INTEGRATED MODEL DEVELOPMENT FOR LIQUID FUELED ROCKET PROPULSION SYSTEMS

Final report for work performed under UAH subcontract associated with NASA contract NAG8-212, Task No. 6.

June, 1993
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Background</td>
</tr>
<tr>
<td>2.0</td>
<td>Analysis Procedure</td>
</tr>
<tr>
<td>3.0</td>
<td>Preliminary Results</td>
</tr>
<tr>
<td>4.0</td>
<td>Recommendations</td>
</tr>
<tr>
<td>5.0</td>
<td>References</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
</tr>
<tr>
<td>A. Tables</td>
<td></td>
</tr>
<tr>
<td>A1. Summary of uncertainties for high pressure fuel turbopump subsystem analyses</td>
<td></td>
</tr>
<tr>
<td>A2. Summary of independent variable predictions for high pressure fuel turbopump subsystem analyses</td>
<td></td>
</tr>
<tr>
<td>A3. Summary of imbalance predictions for high pressure fuel turbopump subsystem analyses</td>
<td></td>
</tr>
<tr>
<td>B. Figures</td>
<td></td>
</tr>
<tr>
<td>B1. Model Hierarchy Diagram</td>
<td></td>
</tr>
<tr>
<td>B3. Branch Pressure Loss</td>
<td></td>
</tr>
<tr>
<td>B4. Branch Energy Balance</td>
<td></td>
</tr>
<tr>
<td>B5. Turbopump Balances</td>
<td></td>
</tr>
<tr>
<td>B6. Nozzle and Overall Balances</td>
<td></td>
</tr>
<tr>
<td>B7. Test Configuration</td>
<td></td>
</tr>
<tr>
<td>C. Documentation</td>
<td></td>
</tr>
<tr>
<td>C1. Description of Routines</td>
<td></td>
</tr>
<tr>
<td>C2. Input/Output Variable Definitions</td>
<td></td>
</tr>
<tr>
<td>C3. Source Code Listing</td>
<td></td>
</tr>
</tbody>
</table>
1.0 BACKGROUND

As detailed in the original Statement of Work, the objective of Phase II of this research effort was to develop a general framework for rocket engine performance prediction that integrates physical principles, a rigorous mathematical formalism, component level test data, system level test data, and theory-observation reconciliation. Specific Phase II development tasks are defined as follows:

Task 1. Identify device modules required for rocket engine analysis. Identify device specific forms of fundamental physical principles. Identify device specific forms of generally accepted engineering approximations.

Task 2. Construct logic for complete thermo-physical analysis within each physical device module.

Task 3. Define physical device module interface with hardware performance characterization.

Task 4. Construct specific computational logic sequencing routine for thermo-physical analysis of engine system.

Task 5. Construct reconciler interface with device modules. Construct reconciler interface with gains model.

Task 6. Write final report documenting integrated software platform development.

SSME steady-state performance is defined by the fluid, flow and hardware characteristics that exist throughout the engine during steady-state operation. A logical platform for effective engine performance prediction must integrate physical principles and empirical information within a rigorous mathematical formalism. Physical principles pertinent to fluid flow and engine performance prediction are listed below.
Physical Principles

1. Conservation of Mass
2. Conservation of Energy
3. The Second Law of Thermodynamics
4. Newton's Second Law
5. Constitutive relations for thermal transport

Hardware characteristics of particular interest in liquid-fueled rocket engine performance prediction include the following:

Hardware Characteristics

1. Turbine performance relations
2. Pump performance relations
3. Duct and other device flow resistance relations
4. Chamber combustion efficiencies
5. Nozzle efficiency.

The objective of an efficient engine performance model is to identify physically consistent values of a complete, yet minimal, set of independent variables that describe the engine operating state. The variables selected must be consistent with the level of model approximation, and provide an adequate basis for insuring compatible hardware operating characteristics (see, e.g. [1] or [2]).

For a one-dimensional steady-state flow model, the engine network can be defined by specifying the following information:

Engine Network Components

1. Flow branches, each consisting of a single inlet, single outlet flow path with associated flow rate and resistance
2. Nodes, each defined as a specific location with an associated temperature and pressure

3. Devices, each associated with a specific hardware component and defined by intersecting branches and boundary nodes

4. Connectivity information defining each branch-branch and branch-device intersection and assigning individual nodes to each branch-branch and branch-device intersection.

