUATED**
- **DMIN=(A/DMIN)/(DX/RT)=1 IT / (1/RT)*(DH/DT)=(DH/DT)**
- **IN DMIN = OF VALUES --- 4, 6, 8, 4, 6, 8**
- **AT 1000 THE SHO ON 500 AND 1000 PSIA**

```plaintext
DC 5 1*1,22

10 DC 11 J=1,72

20 DMIN(1,J)=EXP(1,189-26.6218*4(J)+90.46*4(J)+373.61*4(J)**2)

30 DMIN(2,J)=EXP(1,4561-24.4715*4(J)+87.604*4(J)**2)

40 DMIN(3,J)=EXP(2,2032-32,509*4(J)+91.3599*4(J)**2)

50 DMIN(4,J)=EXP(-1,6309*4(J)+47675*4(J)**2)

60 IF (PT,GT,500) GO TO 100

70 DC 12 J=1,72

80 DMIN(5,J)=DMIN(1,J)

90 DMIN(6,J)=DMIN(2,J)

100 DC 13 J=1,72

110 DMIN(7,J)=DMIN(3,J)

120 DC 14 J=1,72

130 DMIN(8,J)=DMIN(4,J)

140 DC 15 J=1,72

150 DMIN(9,J)=DMIN(5,J)

160 DC 16 J=1,72

170 DMIN(10,J)=-12.3536*4(J)+3.09*4(J)**2+57.844*4(J)**2

180 1 +57,571*4(J)**4+28.4424*4(J)**5+5,46314*4(J)**6

190 DMIN(11,J)=EXP(-1,79,5636*4(J)+67.5366*4(J)**2)

200 1+23.9566*4(J)**2+62.2884*4(J)**3

210 DMIN(12,J)=EXP(-1,4274-24.8294*4(J)+91.7929*4(J)**2)

220 1+31,704.3*4(J)**4-79,3444*4(J)**5+14,17059*4(J)**6

230 GC 1D 200

240 DC 1D J=1,72

250 DMIN(13,J)=EXP(-1,7247-24,4267*4(J)+7,1552*4(J)**2)

260 1+3,496.44*4(J)**4+36.2536*4(J)**5+4,96114*4(J)**6

270 DMIN(14,J)=EXP(-1,9466-25,9076*4(J)+97,9066*4(J)**2)

280 1+11,830.04*4(J)**4+86,7679*4(J)**5+15,987*4(J)**6

290 DMIN(15,J)=-15222-25.6697*4(J)+92,4376*4(J)**2

300 1+77,8322*4(J)**3+54,1176*4(J)**4

310 DC 2D 11,12

320 GC 2D J=1,72

330 DMIN(16,J)=DMIN(1,J)+DA(DX(1)/RT(12))

340 CONTINUE

350 20 CONTINUE

360 RETURN

370 END
```
C:3059  FIX(K)=IX(KM)
C:3060  203 CONTINUE
C:3061  GO TO 4  +1,ASPC
C:3062  204 YKPL(J,K)=YSPEC(J,KM)
C:3063  122 CONTINUE
C:3064  MPSI=PSI-1
C:3065  MPSI=PSI-1
C:3066  LRL=1
C:3067  GO 126 IF(KF)  
C:3068  126 PSI=PSI-DELPSI
C:3069  998 IF(NEWKE,0)GO TO 2000
C:3070  C C  TEST FOR ZERO SLOPE AT EDGE
C:3071  C C  999 IF(ABS(YPsi-U(MPSI))/U(MPSI)-.001) 1001,1001,1004
C:3072  1001 IF(ABS(YPsi-U(MPSI))/U(MPSI)=.001) 1002,1002,1004
C:3073  1002 IF(ICHNE.EG.2)GO TO 300
C:3074  GO 3J2,1,ASPC
C:3075  IF(YSPEC(J,JPSI)-LT-.001)GO TO 302
C:3076  IF(YSPEC(J,JPSI)/YSPEC(J,JPSI)=1.,GT,.001)GO TO 1004
C:3077  302 CONTINUE
C:3078  GO TO 3J1
C:3079  303 DO 3J4,JPSI
C:3080  IF(ALPHA(J,JPSI)-LT,.001)GO TO 54
C:3081  IF(ALPHA(J,JPSI)/ALPHA(J,JPSI)=1.,GT,.001)GO TO 1004
C:3082  304 CONTINUE
C:3083  GO TO 2000
C:3084  C C  EXPAND MESH
C:3085  1004 MPSI=PSI+1
C:3086  MPSI=PSI+1
C:3087  LRL=1
C:3088  MPSI=PSI+1
C:3089  GO 3JU,JPSI,MP
C:3090  TLAGP(J)=TLAGP(NPSI)
C:3091  U(J)=UKPSEI
C:3092  W(J)=UKPSEI
C:3093  T(J)=TPSEI
C:3094  IF(ICHNE.EG.3)GO TO 303
C:3095  FIX(J)=IX(KPSI)
C:3096  INP(J)=IEP(APS)
C:3097  TLD(J)=TLD(NPSI)
C:3098  MLD(J)=MLD(NPSI)
C:3099  303 CONTINUE
C:3100  DO 3J4,JPSI
C:3101  YSPEC(J,JPSI)=YSPEC(J,JPSI)
C:3102  IF(ICHNE.EG.2)GO TO 30
C:3103  DO 3J6,JPSI
C:3104  IF(ALPHA(J,JPSI)-1,1,ASPC
C:3105  IF(ALPHA(J,JPSI))=ALPHA(J,JPSI)
C:3106  306 CONTINUE
C:3107  GO TO 2000
C:3108  C C  MESH
C:3109  1500 IF(INIT=0)  
C:3110  DELPSI=CELPSEI-DELPSI
C:3111  GO 16U J=1,MINT
**ELT HEATT:1.720512: 51544 : 1**