Notably absent from network definition components is detailed geometric information. Therefore, Newton's second law cannot be applied in a typical performance analysis to obtain fluid-structure interaction at the component level.

As part of this investigation, a new one-dimensional steady-state rocket engine performance model has been constructed. This new model incorporates the physical principles described above, with the exception of Newton's second law, in a FORTRAN based computer software package. A simple interface for manufacturer supplied routines describing component hardware performance characteristics is provided. A description of the procedures employed in the new model is presented in the next section of this report.
2.0 ANALYSIS PROCEDURE

The purpose of the new theoretical model is to provide an efficient computational procedure for defining engine flow networks and for determining physically consistent performance characteristics within the engine network. For SSME performance analysis, a minimal set of solution variables is prescribed below. Other operating characteristics can be derived from these primary variables.

Primary SSME Performance Variables

1. Mass flow rates through system branches \([m \text{ (lbm/s)}]\)
2. Pressures at system nodes \([P \text{ (psia)}]\)
3. Temperatures at system nodes \([T \text{ (R)}]\)
4. Controllable valve resistances \([R \text{ (s}^2/\text{in}^2\text{-ft}^3)]\)
5. Turbopump shaft speeds \([N \text{ (rpm)}]\)
6. Heat transfer rates \([Q \text{ (Btu/s)}]\)

A number of computational strategies can be employed for one-dimensional steady-state network flow analysis. Adequate boundary and control setting specification is required to achieve solution closure. For SSME analysis, traditional inputs include flow inlet conditions (fluid composition, pressure, and temperature), the desired oxygen-hydrogen mixture ratio, and the required thrust level or corresponding nozzle stagnation pressure. The following is a list of independent relations used to solve for the set of primary performance variables in the new engine model:
Model Analysis Relations

1. Mass conservation at branch flow junctions
2. Energy conservation in specified system volumes
3. Pressure drop in flow branches as a function of branch resistance
4. Pressure drop across turbines as a function of turbine performance parameters
5. Temperature drop across turbines as a function of turbine performance parameters
6. Head gain across pumps as a function of pump performance parameters
7. Power required by pumps as a function of pump performance parameters
8. System mixture ratio as a function of main combustion chamber (MCC) inlet flows
9. Nozzle mass flow as a function of nozzle stagnation properties and geometry
10. Heat transfer rate as a function of driving temperature difference and thermal resistance.

These relations are presented below in residual form. An exact solution is a set of primary variable values that reduces each equation residual (or balance error) to zero:

**Governing Equations in Residual Form**

\[ \sum_{i} m_{ij} = (\text{mass flow residual})_j \]  \hspace{1cm} (1)

\[ i=1,2,\ldots \text{number of I/O's in mass flow circuit } j \]
\[ j=\text{mass flow circuit number} \]

\[ \sum_{i} m_{ij} h_{ij} - \sum_{k} m_{kj} h_{kj} + Q_j = (\text{energy flow residual})_j \]  \hspace{1cm} (2)

\[ i=1,2,\ldots \text{number of inputs to energy circuit } j \]
\[ k=1,2,\ldots \text{number of outlets from energy circuit } j \]
\[ j=\text{energy flow circuit number} \]
Governing Equations in Residual Form (continued)

\((P_\text{in} - P_\text{out} - R \cdot W^2 / \rho)_j = \text{(pressure drop residual)}_j \)  \hspace{1cm} (3)

\(j=1,2, \ldots\) number of pressure circuits

\[ \{ \rho \cdot N^2 \cdot D^2 \cdot [C_H - C_H(C_Q)] \}_j = \text{(pump } \Delta P \text{ residual)}_j \]  \hspace{1cm} (4)

\(j=1,2, \ldots\) number of pumps

\[ \left[ \frac{m* (P_\text{in} - P_\text{out}) / \rho}{\eta} - \frac{m* (P_\text{in} - P_\text{out}) / \rho}{\eta(C_Q, Ma)} \right]_j = \text{(pump power residual)}_j \]  \hspace{1cm} (5)

\(j=1,2, \ldots\) number of pumps

\[ [(P_\text{in} - P_\text{out}) - (P_\text{in} - P_\text{out}) \text{ characteristic}]_j = \text{(turbine } \Delta P \text{ residual)}_j \]  \hspace{1cm} (6)

\(j=1,2, \ldots\) number of turbines

\[ \frac{(P_\text{in} - P_\text{out}) \text{ characteristic}}{P_\text{in} * f(C_Q, Ma, \gamma)} = \text{ (pump power characteristic))} \]

\[ [ (T_\text{in} - T_\text{out}) - (T_\text{in} - T_\text{out}) \text{ characteristic}]_j = \text{(turbine } \Delta T \text{ residual)}_j \]  \hspace{1cm} (7)

\(j=1,2, \ldots\) number of turbines

\[ \frac{(P_\text{in} - P_\text{out}) \text{ characteristic}}{P_\text{in} * f(C_Q, Ma, \gamma)} = \text{ (pump power characteristic)} \]

\[ \frac{m_{\text{oxygen}}}{m_{\text{fuel}}} - \left( \frac{m_{\text{oxygen}}}{m_{\text{fuel}}} \right) \text{ command} = \text{(mixture ratio residual)} \]  \hspace{1cm} (8)

\[ \sum_i (m_i) - m_{\text{nozzle}} = \text{(nozzle flow residual)} \]  \hspace{1cm} (9)

\(i=1,2, \ldots\) number of nozzle inlet flows

\[ m_{\text{nozzle}} = f(P_n, T_{ch}, \text{nozzle geometry}) \]

\[ (Q - \Delta T_{\text{driving}} / \Delta T_{\text{thermal}})_j = \text{(heat transfer residual)}_j \]  \hspace{1cm} (10)

\(j=1,2, \ldots\) number of unknown heat transfer rates

where

\(m=\text{mass flow rate}\)

\(h=\text{specific enthalpy}\)

\(P=\text{total pressure}\)

\(Q=\text{heat transfer rate}\)

\(R=\text{flow resistance}\)
These highly nonlinear relations are depicted graphically in Appendix B, Figures B2 through B6b.

Many methods for solving systems of nonlinear equations are available (see, e.g. [3]). The objective of any solution procedure is to reduce the sum of all the residuals described above to zero. In practice this is generally not possible because of the approximate nature of both the physical relations and hardware performance curves. An appreciation of this limitation suggests the use of a minimization method to systematically and continuously reduce the residuals sum to an acceptable value.

It is well known that the problem of solving a system of nonlinear equations may be replaced by a problem of minimizing a nonlinear function on \( \mathbb{R}^n \) [3], where \( n \) is the number of independent variables. In addition, the quasi-Newton, or variable metric (VM), methods are particularly robust strategies for minimizing unconstrained nonlinear functions. These facts motivated the use of a VM solution method in the new theoretical model. Specifically, a BFGS [4] multivariate search algorithm was implemented with an Armijo [5] univariate sub-algorithm. In
the present application, the objective of the BFGS-Armijo solution strategy was to select branch mass flow rates, nodal pressures and temperatures, variable valve resistances and turbopump shaft speeds in order to minimize the sum of the squares of the residuals defined above. To provide a consistent scale for the various residuals, each was divided by a user specified uncertainty estimate prior to squaring and summation. The engine operating point was thus obtained as the solution to the following mathematical programming problem:

$$\text{Minimize } F = \sum_i \left( \frac{\text{residual } i}{\text{uncertainty of residual } i} \right)^2$$  \hspace{1cm} (11)

by selection of the primary performance variables.

A stepwise description of the model analysis procedure is given below:

**Analysis Procedure**

1. Define network branches, nodes, and connectivity.

2. Define mass, energy, pressure, turbopump, and nozzle circuits.

3. Enter pertinent fluid property data.

4. Specify uncertainties including mass, energy, and pressure circuit balance uncertainties, turbine pressure and temperature drop uncertainties, pump pressure rise and power variance uncertainties, mixture ratio uncertainty, thrust uncertainty.

5. Select desired mixture ratio and nozzle thrust.
Analysis Procedure (continued)


7. Append nozzle performance curves.

8. Initialize branch flow rates, nodal pressures and temperatures, branch resistances, and turbopump speeds from available data.

9. Solve for branch flow rates, nodal pressures and temperatures, valve controllable resistances, and turbopump speeds that provide engine balance.