00001  
00002  
00003  
00004  
00005  
00006  
00007  
00008  
00009  
00010  
00011  
00012  
00013  
00014  
00015  
00016  
00017  
00018  
00019  
00020  
00021  
00022  
00023  
00024  
00025  
00026  
00027  
00028  
00029  
00030  
00031  
00032  
00033  
00034  
00035  
00036  
00037  
00038  
00039  
00040  
00041  
00042  
00043  
00044  
00045  
00046  
00047  
00048  
00049  
00050  
00051  
00052  
00053  
00054  
00055  
00056  
00057  
00058  

```
C SUGGESTIVE 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
C DIMENSION X(50),DI (50),DO (50),RI (50),XX (100),
1 AIR (100),ARO (100),ASI (50),ASO (50),V (50),
2 AIR (50),AM (50),V (50),RAN (50),SMA (50),TMA (50),
3 SIGMA (50),LTS (50),RMS (50),SUM (50),TD (50),
4 TF (50),AG (50),SLT (21),HSLT (21),sC (21),
5 TB (50),DL (50),AL (50),C (50),RI (50),
6 RHLI (100),ASLO (100),ASLI (50),ETA (50),EL (50),
7 AH (50),ARRO (100),AT (50),AED (50),EL (50),
8 V (50),DI (50),SFF (50),
9 WRITE (6,928)
928 FORMAT ('11.50X:: HEAT TRANSFER INPUT ::')
10 READ (5,1100) HED
C
C INITIALIZE SOME PARAMETERS, MOSTLY FOR OUTPUT PURPOSES.
C
11 HWD = C.
12 EPSL = C.
13 NPASS = 0.
14 WINJ = C.
15 TAU = C.
C
C NO. OF NODES: FLAGS FOR STEADY STATE OR TRANSIENT, FILM COOLING;
C REGEN, LINER, INJECTOR;
C
16 READ (5,101) NNODE,NTYPEF,NLMLF,NMLNF,NLFL,INFL
17 DO 1,5 = 1,ANODE
18 TIN (1) = 460.,
19 THL (1) = 460.,
20 TL (1) = 460.,
21 WSHT (1) = 0.,
22 WCCL (1) = 0.,
23 WPASS(1) = 0.,
24 WPASS(1) = 0.,
25 ETA(1) = 0.,
26 DL (1) = 0.,
27 DLO (1) = 0.,
28 READ (5,102) TSTOP,TPRINT
29 READ (5,102) TSTOP,TPRINT,FLOATING TO NCOUNT AND NPRINT (INTEGER)
30 C THIS CONVERTS TSTOP AND TPRINT (FLOATING) TO NCOUNT AND NPRINT (INTEGER)
31 C FOR STEADY STATE. INPUT MUST HAVE A DECIMAL POINT.
32 IF (NTYPEF.EQ.1) GO TO 20
33 NCOUNT = TSTOP
34 NPRINT = TPRINT
35 IF (NPRINT .EQ. 0) NPRINT = NCOUNT
36 20 NWP = TSTOP + 460.,
37 TSS = TSTOP + 460.,
38 TSS = TSS + 2,
39 TSS = TSS + TSS
40 READ (5,102) RC,RO,CP,TINJ,TINJ2
41 TINF = TINJ
```
TINJ = TINJ + 460,
WTEMP = TINJ
TINOF = TINJ + 460,
TINJO2 = TINJO2 + 460,
DO 202 I = 1,NNODE
200 READ (5,102) X(I),D(I),DO(I)
IF (YFLNFL .EQ. 0) GO TO 202
READ (5,101) NINJ
DO 201 I = 1,NINJ
201 CONTINUE
READ REGEN INPUT AND MAKE SURE REGEN STATIONS MATCH NODES
202 IF (YFLNFL .EQ. 0) GO TO 206
READ (5,109) XRND,NPASX,REGEN
DO 203 I = 1,NNODE
READ (5,102) XR,HPASS(I),NPASS(I),EETA(I)
IF (ABS(XR*X(I)) .GT. .0001) GO TO 204
IF (ABS(XR*REGEN) .LT. .0001) GO TO 205
203 CONTINUE
WRITE (4,150) I
CALL EXIT
205 NAGEN = I
ILESI = I - 1
READ LINER INPUT AND MAKE SURE LINER STATIONS MATCH NODES
206 IF (YFLNFL .EQ. 0) GO TO 216
READ (5,102) EMLDOT,EPSL,XMAX
DO 207 I = 1,NNODE
READ (5,102) XL,DL(I),D(I)
IF (ABS(X(I) - XL) .GT. .0001) GO TO 208
IF (ABS(XL - XMAX) .LT. .0001) GO TO 209
207 CONTINUE
WRITE (4,137) I
CALL EXIT
THIS GUARANTEES A FILM INJECTION STATION AT THE END OF THE LINER
209 YLNODE = 1
IF (YLNODE .NE. 1) GO TO 210
WRITE (4,108)
CALL EXIT
210 IF (YFLNFL .EQ. 1) GO TO 217
NINJ = 1
SLOT(I) = XMAX
HLO(I) = 0.5*(DI(YLNODE) - DLO(YLNODE))
WCOOL(I) = EMLDOT
WFLNL = 1
GO TO 216
217 GO TO 219,111, NINJ
IF (ABS(SLOT(I) - XMAX) .LT. .0001) GO TO 219
IF (SLOT(I) .GT. XMAX) GO TO 221
111 CONTINUE
218 CONTINUE
219 SLOT(I) = XMAX
WCOOL(I) = 0.5*(DI(YLNODE) - DLO(YLNODE))
GO TO 216
211 NINJ = NINJ + 1
LINE
DO 222 J = 1, N
K = KPLSL - J
SLT(K) = SLOT(K-1)
HSLT(K) = HSLT(K-1)
WCOL(K) = WCOL(K-1)
222 CONTINUE
SLT(1) = XLMAX
HSLT(1) = 1/D(NLNODE) - DLO(NLNODE)
WCOL(1) = EMLDOT
216 NL11 = NLNODE + 1
C MATCH FILL 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 (AUE(I) .LE. SLOT(J)) .LT. .0001) GO TO 17
16 CONTINUE
17 HSLT(J) = HSLT(J)
WCOL(J) = WCOL(J)
15 CONTINUE
C READ INJECTOR DATA IF ANY
C IF (INJL .EQ. 0) GO TO 18
READ (5,12) INJS, EMINJ, ARINJ, HGMINJ, ARINJ0, HGMINJ0
C HEAD (5,12) RESINJ, QFINJ, WTINJ, CPINJ
C WRITE THE INPUT
C 18 IF (NTYFPL .LE. 0) WRITE (6,1000) HED,NODE,NTYPE,NFLMFL,NFLN
C 18 IF (NTYFPL .GT. 0) WRITE (6,300) INFL,INFL,NCOUNT,PRINT
C 18 IF (NTYFPL .LE. 1) WRITE (6,1001) HED,NODE,NTYPE,NFLMFL,NFLN
C 18 IF (NTYFPL .LE. 1) WRITE (6,1002) WALL,FST,IND,APP,TYP,EPS,POOF,RC
C 18 IF (NTYFPL .LE. 1) WRITE (6,1012) INJS,EMINJ,ARINJ,HGMINJ,ARINJ0,
C 18 WRITE (6,1049) HPASS,HAND,EPSL
C 18 WRITE (6,1050) DO 40 I = 1, NNODE, 5
C 18 WRITE (6,1051) DO 40 I = 1, NNODE, 5
C 18 LINE = NODE(J)+1
C 18 DO 70 J = 1, LINE
C 18 WRITE (6,1077) K,J,X(J),Y(J),DI(J),DO(J),HSLT(J),WCOL(J),HPASS(J),
C 18 ETAF(J),DLI(J),DLO(J)
C 18 CONTINUE
C IF (NTYFPL .LE. 0 OR. PRINT .GT. 0) GO TO 41
C WRITE (6,110)
C 40 CONTINUE
C 40 GO TO 1C
C 41 CONTINUE
C C CONDUCT GEOMETRIC CALCULATIONS
C C CALCULATE DISTANCE BETWEEN NODES
DO 220 K = 1, N
220 EL(K) = X(K) - XI
ISGRT (EL(1-1) **2 + (R(1-1) - R(1)) **2)

C SET UP X,Y COORDINATES FOR LINER INNER AND OUTER SURFACES

C CHANGE DIAMETER TO RADIUS

C IF (NLFL.EQ.0) GO TO 274

NLLES1 = NNODE - 1
DI 71U = 1
DI 71L = 1
DI 71I = 1
DI 71 = 1
DI 71J = 1

720 RLOI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

720 RROI(J) = 0.5 * RLo(J)

C END LINER ELEMENT INNER SURFACE AREA

C END LINER ELEMENT OUTER SURFACE AREAS

C CONTINUE

NVALUES = NVALUES + 1

740 CONTINUE

DO 750 I = 2, NVALUES

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

750 ACUML(1) = ACUML(1) + 3.141593 * (R(1(1-1)) + R(1)) **2 + (R(1(1-1)) - R(1)) **2

C FIND THE THROAT

NLPLUS1 = NNODE + 1

DO 320 J = 1, NLPLUS1

320 J = NLPLUS1 - 1
CC0299 IF (DI(J) = DI(J-1)) 325, 320, 320
CC0300 320 CONTINUE
CC0301 325 DSTAR = CI(J)  
CC0302  ASTAR = .755398 * DSTAR ** 2  
CC0303  NTWR = J  
CC0304 IF (NLFL .EQ. 0) GO TO 276
CC0305 C INITIALIZE LINER TEMPERATURES
CC0306 C DO 760 J = 1, NLNODE  
CC0307 TL(J) = TW  
CC0308 760 TL(J) = TJU  
CC0309 C CONVERTOR HEAT TRANSFER CHARACTERISTICS
CC0310 C COUPLATION BETWEEN NODES + ADMITTANCE
CC0311 C DO 280 J = 1, ALES1  
CC0312 280 ADM(1) = CAPPA * AXAVE(1) / ELS(1)
CC0313 C CAPACITIES
CC0314 C CC0315 IF (NLFL .EQ. 0) GO TO 306
CC0316 C INJECTOR CALCULATION
CC0317 C 305 IF (NLFL .EQ. 0) GO TO 311  
CC0318 IF (NITR .EQ. 0) TI = TW  
CC0319 306 RADI = .1304E-15 * ALNJ + EMINJ * (TSCJ + TI) + (TSSQ + TI + (TSR  
CC0320 * TI))  
CC0321 SUM1 = RADI + ARINJ * HGINJ + ARINJ * HGINJ + 1. / RESINJ  
CC0322 CI = UTINJ / CAPINJ  
CC0323 DTAU1 = CI / SUM1  
CC0324 307 TV(J) = TI / RESINJ + ADM(1) / TW(1) / (ADM(1) + 1. / RESINJ)  
CC0325 IF (. NOTVFL .EQ. 0) GO TO 342
CC0326 C INJECTOR TEMPERATURE, IF STEADY STATE
CC0327 341 TI = (WACI + TSR + ARINJ * HGINJ + HTADPR + APINJ + HGINJ  
CC0328 + 2TINJU) = WAINJ + .755398 * D11(1)**2 + TV(1) / RESINJ / SUM1  
CC0329 TIF = TI - 400.  
CC0330 342 IF (NITR .EQ. 0) GO TO 306
CC0331 C WRITE THE OUTPUT FOR TRANSIENT AT INTERVALS DETERMINED BY TPRINT
CC0332 C 311 IF (NITPFL .EQ. 0) GO TO 308  
CC0333 IF (((TAU - TAU1) / LT * TPRINT) .GE. 0.0) GO TO 308
CC0334 WRITE (4,1010) HEAD, TAU, OF, TZERO, VDOT
CC0335 IF (NLFL .EQ. 1) WRITE (6,1013) TIF
CC0336 WRITE (6,1011)  
CC0337 351 DO 35 I = 1, NLNODE 6
CC0338 358 WRITE (6,1005)
LINE = MID (I+4, NNODE)
DO 36 J = 1, LINE
TF = T J (J) - 400.
36 WRITE (4,1006) J, X(J), TF, TFF(J)
TAU = T J - THAO( TAU, PRINT)
CONTINUE IF ( T J GT. TSTOP ) GO TO 10
GO TO 36

WRITE THE OLOPTC FOR STEADY STATE AT INTERVALS DETERMINED BY NPRINT.
BUT ALWAYS PRINT LAST 4 ITERATIONS

NPR = NCOUNT - NITR
IF ( NPR LE. 3 ) GO TO 307
IF ( NITR+ NPRINT ) NE. 0 ) GO TO 306
WRITE (4,1003) MED, NITR, OF, TIER, WDOT
IF ( NINFL .EQ. 1 ) WRITE (4,1013) TIP
WRITE (4,1004)
DO 37 J = 1, NNODE
WRITE (4,1005)
LINE = MID (J+4, NNODE)
DO 38 J = 1, LINE
TF = T J (J) - 400.
TLF = T J (J) - 400.
THLF = THL(J) - 400.
WRITE (4,1006) J, X(J), TFF(J), TBF, TLF, THLF
CONTINUE
IF ( NITR .LE. NPRINT ) GO TO 10
GO TO 36

RAJAIN ADMITTANCES

DO 39 J = 1, NLES2
RALJ(NH) = U/31*(TSCU + TW(J+1) + TSSG + TW(J+1)
1(TSH + TW(J+1)) + ASI(J)
IF ( NLEJ2 + LT, THRT ) GO TO 6360
ELL * XJNYORE / XTHRT
LOD = UJ / NRTAR
FLOD = 444.439 * LOD * ((-814.6) + LOD* (444.4397777 + 744,047 * LOD))
DO 40 J = 1, NRTAR
OLES2 = XOL ((X(J)+1) / XTHRT) / ELL

VIEW FACTOR FOR THE TAILPIPE ( 15 DEGREE CONICAL ANGLE)

VIEW = FLOD + XOL * (((-242)2) + LOD * 28170 + XOL * (-5139) +
1.4473 + XOL))
RADP = EPS1 + (TSCU + TW(J+1) + TSSG + TW(J+1) + (TSR +
TW(J+1)) + ASI(J)) + VIEW
RADDM(I) = RADAM(I) + RADTP
CONTINUE
CONTINUE
IF ( NITR .GT. 0 ) GO TO 326

EVALUATE SOME PARAMETERS FROM DATA WHICH IS STORED AS FUNCTIONS
CP OF THE COMBUSTION GAS
IF ( OF .GT. 8.) GO TO 390
C PG = 3.322 + OF*(-1.9447) + OF*1.6478 + OF*(+.09804) + OF*
1.0068(A))
GO TO 351
390 CPG = 3.322 + OF*(-11.5075) + OF*(1.3548 + OF*(+.0824)) + OF*
1.002196 + OF*(-2.3872E-5)))
391 CONTINUE
C MOLECULAR WT OF THE COMBUSTION GAS
C WTMOL = FMOLWT(OF)
C GAMMA
C GAM = SPHEAT(OF)
C COMBUSTION CHAMBER TEMPERATURE
C TZERO = TGRIPD(OF)
C CORRECT TZERO FOR H2 AND O2 INJECTION TEMPERATURES
C C1OF = ((-0.0506) + OF* (.0122) + OF*(-0.0382))
1 + OF* (-0.0614) + OF* (-0.00507) + OF* (2.28E-5)
2 + OF* (-5.372E-7) + OF* (4.413E-9) ))))
C C2OF = (-5.002 + OF*(-0.0427) + OF*(-0.0799))
1 + OF* (.0199) + OF* (9.937E-9)
2 + OF* (-2.168E-6) + OF* (2.398E-8) ))))
C TZERO = TZERO + C1OF* (( TINJ02 - 335) ) + C2OF* (TINJ) = 535)
C TZERO = TZERO -400.
C PRANDTL NUMBER
C PRAND = (4. + GAM )/ (9. + GAM - 5.1)
C VISCOSITY
C VISC = 46.6E-10 + SORT (WTMOL) + (TZERO) ** .6
C FLOW WEIGHT FLOW
C WDOT = FC + ASTAR* SORT ((( WTMOL + GAM )/ (48.0634 + TZERO))** .5)
1(2. + (GAM + .1) )** .6 + (GAM + 1. )/ (GAM + 1.)
C CALCULATE HC DIVIDED BY (ASTAR/4)*SIGMA
C
C MOVAS = (0.26* (VISC/ASTAR)**.2 + CPG* (WDOT/ASTAR)**.8 *
1(4ASTAR**.2) )/ PRAND**.6
C CALCULATE HC(1) OVER SIGMA
C
C DO 321 IF1 =1.KNODE
C 321 HC0VS1(1)* MOVAS* (ASTROA(1))**9
C
C DO 322 I = 1 + KNODE
C 322 TF1(1) = TZERO
C IS THERE FILM COOLING?
C 323 IF (NFRMFL .NE. 1) GO TO 490
326 DO 330 I=1,NLEGS
   SIGMA(I) = ( 2.* TF(I+1) ) / ( TF(I+1) + TW(I+1) ) ** 8
327      HG(I) = HGVS(I+1) * SIGMA(I) * AS(I)
328      SUM(1) = HG(I) + RADADM(I) + ADM(I) + ADM(I+1)
   C   C RGEN
C   C REGN
C   IF ( RGAFL .EQ. 1 ) GO TO 600
C   C LIVEN
C   IF ( NLFL .EQ. 1 ) GO TO 700
C   C STEADY STATE OR TRANSIENT ?
C   IF ( NTVPL ) 327,327,335
C   C WALL TEMP FOR STEADY STATE: MAY HAVE MULTI-SLOT FILM COOLING AND
C   C INJECTOR, BUT NO REGEN OR LINER
C   C WALL TEMP FOR TRANSIENT: MAY HAVE MULTI-SLOT FILM COOLING AND INJECTOR, BUT
C   C NO REGEN OR LINER.
C   C MAXIMUM DELTA TIME FOR STABILITY
C   TAU = 0.
C   DO 339 I = 1,NLEGS
C   DO 340 I = 1,NLEGS
C   DTAU = DT(I) / SUMADM(I)
C   DO 340 I = 1,NLEGS
C   DTAU = DT(I) / SUMADM(I)
C   IF ( J ) GO TO 341
C   DTAU = MIN(DTAU,TAU)
C   GO TO 335
C   C INJECTOR TEMPERATURE IF TRANSIENT
C   TI = ( DTAU / CI ) * ( PDI + TS + ARINJ + HG1NJ + XTEMP + ARINJ4 )
C   HG1NJ = TINJU + QFINJ + .75598 * D11**2 + TM11/RESINJ
C   2* ( SUM(DTAU/CI) ) * TIOLD
C   341 TAU = TAU + DTAU
C   C CALCULATE WALL TEMP AT NEXT TIME STEP
C   DO 345 I = 1,NNODE
C      DO 346 I = 1,NLEGS
C      TOLD(I) = TW(I)
C      DO 350 I = 1,NLEGS
C      TW(I) = ( HG(I) * TF(I) ) / ( RADADM(I) + TS + ADM(I) + TOLOD(I) )
C      1ADM(I) = TW(I+1) - SUMADM(I) + TOLOD(I+1) + ( DTAU / CAP(I) )
C      TW(I) = TW(I) + 2*TOLD(I+1)
C      345 TOLD(I) = TW(I)
C      346 1 = 1,NNODE
C      345 1 = 1,NNODE
C      350 1 = 1,NLEGS
C      350 1 = 1,NLEGS
C
IF ( JNE, 1 ) TF(1) = .5 = ( TFBAR1 + TFBAR2 )