A hierarchy diagram displaying routines used in the new performance prediction model is presented in Figure B1. A functional description of these routines together with a computer code listing of the new performance model is presented in Appendix C. Results based on preliminary testing of the new model are presented in the next section of the report.
3.0 PRELIMINARY RESULTS

In order to test execution of the new model, a series of performance analyses was conducted on the SSME high pressure fuel turbopump subsystem depicted in Figure B7. In this figure, nodes 1 and 2 correspond to oxygen and hydrogen preburner inputs respectively. Nodes 3 and 4 are hot gas flow locations, and nodes 5 and 6 correspond to liquid hydrogen flow locations. Pump and turbine performance curves were approximated from predictions returned by Rocketdyne's SSME power balance model (PBM) over a typical range of engine power levels. All analyses were initiated at PBM solution values corresponding to 109% of SSME rated power level. Inlet temperatures and pressures at nodes 1, 2, and 5 were fixed. In order to provide the basis for theoretical closure, command values of the preburner mixture ratio and pump power input were also designated at PBM solution values.

Numerous test case analyses were conducted using somewhat arbitrary values of the residual scaling uncertainties. Results of three such analyses are presented in Appendix A. Table A1 presents the scaling uncertainties used in each analysis. Specific values of these uncertainties were originally selected (Case A) to obtain common order of magnitude scaling of the various residual terms in Equation 11. In effect this made satisfaction of each residual relation approximately equal in importance.
In subsequent analyses, uncertainty scaling was used to emphasize or de-emphasize various relations. Case B uncertainty values were chosen identical to Case A values except for the preburner mass balance uncertainty which was an order of magnitude smaller in Case B. This indicates an increased emphasis on obtaining mass flow balance in Case B. Energy related uncertainties in Case C were reduced while uncertainties in the other turbine and pump performance relations were increased. This indicates an increased emphasis on obtaining overall power balance, and decreased confidence in hardware performance relations.

Results of the three case analyses are presented in Tables A2 and A3. Approximate solution values of the independent variables for each case are presented in Table A2. Residuals associated with each of the governing balance relations are presented in Table A3. In no case were all residual values reduced to zero identically. Therefore, an exact solution was not obtained in any of the three analyses. This was disconcerting although not surprising based on the approximate nature of the hardware performance curves used in the test cases, and the fact that the current PBM does not achieve exact solutions even with the best available performance information.

Significant reductions in residuals were achieved at the approximate solutions returned by the new model. As shown in Table A3, the Case A solution provided significant reductions in power and pressure residuals accompanied by a small rise in
preburner flow residual. In order to reduce the flow residual (i.e., improve preburner mass flow balance), the flow balance uncertainty was reduced by an order of magnitude in Case B. A substantially different solution was obtained, with a decrease in flow residual from Case A and overall improvements in all power balances, although not as dramatic as provided by the Case A solution values. Similar comments could be made regarding the Case C solution residuals.

The results of Case B were considered more realistic because of the low preburner flow imbalance. The Case B solution prescribes lower preburner and turbine flows, little change in initial pressure estimates, reduced turbine discharge temperature, and increased pump discharge temperature reflecting a decrease in pump efficiency.
4.0 RECOMMENDATIONS

A list of recommendations based on construction, implementation, and run experience with the new performance model is presented below:

1. Develop a strategy for assigning specific residual scaling uncertainty estimates for future model testing.

2. Interface existing turbopump performance curves to new model.

3. Expand current property routine to include potential engine states.

4. Streamline and structure property input interface to the new model.

5. Implement postprocessing capability to recover additional hardware performance and design characteristics.

6. Construct definition and connection input file for the full SSME engine network.

7. Perform an extensive computational test program on the new model applied to the full SSME engine system network in order to determine model efficiency and performance accuracy.

8. Compare new model computational results with existing power balance model predictions and Technology Test Bed (TTB) experimental data in order to assess integrity of existing PBM.
5.0 REFERENCES


APPENDIX A

TABLES
Table A1. Summary of uncertainties for high pressure fuel turbopump subsystem analyses

<table>
<thead>
<tr>
<th>UNCERTAINTY ESTIMATE</th>
<th>CASE A</th>
<th>CASE B</th>
<th>CASE C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preburner power balance (Btu/s)</td>
<td>100</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Preburner flow (lbm/s)</td>
<td>0.1</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>O₂ resistance pressure drop (psi)</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>H₂ resistance pressure drop (psi)</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Pump flow head gain (psi)</td>
<td>10</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Pump power requirement (hp)</td>
<td>100</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Turbine flow head drop (psi)</td>
<td>10</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Turbine flow temp drop (deg R)</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Turbopump power (Btu/s)</td>
<td>100</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Preburner O₂/H₂ ratio</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
</tbody>
</table>
Table A2. Summary of independent variable predictions for high pressure fuel turbopump subsystem analyses