500 CONTINUE
AFIL = AFIL + 1
GO TO 326

C REGEN CALCULATIONS, STEADY STATE ONLY, MAY HAVE MULTI-SLOT FILM

C COOLING AND INJECTOR, BUT NO LINER

600 CONTINUE
620 IF ( NITE .GT. 0) GO TO 636
ELSE = X(IREGEN) = X(1)

C INITIALIZE REGEN TEMPERATURES

623 DO 630 K=1, NAREGEN
630 T(K) = WTTPR
C FIND TEMPERATURES FOR REGEN SECTION

C 635 CONTINUE
DO 630 LAD = 1, ILEG1
APASS = HPASS * HPASS(LAD+1) = WPASS(LAD+1)
ADMIAD = ADM(LAD) = ( 1. - APASS/AVE(LAD))
630 CONTINUE

C EVALUATE CONDUCTIVITY, VISCOSITY, AND CP OF HYDROGEN

C 640 TBF = TBF(L ) = 460.
L CAPPB = 2.1296E-6 + TBF * ( 3.8194E-9 + TBF *
11(-1.848E-12 + TBF * ( -911E-13 + 1.5602E-19 - TBF )))

C H_VISC = 4.37E-7 + TBF * ( 6.374E-10 + TBF * ( -2.83E-13 +
1TFC = ( 1.5E-10 - 3.14E-20 + TBF )))

C HMCW = ( HCOIST + HCCAPPA + ( TML+1 ) / TML+1))**(-.55); /; C HMCW = HVISC

C ASURF = SURFOL * EL(L+1) * MCG(L ) * RADM(L ) * ADM(L ) * ADM(L+1) *
1HGCW + ETAEF(L+1) = ASURF

C FIND TWALL LP TO REGEN INJECTION POINT

C 645 DO 645 L = 1, ILEG1
640 CONTINUE
645 TB = TB(L ) = ( ( MGL ) = TF(L-1) + RADM(L ) = TSR + ADM(L ) +
TW(L-1) = ( HGL ) = TF(L-1) + RADM(L ) = TSR + ADM(L-1) + TW(L-2)

11 + SUMAD
645 CONTINUE
DO 645 L = 1, ILEG1
640 CONTINUE
645 TB = TB(L ) = ( ( MGL ) = TF(L-1) + RADM(L ) = TSR + ADM(L ) +
TW(L-1) = ( HGL ) = TF(L-1) + RADM(L-1) + TW(L-2)

11 + SUMAD
645 CONTINUE
DO 645 L = 1, ILEG1
640 CONTINUE
645 TB = TB(L ) = ( ( MGL ) = TF(L-1) + RADM(L ) = TSR + ADM(L ) +
TW(L-1) = ( HGL ) = TF(L-1) + RADM(L-1) + TW(L-2)

11 + SUMAD
C FIND HYDROGEN TEMPERATURE IN REGEN PASSAGE
C
C TG(JB) = (HGCW + ETAEF(JB) + ASURF + TW(JB)) * ADCLNT + TB(JB*1))
C
C 1(HVISC) = .8
C
C 645 CONTINUE
C
C C FIND WALL ROT OF REGEN INJECTION POINT
C
C DD ASU = Y(RGEN) + NLES2
C
C T(W+1) = (HGM + RAD(M) + TSA + ADMM) * RGEN
C
C 650 CONTINUE
C
C C LIVER CALCULATIONS, STEADY STATE ONLY, MAY HAVE MULTI-SLOT FILM
C
C C COOLING, WILL HAVE A FILM AT END OF LINER WHETHER INPUT OR NOT, MAY
C
C C HAVE INJECTOR, BUT NO REGEN
C
C C RADIANT ADMITTANCES BETWEEN LINER AND WALL
C
C 700 ELSE X(NLNODE) = X(I)
C
C DD 770 I = 2*NLNODE
C
C EPSBAR = 1 / (1 / EPSL) + ((DLO(I) / DI(I))**2) * (1 / EPSL) + 1
C
C 771 RADL = 3.394*15 + EPSBAR * (TW(I)**3 + TL(I) + (TW(I)**2) + (TL(I)**3) + ASLO(I-1)
C
C C HEAT TRANSFER COEFFICIENTS FOR THE LINER
C
C 772 SIGNAL = ((2 + TF(I)) / (TF(I) + TL(I)))**1.8
C
C 773 HGLA = ((STAR / DL(I))**1.8) * SIGNAL * HOVAS * ARL(I-1)
C
C C MCCARTHY-WOLF H2 CONVECTION
C
C 775 THF = TL(I) - 440.
C
C HCAPPA = 2.1256*6 + TB
C
C 778 C PM = 3.39* TF + 0.00104 * TF + ((-3.165*6) + 0.0256*9)
C
C 779 1 TF = (((911*15 - 1.5602*19) + TB))
C
C 780 1(HVISC) = 8.74*10 + TB
C
C 781 CPB = 3.39* TF + (0.00104* TF + ((-3.165*6) + 0.0256*9))
C
C 782 THF = (((1.6*12 - 2.39*16 + 0.00104))
C
C C HGCW = 0.58233* (DLO(I)) / (DI(I) + DLO(I))
C
C 785 1 = (0.00104 + DLO(I))**(-0.8) * ((1) / (DI(I) / THF)) * ELB
C
C 786 2 / (DI(I) - DLO(I)) **(-0.19)
C
C 787 SUML = RADL + HGLA + HGCW
C
C 788
1010 FORMAT ('1 --ROCKET HEAT TRANSFER MODEL-- ',18A4//6X,1) TIM
2E CORE OF CORE TEMP WT FLOW/32X DEG F LB/SEC/
3X/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
CTION 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,F12,9,F12,0,F10,3,F11,4,F1
50,3)
.1013 FORMAT ('0',6X,'INJECTOR TEMPERATURE #1',F8,1)
END
.1013 FORMAT ('0',6X,'INJECTOR TEMPERATURE #1',F8,1)
IF(FIX(KPSI).LE.TMAX) GO TO 176

000060

FIX(KPSI)=TMAX

000064

176 IF(DUM+FIX(KPSI))90,90,31

90 DUM=UIN

000068

CC0043

AVN=

000071

CC0044

GO TO 32

000075

31 AVN=DUM/FIX(KPSI)

000078

CC0060

XH=UIN/AVN

000081

CC0061

IF((IXA(KPSI)).LT.3) GO TO 35

000085

CC0062

AMH=UIN/HHLD(KPSI)-1.0

000088

CC0063

AVEP=AVN

000091

CC0070

AT=AVP/THLD(KPSI)-1.0

000094

CC0071

AT=AVN

000098

CC0073

IF(LR.EQ.1) GO TO 61

0000101

CC0074

IF(IXE=U+U).GT.1) GO TO 61

0000104

CC0075

60 IF(AT-PLT).LT.61,61,61

0000108

CC0076

62 NUT=1

0000111

CC0077

FIX(KPSI)=2.0-FIX(KPSI)

0000114

CC0078

GO TO 76

0000118

CC0079

61 FIX(KPSI)=.64,64,64

0000121

CC0080

64 FIX(KPSI)=.5-FIX(KPSI)

0000124

CC0081

GO TO 76

0000127

CC0082

C

0000130

C00084

55 N-0

0000133

C00085

TIME=0

0000137

C00086

IF(HY.EQ.0) GO TO 4

0000140

C00087

WRITE(10,13)CUT,KPSI

0000143

C00088

WRITE(6,103)HHLD,RING,T,H,N,FIX(KPSI),DT,AVN,TIME,DUM,TMAX

0000146

C00089

WRITE(6,100)ALPHA

0000150

C00090

WRITE(6,100)AW

0000153

C00091

3 FORMAT(1453)

0000156

C00092

100 FORMAT(E15.7)

0000160

C00093

103 FORMAT(E15.7)

0000163

C00094

C

0000166

C00095

4 IF(DT,GT.(TIME+TIME)) DT=TIME-TIME

0000169

C00096

DD 85 J=J+1

0000172

C00097

85 HXY(UJ)=AW(U)

0000175

C00098

C

0000178

C00099

40 CALL FC

0000181

C00100

C

0000184

C00101

IF(HY.GT.1)WRITE(6,100)J,C

0000187

C00102

C

0000190

C00103

CALL PRL

0000193

C00104

C

0000196

C00105

IF(YGTH.O)WRITE(6,100)C

0000199

C00106

DD 65 J=J+1

0000202

C00107

IF((C()+AW(U)).LT.0.01) GO TO 19

0000205

C00108

IF((C()+C(U)).LT.0.01) GO TO 19

0000208

C00109

IF((C()+AW(U)).LT.0.01)GO TO 19

0000211

C00110

65 CONTINUE

0000214

C00111

C

0000217

C00112

66 DUM=0

0000220

C00113

DD 73 J=J+1

0000223

C00114

73 AW(U)=AW(U)+C(U)

0000226

C00115

C00116

GUM=DUM*AW(U)