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>INITIAL</th>
<th>CASE A FINAL</th>
<th>CASE B FINAL</th>
<th>CASE C FINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 (lbm/s)</td>
<td>79.63</td>
<td>67.99</td>
<td>74.68</td>
<td>68.81</td>
</tr>
<tr>
<td>M2</td>
<td>84.08</td>
<td>72.27</td>
<td>78.12</td>
<td>72.69</td>
</tr>
<tr>
<td>M3</td>
<td>163.72</td>
<td>140.51</td>
<td>152.80</td>
<td>141.51</td>
</tr>
<tr>
<td>M4</td>
<td>162.25</td>
<td>148.53</td>
<td>152.16</td>
<td>162.70</td>
</tr>
<tr>
<td>P1 (psia)</td>
<td>7733.1</td>
<td>(same)</td>
<td>(same)</td>
<td>(same)</td>
</tr>
<tr>
<td>P2</td>
<td>6088.0</td>
<td>(same)</td>
<td>(same)</td>
<td>(same)</td>
</tr>
<tr>
<td>P3</td>
<td>5516.6</td>
<td>5790.5</td>
<td>5515.2</td>
<td>5847.9</td>
</tr>
<tr>
<td>P4</td>
<td>3717.7</td>
<td>3931.5</td>
<td>3717.7</td>
<td>3873.4</td>
</tr>
<tr>
<td>P5</td>
<td>271.7</td>
<td>(same)</td>
<td>(same)</td>
<td>(same)</td>
</tr>
<tr>
<td>P6</td>
<td>6738.7</td>
<td>6405.5</td>
<td>6725.6</td>
<td>6096.9</td>
</tr>
<tr>
<td>T1 (deg R)</td>
<td>208.3</td>
<td>(same)</td>
<td>(same)</td>
<td>(same)</td>
</tr>
<tr>
<td>T2</td>
<td>278.4</td>
<td>(same)</td>
<td>(same)</td>
<td>(same)</td>
</tr>
<tr>
<td>T3</td>
<td>1929.3</td>
<td>1897.8</td>
<td>1927.8</td>
<td>1903.6</td>
</tr>
<tr>
<td>T4</td>
<td>1777.6</td>
<td>1726.8</td>
<td>1765.1</td>
<td>1728.4</td>
</tr>
<tr>
<td>T5</td>
<td>42.4</td>
<td>(same)</td>
<td>(same)</td>
<td>(same)</td>
</tr>
<tr>
<td>T6</td>
<td>96.7</td>
<td>110.0</td>
<td>105.0</td>
<td>104.6</td>
</tr>
<tr>
<td>N (rpm)</td>
<td>36116</td>
<td>34991</td>
<td>36120</td>
<td>36993</td>
</tr>
</tbody>
</table>
Table A3. Summary of imbalance predictions for high pressure fuel turbopump subsystem analyses

<table>
<thead>
<tr>
<th>IMBALANCE</th>
<th>INITIAL</th>
<th>CASE A FINAL</th>
<th>CASE B FINAL</th>
<th>CASE C FINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prebnr Flow</td>
<td>0.0004</td>
<td>-0.2444</td>
<td>0.002</td>
<td>-0.0119</td>
</tr>
<tr>
<td>(lbm/s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prebnr Power</td>
<td>-6493.7</td>
<td>113.1</td>
<td>-1265.2</td>
<td>1009.1</td>
</tr>
<tr>
<td>(Btu/s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O₂ Resis ΔP</td>
<td>-455.5</td>
<td>-5.3</td>
<td>-131.9</td>
<td>-110.0</td>
</tr>
<tr>
<td>(psia)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H₂ Resis ΔP</td>
<td>179.7</td>
<td>8.1</td>
<td>234.7</td>
<td>-52.6</td>
</tr>
<tr>
<td>(psia)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPFP ΔP</td>
<td>-34.8</td>
<td>-14.7</td>
<td>-104.0</td>
<td>-1015.6</td>
</tr>
<tr>
<td>(psia)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPFP Power</td>
<td>-10997.2</td>
<td>-218.3</td>
<td>-5209.2</td>
<td>-1662.6</td>
</tr>
<tr>
<td>(hp)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPFT ΔP</td>
<td>12.7</td>
<td>11.0</td>
<td>8.9</td>
<td>47.4</td>
</tr>
<tr>
<td>(psia)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPFT ΔT</td>
<td>-0.0016</td>
<td>24.6</td>
<td>10.8</td>
<td>22.1</td>
</tr>
<tr>
<td>(deg R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPFTP Power</td>
<td>1006.1</td>
<td>-794.7</td>
<td>420.5</td>
<td>-347.4</td>
</tr>
<tr>
<td>(Btu/s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O₂/H₂ Ratio</td>
<td>0.0</td>
<td>-0.0063</td>
<td>0.0089</td>
<td>-0.0004</td>
</tr>
</tbody>
</table>
APPENDIX B