0000229

C00117

C

0000232

C00118

USE ENTALPY TO FIND TEMPERATURE

0000235
C 19 PYTHON/3:12;27;112)C:TEST
RETURN;

C 20 X = Y + P + Q + R
RETURN;

C 21 X = Y + P + Q + R
RETURN;

C 22 X = Y + P + Q + R
RETURN;

C 23 X = Y + P + Q + R
RETURN;

C 24 X = Y + P + Q + R
RETURN;

C 25 X = Y + P + Q + R
RETURN;

C 52 IF (X > 0.00) GO TO 92
RETURN;

C 91 MY = 1
RETURN;

C 92 IDA MPS12 = IDAI(KPS1) - 1
END
SUBROUTINE INJECT (HELEM,DCHAMPS,TH,EMR,MPS1,DELPS1)

EXTERNAL FUEL

COMMON/LECTWT,PO,FM

COMMON/ALPHA(3,200),ALPHAM(3,200),YSPEC(9,200),YSPEC(5,200)

1,(9,200),SIGMA(1),XLE(1),RUT(200),CPHAR(200),XNL(200),UT(200)

2,(1200),RHO(200),SIGMA(200),T(200),RH(200),SMALL(200),K(200)

COMMON/ALPHA(3,200),RHO(200),PSI(200),PSIS(9,200),PSFS(200)

DIMENSION R(60),RPD(60),PSI(60),PSIS(9,200)

COMMON/XCUM(60)

DATA PI/3.14159/9

DATA ALFOX,ALFUEL/0,'FUEL'

FM PERCENT OX OVERALL, INPUT

EMR DESIRED MIXING EFFICIENCY, INPUT

WOF OX FLOW IN FUEL RICH REGION

WFF OX FLOW IN FUEL RICH REGION

WDF FUEL FLOW IN OX RICH REGION

WDO OX FLOW IN OX RICH REGION

PMO PERCENT OX IN OX RICH REGION

PMF PERCENT OX IN FUEL RICH REGION

EM CALCULATED MIXING EFFICIENCY

WT TOTAL MASS FLOW, INPUT

TO INLET OX TEMP, INPUT

TM INLET FUEL TEMP, INPUT

DP INLET PRESSURE, INPUT

HELEM NUMBER ELEMENTS, INPUT

RCHAMD = CCHAMP/2.

KOUNT = G

DEL = FF/(1.0 + FM)

WOFP = 0.5 * FM + WT + DEL

WDPW = WFF - 2.0 * DEL

OF = FF/(1.0 - FM)

WFP = 0.5 * WT * (1.0 + DEL) + OF

WFPW = WFF * DEL / OF

DPW = WFPW

DPPW = WDPW

FPW = WDPW / DPP

FPPW = WFPW / DPP

EM = 1.0 - (DP * WT) / (FM * WFP) / (DPP * WT) / (FM * WPPW) / (1.0 - FM)

ADD = (EM - EMR) / EMR

IF (ADD <= 0.001) GO TO 2

DEL = DEL + 0.001 * FM * ADD

KOUNT = KOUNT + 1

IF (KOUNT GT 1000) GO TO 500

GO TO 1

CONTINUE

NOW CALCULATE EFFICIENCY + CHECK AGAINST DESIRED MIXTURE EFFICIENCY.

EM = 1.0 - (DP * WT) / (FM * WFP) / (DPP * WT) / (FM * WPPW) / (1.0 - FM)

ADD = (EM - EMR) / EMR

IF (ADD <= 0.001) GO TO 2

DEL = DEL + 0.001 * FM * ADD

KOUNT = KOUNT + 1

IF (KOUNT GT 1000) GO TO 500

GO TO 1

CONTINUE

CALCULATE FOR GAS MIXTURE
C  CALCULATE MOLECULAR WEIGHT
CFFMP=64.558/(2.016*FHP+(32.0*(1.0-FMP)))
CFFMP=64.558/(2.016*FHP+(32.0*(1.0-FMP)))
C
C  COMPUTE CROSS SECTIONAL AREA
AREA=PI*DCMAB*DCMAB/4.0
ARATIO=SQR(TP+FHP/(TP+FHP)) * DP/DP
APP=AREA/(1.0+ARATIO)
AP=AREA-APP
C
C  CALCULATE NUMBER ANNU
C
C  ADD*DCMAB/2*NAN
C  DO 17 1=1,NAN
C  17 R(1)=1
C  ADD*NAPP
C  ADD*U*APP/ADD
C  RF(I)=SQR(AP/TP(I)/R(I)/(AP+APP))
C  WR(2)=R(1)
C  DO 22 U=1,NAN
C  R(I)=R(I-1)+2
C  RI=CPI(2)+2
C  RDI=AC(R(I)-R(I-1))
C  KK=1
C  RF(K)=SQR(RI + RDI)
C  KK=K
C  RF(K)=SQR(R2 - RDI)
C  KK=K
C  20 KK=1
C  WR(1)=R(I)
C
C  CALCULATE VEL IN FUEL RICH REGION
C
C  RHOC=PC*FHP/TP/18540.
C  ACCEL=SQR(1.4*18540.0*12.0*32.2*TP/FHP)
C
C
C  CALCULATE VEL IN OX RICH REGION
C
C FIND RICH AT EACH FUEL RICH-OK RICH DISCONTINUITY
J=2
READ(1)RF(1)
LIMIT=((3*NN))=1
DO 21 J=3,LIMIT+3
RF(J)=RF(1)
20 FORMAT (6F2.5)
RF(J+1)=RF(1)
J=J+2
21 CONTINUE
LIMIT=2*NN
RF(LIMIT)=PCAMB
WCUM(1)=0.0
WCUM(2)=RHQVP*RF(1)**2.1
WCUM(3)=WCUM(2)+RHQVP*RF(2)**2.1
IF (NN,E0.1) GO TO 7735
DO 7745 J=3,LIMIT+2
WCUM(J)=WCUM(J-1)+RHQVP*RF(J)**2.1
7745 CONTINUE
7735 CONTINUE
C FIND YS I P R I S COORD AT EACH FUEL RICH-OK RICH DISCONTINUITY
PSIIN(1)=0.0
DO 22 J=1,LIMIT
PSIIN(J)=RF(J)**((WCUM(1)+1)/(PSI-1)**2.1)**2.5
22 CONTINUE
C FIND FLOW PROPERTIES AT PSI COORDINATES
DEPSI=(WT/P)**-0.51/(MPSI-1)
OLPSI=1.73492*DEPSI
PSIN(1)=0.0
T(1)=TP
U(1)=UP
YSPEC(4,1)=(1,-FMP)
YSPEC(5,1)=FMP
DO 7725 T=1,PSI
PSISI(1)=PSISI(1-1)+DEPSI
LIMIT=2*NN+1
DO 7724 J=1,LIMIT
IF (PSISI(1)-LE.PSIIN(J)) GO TO 7723
7724 CONTINUE
7723 CONTINUE
DO 7722 J=1,LIMIT+2
IF (J-LE.J) GO TO 7721
7722 CONTINUE
GO TO 772
C SET PROPERTIES FOR PSI(1) FOR OXIDIZER RICH ZONE
721 T(1)=TP
U(1)=UP
YSPEC(4,1)=(1,-FMP)
YSPEC(5,1)=FMP
GO TO 725
C SET PROPERTIES FOR PSI(1) FOR FUEL RICH ZONE
720 T(1)=TP
U(1)=UP
YSPEC(4,1)=(1,-FMP)
YSPEC(5,1)=FMP
725 CONTINUE
C ZERO SPECIES MASS FRACTIONS OTHER THAN H2 AND G2

DO 720 J=1,KPS
DO 727 J=1,3
727 VSPEC(J)=0.0
DD 728 #P.9
728 VSPEC(J)=0.0
728 CONTINUE
RETURN

ERROR CONDITIONS

WRITE(6,501) EM
501 FORMAT('1','5X,'COULD NOT CONVERGE ON MIXING EFFICIENCY IN SUBROUTI'
INE INJECT'/' 5X,'LAST EFFEC' '1,1PE15,6)
CALL EXIT