FIGURES
FIGURE B1. MODEL HIERARCHY DIAGRAM

MAIN

START

OPT

PROP

PRPSAT

PRPMIX

ITERP1

OBJF

TPIMP

CURVES

GRAD

ARMijo

BFGS

DFP
FIGURE B2. JUNCTION FLOW BALANCE

\[ M_1 - M_2 - M_3 = \text{FLOW RESIDUAL} \]
(P1-P2) - R * W**2 / Dens(P1,T1) = PRESSURE RESIDUAL

W = M * (gstd/gc)
**FIGURE B4. BRANCH ENERGY BALANCE**

\[ h(P_1, T_1) - h(P_2, T_2) = \text{ENERGY RESIDUAL} \]

\( P_1, P_2 = \text{TOTAL PRESSURES} \)
M1 \times [h(P1,T1) - h(P2,T2)] + M2 \times [h(P3,T3) - h(P4,T4)] = \text{POWER RESIDUAL}
FIGURE B5b. TURBOPUMP BALANCES

\[ \text{Dens}(P_1,T_1) \times (n \times D)^2 \times [\text{CH} - \text{CH}(CQ)] = \text{PUMP DELTA P RESIDUAL} \]

**Definitions:**
- \( \text{CH} \): computed head coefficient
- \( \text{CH}(CQ) \): pump head coef as a function of flow coef
- \( \text{Eff} \): computed pump efficiency
- \( \text{Eff}(CQ) \): pump efficiency as a function of flow coef

\[
\begin{align*}
\frac{M_1 \times (P_2 - P_1)}{\text{Dens}(P_1,T_1)} & \quad - \quad \frac{M_1 \times (P_2 - P_1)}{\text{Dens}(P_1,T_1)} \\
\text{Eff} & \quad - \quad \text{Eff}(CQ)
\end{align*}
\]

\[ = \text{PUMP POWER RESIDUAL} \]
[P3-P4 (assigned)] - [P3-P4 (characteristic)] = TURBINE DELTA P RESIDUAL

[T3-T4 (assigned)] - [T3-T4 (characteristic)] = TURBINE DELTA T RESIDUAL

P3-P4 (assigned) = program assigned delta P
P3-P4 (characteristic) = delta P based on turbine characteristics
= P3 * f[CQ, Ma, gamma(P3, T3)]

T3-T4 (assigned) = program assigned delta T
T3-T4 (characteristic) = delta T based on turbine characteristics
= T3 * f[CQ, Ma, gamma(P3, T3)]

CQ = turbine flow coefficient
Ma = turbine Mach number
gamma(P3, T3) = hot gas specific heat ratio at P3 T3
FIGURE B6a. NOZZLE AND OVERALL BALANCES

P1 T1  M1  FPOV R1

P2 T2  M2  R2 OPOV

P3 T3 M3
FIGURE B6b. NOZZLE AND OVERALL BALANCES

\[
\frac{M_2}{M_1} - \text{COMMANDED RATIO} = \text{MIXTURE RATIO RESIDUAL}
\]

\[
M_1 + M_2 - M_3 = \text{FLOW RESIDUAL 1}
\]

\[
M_3 - M_3(P_3,T_3,\text{nozzle geometry}) = \text{FLOW RESIDUAL 2}
\]

\[
M_1h(P_1,T_1) + M_2h(P_2,T_2) - M_3h(P_3,T_3) = \text{ENERGY RESIDUAL}
\]

\[
(P_1-P_3) - R_1W_1**2/Dens(P_1,T_1) = \text{PRESSURE RESIDUAL 1}
\]

\[
(P_2-P_3) - R_2W_2**2/Dens(P_2,T_2) = \text{PRESSURE RESIDUAL 2}
\]

\[
\text{Thrust}(M_3,P_3,T_3,\text{nozzle geometry}) - \text{COMMAND THRUST} = \text{THRUST RESIDUAL}
\]
FIGURE B7. TEST CONFIGURATION
## C1. DESCRIPTION OF ROUTINES

<table>
<thead>
<tr>
<th>ROUTINE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>variable blocks configured, I/O data files identified, input data read</td>
</tr>
<tr>
<td>START</td>
<td>solution variables initialized for solver, solver output routed to storage files</td>
</tr>
<tr>
<td>PROP</td>
<td>property data arrays initialized, routing based on input property values provided</td>
</tr>
<tr>
<td>PRPSAT</td>
<td>NBS properties of parahydrogen and oxygen near the respective saturation curves calculated</td>
</tr>
<tr>
<td>PRPMIX</td>
<td>properties of gaseous mixtures of parahydrogen, oxygen, and water calculated using the Dalton model</td>
</tr>
<tr>
<td>ITERP2</td>
<td>table constructed functions of two variables interpolated to estimate function value at non-table independent variable values</td>
</tr>
<tr>
<td>ITERP1</td>
<td>table constructed functions of one variable interpolated to estimate function value at non-table independent variable values</td>
</tr>
<tr>
<td>OPT</td>
<td>iterative logic sequence for BFGS-Armijo solver provided</td>
</tr>
<tr>
<td>BFGS</td>
<td>BFGS variable metric Hessian update calculated</td>
</tr>
<tr>
<td>DFP</td>
<td>Davidon-Fletcher-Powell variable metric Hessian update calculated</td>
</tr>
<tr>
<td>ARMIJO</td>
<td>Armijo univariate search conducted to determine residuals minimum in search direction provided by OPT</td>
</tr>
<tr>
<td>OBJF</td>
<td>residuals function of system governing equations calculated</td>
</tr>
<tr>
<td>GRAD</td>
<td>forward difference finite difference approximation of residuals function gradient calculated</td>
</tr>
<tr>
<td>TPIMB</td>
<td>turbine and pump residuals based on turbopump performance curves calculated</td>
</tr>
<tr>
<td>CURVES</td>
<td>turbine and pump performance characteristics provided by manufacturer</td>
</tr>
</tbody>
</table>
C2. INPUT/OUTPUT VARIABLE DEFINITIONS

Program Input File Name = TM2-IO.DAT

Data Type = I/O data file identification

<table>
<thead>
<tr>
<th>INPUT VARIABLE NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDAT</td>
<td>name of file containing fluid property data</td>
</tr>
<tr>
<td>TDAT</td>
<td>name of file containing solution initiation data</td>
</tr>
<tr>
<td>UDAT</td>
<td>name of file containing residual uncertainty estimates</td>
</tr>
<tr>
<td>VDAT</td>
<td>name of file containing volume definition and connection information</td>
</tr>
<tr>
<td>ODAT</td>
<td>name of file containing solution output</td>
</tr>
</tbody>
</table>
Program Input File Name = VDAT

Actual Access File Name = "named in TM2-IO.DAT"
Data Type = system component definition and connection data