TOL*ALFCX
GO TO 530

TOL*ALFLEL

WRITE(6,531) TOL*FMACH
531 FORMAT('1','5X,'COULD NOT CONVERGE ON MACH NO IN CONTINUITY EQ FOR'
1,'44,' RICH AREA SUBROUTINE INJECT'/' 5X,'LAST MACH' '1,1PE15,6)
CALL EXIT
RETURN
.END
0101  SUBROUTINEacock
0103  COMMON/LAL1P(3,200),RAPH(3,200),YSPEC(9,200),RYSPEC(9,200)
0103  COMMON/091,092,1,001,002,SIGMA(1),XLE(1),RUL(200),CPR(200),MK(200),U(200)
0103  2,AT(2),W(2),Y(200),PSI(200),TI(200),RM(200),SALLC(200),M(200)
0103  COMMON/WTIX(9,2),RT(200),AUT(200),HUT(200),TELAM(200),EMOT(200)
0103  4,EEL(20))
0104  COMMON/EXTRA,AMP(17),NREAG,TMLD(200),HHL(200),FR(200)
0104  COMMON/ZC,TMLD(9),TTL(12),CP(7,41),X(7),XX(7)
0105  1,ASCALE,TX(7,4)
0105  COMMON/LDAPS,PSI,EFINS,ICHEM,ITURB,IPRES,IPUT,IPC,MY,INTYPE
0106  2,PSL,LW,LW,LX,LX,LX,NSPC,MA,KM,KM,MD,ME,ME,ME,MN
0106  4,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC,VEC
0106  5,DELPS1,DX,XPS,PRNT,PCNT,XX2,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0106  6,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0107  7,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0107  8,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0107  9,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0107  10,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0107  11,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0108  12,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0108  13,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0108  14,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0108  15,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0108  16,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  17,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  18,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  19,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  20,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  21,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  22,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  23,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0109  24,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT,OPT
0110  IF(ICH.MEG,2)GO TO 120
0111  DC 139 T41,PS1
00230 87+ 1 = 2
00231 88+ 2CU (P, U = MPS1
00234 87+ 4( (Y(1) * Y(1) A (1) * Y(1) * Y(1) * P8(1))
00236 90+ G0 TO 2)
00237 91+ 144 GN 142 (I = 1, MPS1
00242 92+ 142 G(1) * P(1) * RLT(1)
00243 93+ C
00244 94+ IF (MPS1; EG.1) GO TO 300
00245 95+ C
00246 96+ 20 CALL WALL
00247 97+ C
00248 98+ DETERMINE STEP SIZE
00249 99+ 300 CONTINUE
00250 100+ CALL STEP
00251 101+ C
00252 102+ DETERMINE WHETHER TO OUTPUT
00253 103+ 350 CONTINUE
00254 104+ 555 IF (NICUT.LE.1), OR ((FINISH. GE. 2) GO TO 600
00255 105+ KKK = KKK 1
00256 106+ PCT = PCT+DX
00257 107+ C
00258 108+ REMOVED WRITE STATEMENT LATER
00259 109+ WRITE 5, (E, O(1)) X, XMAX
00260 110+ C
00261 111+ 651 IF (PLOT.LE.PCNT) GO TO 401
00262 112+ 401 IF (KKK.LT.PCT), GO TO 600
00263 113+ 402 IF (KKK.IEQ.PCT), GO TO 600
00264 114+ 403 GO TO 7)
00265 115+ C
00266 116+ 600 CALL PRINT
00267 117+ KKK = 0.
00268 118+ PCT = 0.
00269 119+ IF (FINISH.EQ.2) RETURN
00270 120+ C
00271 121+ 700 IF (CHM.TS) GO TO 50
00272 122+ 50 710 IF (41 = MPS1
00273 123+ NIC 710 IF (1 . EQ. MSFC
00274 124+ 710 WHT(J+1), WHT(J), WHT(J-1)
00275 125+ C
00276 126+ IF (CHM.EQ.3) CALL PREPARE
00277 127+ 50 WHT(X+1)
00278 128+ IF (H1H.GT.1) GO TO 36
00279 129+ C
00280 130+ CALL PREPARE FROM INPUTTED POLYNOMIALS FOR P(X)
00281 131+ CALL PREPARE FROM INPUTTED POLYNOMIALS FOR P(X)
00282 132+ CALL AREA
00283 133+ HELM. + DL = 0
00284 134+ 38 CALL AREA
00285 135+ 38 CALL AREA
00286 136+ C
00287 137+ HELM. + DL = 0
00288 138+ HELM. + DL = 0
00289 139+ HELM. + DL = 0
00290 13A+ IF (CHM.EQ.4), CALL CONSER
00291 13B+ CALL CONSER
00292 13C+ CALL CONSER
00293 13D+ CALL CONSER
00294 13E+ CALL CONSER
00295 13F+ CALL CONSER
00296 13G+ CALL CONSER
00297 13H+ CALL CONSER
00298 13I+ CALL CONSER
00299 13J+ CALL CONSER
00300 13K+ CALL CONSER
00301 13L+ CALL CONSER
00302 13M+ CALL CONSER
00303 13N+ CALL CONSER
00304 13O+ CALL CONSER
00305 13P+ CALL CONSER
00306 13Q+ CALL CONSER
00307 13R+ CALL CONSER
00308 13S+ CALL CONSER
00309 13T+ CALL CONSER
00310 13U+ CALL CONSER
00311 13V+ CALL CONSER
00312 13W+ CALL CONSER
00313 13X+ CALL CONSER
00314 13Y+ CALL CONSER
00315 13Z+ CALL CONSER
00316 13A+ CALL CONSER
00317 13B+ CALL CONSER
00318 13C+ CALL CONSER
* ELT NOZ, 1, 20512, 51645 *

000001 SUBROUTINE NOZ(RT, RE, XN, DADX, AR, PERBEL)
000002 DIMENSION DADX(22), AR(22)
000003 C PROGRAM TO COMPLETE NOZZLE AREA RATIO GRADIENT AS FUNCTION OF AREA RATION
000004 C RT = THROAT RADIUS - INCHES
000005 C RE = EXIT RADIUS - INCHES
000006 C XN = LENGTH OF NOZZLE - INCHES
000007 C DADX = NOZLE GRADIENT
000008 C AR = AREA RATIOS CORRESPONDING TO NOZZLE GRADIENT
000009 C ADES = DESIGN AREA RATIO
000010 C PERBEL = PERCENT BELL IN PERCENT
000011 XRT = X/RT
000012 XATEE = RT
000013 ADES = 10 ** (33.22-1.034*PERBEL+.01132*PERBEL**2) *(4.872E-9)*PERBEL
000014 i
000015 AR(i) = 1.001
000016 DO 5 i = 1, 10
000017 AR(i+1) = AR(i) + 0.05
000018 5 CONTINUE
000019 DC 3 j = 11, 15
000020 J1 AR(j+1) = AR(j) + 1
000021 DO 3 j = 16, 21
000022 AR(j+1) = AR(j) + 0.5
000023 ADES = ADES * (RE/RT)**2
000024 DO 1 j = 1, 22
000025 IF (AR(j) .LT. 1.15) GO TO 20
000026 DADX(j) = -.9203* (1.0029*ADES3) + (2.1024*AR(j)) + (7.875E-6)*ADES3
000027 1 = 4.832E-9*AR(j)**2 + (4.832E-9)*ADES3 + j068*AR(j) + 3)* (6003376
000028 2 = ADES * AR(j) + (9.692E-8)*ADES3 * AR(j)**2
000029 DC TC 10
000030 20 DADX(j) = (AR(j) - 1)**6.2
000031 10 CONTINUE
000032 RETURN
000033 END
SUBROUTINE PADE
COMMON/FCC/THP*HC*ALPHI(8),XPSI(8),AN(9),B(8),RHOB,TIME,DT

DO 10 K=1,N
DO 25 J=1,N
S=S+A(K,J)*A(K,J)
R(R)*=R
25 CONTINUE
S=S+A(K,J)*A(K,J)
R(R)*=R
10 CONTINUE
CALL PICT
RETURN
END
COMMON/FCE,PER
C PER IS THE PERCENT/100 OF FUEL FLOW AT THE INJECTION POINTS
C T IS THE INJECTION TEMPERATURE OF THE COOLANT HYDROGEN
DIMENSION CAD(22),AR(22),ET(200),SIF(200),SIS(200),OF(200)
DIMENSION A(200),W(200),OMIN(200),AFRZ(6),ISPK(6),SIF(200),R(200)
DIMENSION RW(200)
REAL ISPK,W
C THE NEXT 7 STATEMENTS CONVERTS THE OUTPUT FROM THE COMBUSTION
C PROGRAM WHICH ARE IN THE SI UNITS TO THE OLD FASHIONED LB SEC IN UNITS

OC107 11  C
OC110 12  "OC 2" I=1,5
OC113 13  IF(I.EQ.1)EN=1.0D0
OC114 14  1.1=1.0D0
OC115 15  1.1=1.0D0
OC116 16  1.1=1.0D0
OC117 17  20 (/I/)=1.1=0.01422
OC121 18  1.1=1.0D0
OC122 19  1.1=1.0D0
OC123 20  1.1=1.0D0
OC124 21  1.1=1.0D0
OC125 22  1.1=1.0D0
OC126 23  1.1=1.0D0
OC127 24  1.1=1.0D0
OC128 25  1.1=1.0D0
OC129 26  1.1=1.0D0
OC130 27  1.1=1.0D0
OC131 28  1.1=1.0D0
OC132 29  1.1=1.0D0
OC133 30  1.1=1.0D0
OC134 31  1.1=1.0D0
OC135 32  1.1=1.0D0
OC136 33  1.1=1.0D0
OC137 34  1.1=1.0D0
OC138 35  1.1=1.0D0

C RT RADIUS THROAT INCHES
C RC RADIUS EXIT INCHES
C RN LENGTH FROM THROAT TO EXIT INCHES
C PEFF EXIT PERCENT BELL PERCENT
C PC TOTAL AVERAGE PRESSURE FROM COMBUSTION PROGRAM PSIA
C EY TOTAL ENTHALPY EACH STREAMLINE(CAL/GRAM)
C G5 SPACE RATIO EACH STREAMLINE (AS)
C GS CHARACTERISTIC VELOCITY FT/SEC
C RG RADIUS DIMENSION EACH STREAMLINE INCHES
C W W/4 EACH STREAMLINE LB/IN 50 SEC
C MW MOLECULAR WEIGHT AVE
C PG GAS TEMPERATURE TOTAL AVE DEG R
C CD TIME TIME FROM START OF VALVE ELECTRICAL SIGNAL
C CF THRUST COEFFICIENT BASED ON PC
C CHAMBER PRESSURE FROM DYNAMIC PROGRAM

C T=HE(RT)+T
C CALL MGZ(TE, IN, RADIUS, AR, PERCENT)
C CALL SIGMA1(TE, SIGMA, SIGMA, SIGMA, SIGMA)
C WRITE(6,20) (LY(I),I=1,200)
C WRITE(6,20) (SIF(I),I=1,200)
C WRITE(6,20) (SIG(I),I=1,200)
C WRITE(6,20) (OPS(I),I=1,200)
C WRITE(6,20) (FAS(I),I=1,200)
C CALL GradeCDX(RT, RADIUS, PT, OXIN)
C CALL CPHM(TMP, RADIUS, AR, FT, PT)
C CALL PRS1F(RADIUS, PT, G1, G2)
C CALL DIVERGE(RT, OXIN)
C (R(2)+R(1))/2,==2)*3.14159
C NAX=1
C 10 A(I)**(R(I)+R(I-1))/2,==2)=R(I)*R(I-1))/2,==2)*3.14159
C A(NAX)**(R(NAX)+R(NAX-1))/2,==2)=R(NAX)*R(NAX-1))/2,==2)*3.14159
C W=0
C K=0
C M=11 I=1,5
C T=12 I=2,5
C SPECIFIC IMPLSE AFTER KINETIC LOSS
C SPECIFIC IMPLSE LOSS DUE TO BOUNDARY LAYER
C CALL BLI(R, RT, PT, WW, MW, TG, DF)
SUBROUTINE PICT
COMMON/FCC/TP,HC,ALPHI(9),KPS,AAA(8,8),AK(9),R(8),RHOB,TIMEF,DTF
COMMON/FCY/AA(8,8),ABS(8,8),N,M,0
DOUBLE PRECISION DP
P=1.0
I=0
J=0
K=0
I=0
J=0
DO 40 I=1,N
EP=EPS
DO 40 I=1,N
EPS=EPS+ABS(A(I,I))
40 CONTINUE
DO 90 J=1,KM
DO 90 K=1,N
90 CONTINUE
C
C COMPUTE SIGMA, ALPHAS AS A CHECK SUM
000990 SUM(1) = 0
000990 DO 10 J = 1, NSPC
000990 XMUL(J,1)=SPEC(J,1)*WTMX1(J)/WTMOLE(J)
000990 10 SUM(1)=SUM(1)+XMUL(J,1)
000990 IF(MPSI.EQ.1) GO TO 25
C
C IF(NB.NE.1) GO TO 221
C IF(NH.NE.0) GO TO 222
C IF(NW.NE.0) GO TO 223
C Y=CONT(Y(LR)**2*(PSI**2-PSI(LR)**2)/RUT(LR))
C GO TO 221
C Y=CONT(LR)
C GO TO 222
C 221 CONTINUE
C IF(EUBL(1),LE.,0) GO TO 181
C FEHL17, J;365079*EUBL(LH)/EUBL(LO)
C GO TO 182
C 181 FEHL1; E30
C IF(EUBL(1),LE.,0) GO TO 183
C OFBL=EUBL(LO)/EUBL(LH)
C GO TO 184
C 183 OFBL; E30
C 184 CONTINUE
C SUMMF=2.C
C DO 200 J = 1, ASPC
C SUMWF=SLPHF*SML(J)*DEL(J)
C 200 WSBL=EUBL-SUMWF
C XMUL=PM=WMHL
C GIBL=XMUL/(XMBL-RG)
C SCHL=RT*(GIBL*RG/WMBL*TBL)
C EMBL=UHL/SBL
C 2010 DNL=EMHL=1
C DNL=EML=EMBL*ENB
C TBL=P**(R*(GIBL/DN))
C TBL=TBL*SM
C EMBL=EMBL
C ARE=YY
C IF(NH.NE.0) GO TO 20
C ARE=YY=YY=YY
C 20 IF(RP.*PST*PIM) GO TO 25
C 25 CONTINUE
C PS14=PS1.0.1325E5
C XD=X/WK(1)
C IPGE=IPGE+1
SUBROUTINE RTNI(X,F,DERF,FCT,XST,EPS,IERD,IER)
C
C PREPARE ITERATION
C IEN=0
C X=XST
C TOLL*
C CALL FCT(TOLL,F,DERF)
C TOLF1%EPS
C
C START ITERATION LOOP
C ON 1 IEN, IEND
C IEN=1
C IFDERF=12,8+2
C ITERATION IS POSSIBLE
C ON 2 DXF/FDERF
C TOLL*
C CALL FCT(TOLL,F,DERF)
C
C TEST ON SATISFACTORY ACCURACY
C TOLL*
C IF(FABS(X)-TOL15.5+6
C IF(FABS(F)-TOLF17.7+6
C CONTINUE
C END OF ITERATION LOOP
C
C NO CONVERGENCE AFTER IEND ITERATION STEPS, ERROR RETURN,
C IEN=1
C RETURN
C
C ERROR RETURN IN CASE OF ZERO DIVISOR
C IEN=2
C RETURN
C
C END
C PROGRAM TO CALCULATE THE THEORETICAL SHIFTING ISP + KINETIC ISP
C OF EACH STRAIN/LINE - LINEAR INTERPOLATION USED TO PREDICT ISP
C CALCULATE KINETIC RECOM PERCENT AT 0/F4,6,8
00007
IF(PT.GE.50D0) GO TO 110
00008
AA=69697
00009
PB=1.0
00010
GC TC 120
00011
110
AA=10103
00012
PB=900.0
00013
CONTINUE
00014
DC 2C=1.3
00015
20 SIKO1=FPTK1)*((ISP1K1-ISP1K1))&sILOG10(PF1)-ALOG10(PF1))
00016
DC IC=1.1,1.5
00017
IF(0S(K1).LE.2.) GO TO 30
00018
IF(0S(K1).LE.4.) GO TO 40
00019
IF(0S(K1).LE.6.) GO TO 50
00020
IF(0S(K1).LE.8.) GO TO 60
00021
CONTINUE
00022
30 SIKO1=1.0
00023
GC TC 100
00024
GO TC 10U
00025
40 SIKO1=1.1,1.5
00026
GO TC 10U
00027
50 SIKO1=SIK211+((OF)(1)-4.1)/2.1*(SIK212-1.4)
00028
GC TC 100
00029
60 SIKO1=SIK211+((OF)(1)-4.1)/2.1*(SIK212-1.5)
00030
GC TC 100
00031
70 SIKO1=SIK213
00032
GO CONTINUE
00033
10 CONTINUE
00034
DC IC=1.1,1.5
00035
IF(SIK11.LE.0.5) SIK11=0
00036
GO SIK11=ASIF11&n+SIF11&n+ASF111)
00037
RETURN
END
WRITE(6,2) ICUT, ISOBAT, KTWS(K), TWB, ETA(K), RYKC(K), RHC, MTC(K), SG(K)

2 FORMAT(4X,36H100 MANY WALL TEMPERATURE ITERATIONS/319.7E15.7)
GO TO 170

133 CONTINUE
T*MUS(E(K))
GO TO 140

130 CONTINUE
T=MUS(A5SLCR)
MUS=MUSE*1

250 CONTINUE
WHUT*WUTBL
WHU*WHL
WHA*WHCB
RCP*CPBL
MV*S*VG
DO 251 ,#1, ASPC
HALAM(J)=HALAM(J)

251 CONTINUE
IF(MUSE.EQ.0.0) GO TO 252
RETURN
END
FUNCTION TO COMPUTE GAS TEMP IN DEG RANKIN

FUNCTION TGR(P, X)

GIVEN
1. CHAMBER PRESSURE  P
2. OX MIXTURE RATIO  X

DATA A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13
1 435.775, 2113.65, -266.074, 10.44714, 24383.8, 691573, 669524,
2 1.017397E-6, -9270.038E-3, 235999E-1, 1.08078E-7
3 -1.04847E-5
DATA D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13
1 5.35751E-4, 1.232332, -13.65323, 1.6943759, 1.6373598E-2, 1.577078E-2,
2 8.43751E-4, -1.69671E-1, 9.210784E-3, 1.2173636E-3, 1.0391393E-2,
3 9.3973E-7

C

X2*X2
X3*X2
X4*X3
P2*P
P3*P2
P4*P3

IF(X .GT. 8.0)GO TO 1

C

TGR = A1 + A2*X + A3*X2 + A4*X3 + A5*X4 + A6*X5 + A7*X6 + A8*X7 + A9*X8 + A10*X9 + A11*X10 + A12*X11 + A13*X12
RETURN

1

TGR = B1 + B2*X + B3*X2 + B4*X3 + B5*X4 + B6*X5 + B7*X6 + B8*X7 + B9*X8 + B10*X9 + B11*X10 + B12*X11 + B13*X12
RETURN

END
DO 70 J=1,NSPC
CC0120  IF(J.EQ.NT)GO TO 71
CC0121  IF(J.GT.ACL)GO TO 72
CC0122  YSPEC(J,PSI)*YSPEC(J,JR)-SCW*YSPEC(J,LR)
CC0123  GO TO 7C
CC0124  YSPEC(NNT,PSI)*YSPEC(NNT,JR)-SCW*(ALC(NNT)-YSPEC(NNT,LR))
CC0125  GO TO 7C
CC0126  YSPEC(NCL,PSI)*YSPEC(NCL,JR)-SCW*(1.-YSPEC(NCL,LR))
CC0127  CONTINUE
CC0128  GO TO 250
CC0129  EVALUATE ELEMENT WALL B.C.
CC0130  CONTINUE
CC0131  IF(SCW.LT.E.1.)AND.(SCW.GE.0.)GO TO 81
CC0132  WRITE(6,3)ICUT,NUSE,ISBAT,SCW,RC,FNT,ELC,LNT,TW,TF,
CC0133  WRITE(6,5)
CC0134  9 FO*HAT(40X.*3.WCOOLANT MASS FLUX (R TOO BIG OR H NEGATIVE)
CC0135  IF(SCW.GT.1.)SCW=1,
CC0136  IF(SCW.LT.0.)SCW=0,
CC0137  CONTINUE
CC0138  ALPHA(LH,PSI)*ALPHA(LH,JR)*SCW*(1.-ALPHA(LH,LR))
CC0139  ALPHA(LC,PSI)*ALPHA(LC,JR)*SCW*(1.-ALPHA(LC,LR))
CC0140  ALPHA(LN,PSI)*ALPHA(LN,JR)*SCW*(1.-ALPHA(LN,LR))
CC0141  CONTINUE
CC0142  RH=1/RH
CC0143  RFX=1/RFX
CC0144  RC=FCT
CC0145  RO=REL
CC0146  RH=FAC
CC0147  END
CC0148  DO 25; J=1,NSPC
CC0149  RALAN(J)=ALAN(J)
CC0150  CONTINUE
CC0151  RAVCL*'=RALAN(NCL).
CC0152  CONTINUE
CC0153  IF(NUSE.EQ.0.)GO TO 292
CC0154  RETURN
CC0155  END

UCC
UNIVERSITY COMPUTING COMPANY
SUBROUTINE VASC(XMU, TT, ALP)
CC0011  COMMON/ TC, NPSI, MPSI, IFINIS, IChem, ITURB, IPRESS, ICTUT, IPAGE, MY, NTYPE,
CC0013  1LX, LST, LUC, LUV, LW, LXY, LZY, NSPC, MA, MB, MC, MD, ME, MF, MS, MH
CC0017  3: MINT, MSLF, NGAS, KOPT, NEL, LO, LW, NHTO, NHT, NHT, NHTV, LUV, MP, IE09AT
CC0019  4: NWE, MG, MK, LN, NNT, JR, NSLCR
CC0021  DIMENSION ALP(9), AL(9)
CC0023  FORMAT(4,15.7)
CC0025  T=TT
CC0027  DO 20 J=1, NSPC
CC0029  20 AL(J)=ALP(J)
CC0031  CVV=47.87
CC0033  DO TC(3C, 40, 50, 60), LW
CC0035  30 CONTINUE
CC0037  XWUT=3.346E-8*CNV*T**1.5/(T**10.4)
CC0039  DO TC 98
CC0041  40 CONTINUE
CC0043  DO TO 98
CC0045  50 CONTINUE
CC0047  DO TC 98
CC0049  60 CONTINUE
CC0051  98 XMUXMHT
CC0053  RETURN
CC0055  END
SUBROUTINE VISC

COMCCH(2A/ALPHA(3,200),RbPHMA(3,200),VSPCEC(0,200),SRSPEC(0,200))

1. N(9,2,1),SIGMA(1),XLYE(1),H(200),CPB&K(200),XM(200),UL(200)

2. A(200),RUH(200),Y(200),PSI(200),T(200),R(200),SMALLW(200),W(200)

3. WTS(1X2D0),RTU(2D0),RUR(2D0),TELAP(2D0),EMD(2D0)

4. E(2D0)

5. CONC(1C/WTMOLE(9),TITLE(12),CNP(7,4),X(7,7))

6. GCALL(1C=K(7,4))

7. CONC(1C=100/VST(LH,1F4.1),CHEM(1),N99H(1),PRES(1),OUTP(1),PAGE(1),MTYPE(1),MP

8. L(3),L(VL,LV,LZ,XSPL,HSPL,H,M;L,M;M;M;M;M;M)

9. TUB(1),XUHY,IGAS,KOPT,REL,LO,LH,HNOT,N1T,N99T,N99T,LUV,MP,ISOBAT

10. N(9,10),PLN,N1T,JX,N1TB

11. CONC(1C=P(1X4),XUHY,DELFSI,DX,XMPS,PRNT,PCNT,KX2,DPD,XTRA,HST

12. C,OSY,PMAM,PMAM,A,AKA

13. CONC(1C=1F6,W(100),W(100),K(100),W(100),WE(100),TT(100),VL(100),CPD,W(100),HBB)

14. MRBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL,MBL,STBL

15. FORMAT(FL15.7)

16. FROM SL/GFT/SEC TO KG/M-SEC

17. CVX=47.27

18. G0 TO 2C,75,76,77,78,79,1TURB

19. CONTINUE

20. CONTINUE

21. TURB=2 = WIRSCH ROCKET CHAMBER MODEL

22. ALPHA(3,200) = 200

23. XUHY(1.X,2,1) = 14.0

24. G0 TO 9A

25. CONTINUE

26. TURB=2 = VISCOSITY IS AN INPUTTED CONSTANT

27. CONTINUE

28. G0 TO 9A

29. CONTINUE

30. G0 TO 9A

31. CONTINUE

32. G0 TO 9A

33. CONTINUE

34. G0 TO 9A

35. CONTINUE

36. CONTINUE

37. CONTINUE

38. CONTINUE

39. CONTINUE

40. CONTINUE

41. CONTINUE

42. CONTINUE

43. CONTINUE

44. CONTINUE

45. CONTINUE

46. CONTINUE

47. CONTINUE

48. CONTINUE

49. CONTINUE

50. CONTINUE

51. CONTINUE

52. CONTINUE

53. CONTINUE

54. CONTINUE
CALL HEAT(LR, MPSI)
IF(T(1), LE., 0) GO TO 20
GO TO 29
300 CONTINUE
IF(X. LT., XS(1)) GO TO 301
CALL SLET
301 CONTINUE
W(MPSI) = W(JR)
W(MPSI) = W(JR)
GAM = 2
CALL HEAT(MPSI, MPSI)
IF(T(MPSI), LE., 0) GO TO 20
GO TO 29
20 IF(N(X) .LT. 2)
WRITE(6, 4) SIMULATION DUE TO WALL El. C; = STOP
4 FORMAT(/'45x')
RETURN
25 CALL DEASE(MPSI, MPSI)
RETURN
33 CALL MPSI
RETURN
END
END CLA