<table>
<thead>
<tr>
<th>INPUT VARIABLE NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEC</td>
<td>number of energy circuits in system</td>
</tr>
<tr>
<td>NMC</td>
<td>number of mass flow circuits in system</td>
</tr>
<tr>
<td>NPC</td>
<td>number of pressure circuits in system</td>
</tr>
<tr>
<td>NTPC</td>
<td>number of turbopump circuits in system</td>
</tr>
<tr>
<td>NBRH</td>
<td>number of mass flow branches in system</td>
</tr>
<tr>
<td>NNOD</td>
<td>number of energy (P,T) nodes in system</td>
</tr>
<tr>
<td>NHGNOD</td>
<td>number of hot gas nodes in system</td>
</tr>
<tr>
<td>NRPM</td>
<td>number of turbopump shaft speeds</td>
</tr>
<tr>
<td>INOZBR</td>
<td>mass flow branch number of nozzle flow</td>
</tr>
<tr>
<td>INOZEC</td>
<td>energy circuit number containing nozzle flow</td>
</tr>
<tr>
<td>INOZMC</td>
<td>mass flow circuit number containing nozzle flow</td>
</tr>
<tr>
<td>INOZN</td>
<td>energy node number of nozzle flow</td>
</tr>
<tr>
<td>IHGNOZ</td>
<td>hot gas node number of nozzle flow</td>
</tr>
<tr>
<td>IMIXR</td>
<td>TTB array location containing mixture ratio value</td>
</tr>
<tr>
<td>IATHRT</td>
<td>TTB array location containing nozzle throat area value</td>
</tr>
<tr>
<td>IA(I)</td>
<td>(not used)</td>
</tr>
<tr>
<td>IM(I)</td>
<td>TTB array location containing initial estimate of mass flow rate through branch I</td>
</tr>
<tr>
<td>IP(I)</td>
<td>TTB array location containing initial estimate of pressure value at node I</td>
</tr>
<tr>
<td>IR(I)</td>
<td>TTB array location containing initial estimate of flow resistance value for branch I</td>
</tr>
<tr>
<td>IS(I)</td>
<td>TTB array location containing initial estimate of turbopump I shaft speed</td>
</tr>
<tr>
<td>IT(I)</td>
<td>TTB array location containing initial estimate of temperature value at node I</td>
</tr>
<tr>
<td>MAT(I)</td>
<td>material type number of flow at node I</td>
</tr>
<tr>
<td>1 = hydrogen (H2)</td>
<td>2 = oxygen (O2)</td>
</tr>
<tr>
<td>3+ = hot gas mixture</td>
<td></td>
</tr>
<tr>
<td>NEIO(I)</td>
<td>number of mass flow I/O's across boundary of energy circuit I</td>
</tr>
<tr>
<td>EDIR(I,J)</td>
<td>direction number of mass flow J across boundary of energy circuit I</td>
</tr>
<tr>
<td>1 = inflow</td>
<td>-1 = outflow</td>
</tr>
<tr>
<td>IEN(I,J)</td>
<td>node number associated with flow J across boundary of energy circuit I</td>
</tr>
<tr>
<td>IMBEC(I,J)</td>
<td>branch number of mass flow J across boundary of energy circuit I</td>
</tr>
<tr>
<td>NMIO(I)</td>
<td>number of mass flow I/O's across boundary of mass circuit I</td>
</tr>
<tr>
<td>MDIR(I,J)</td>
<td>direction number of mass flow J across boundary of mass circuit I</td>
</tr>
<tr>
<td>1 = inflow</td>
<td>-1 = outflow</td>
</tr>
<tr>
<td>IMBMC(I,J)</td>
<td>branch number of mass flow J across boundary of mass circuit I</td>
</tr>
</tbody>
</table>
**Program Input File Name** = *VDAT* (continued 2)

**INPUT VARIABLE NAME** | **DEFINITION**
---|---
PDIR(I,J) | direction number of mass flow J across boundary of pressure circuit I  
| 1 = inflow  
| -1 = outflow
IPN(I,J) | node number of mass flow J across boundary of pressure circuit I
IMBPC(I,J) | branch number of mass flow J across boundary of pressure circuit I
NH2HG(I) | number of pure hydrogen flows contributing to hot gas flow I
NO2HG(I) | number of pure oxygen flows contributing to hot gas flow I
IHGN(I) | node number associated with hot gas flow I
ICEFF(I) | TTB array location containing the combustion efficiency value associated with the flow at hot gas node I
IH2HG(I,J) | branch number of hydrogen flow J entering hot gas flow I
IO2HG(I,J) | branch number of oxygen flow J entering hot gas flow I
IBTYPE(I) | type number of branch flow I  
| 0 = fixed  
| >0 = variable
INTYPE(I) | type number of node I  
| 0 = fixed pressure and temperature  
| 1 = variable pressure, fixed temperature  
| >1 = variable pressure and temperature
IRTYPE(I) | type number of resistance in pressure circuit I  
| 0 = fixed resistance  
| >0 = variable resistance
ITPTY(I) | type number of turbopump I  
| 1 = one turbine, one pump  
| >1 = one turbine, two pumps
ITPD(I,J) | TTB array location containing the impeller diameter of turbomachine J in turbopump circuit I
| J = 1 turbine  
| J = 2 or 3 pump
ITPN(I,J) | node number associated with mass flow J crossing boundary of turbopump I  
| J = 1 turbine inlet  
| J = 2 turbine outlet  
| J = 3 pump 1 inlet  
| J = 4 pump 1 outlet  
| J = 5 pump 2 inlet  
| J = 6 pump 2 outlet
ITPS(I) | shaft number associated with turbopump I
IMBTP(I,J) | branch number associated with mass flow J crossing boundary of turbopump I  
| J = 1 turbine inlet  
| J = 2 turbine outlet  
| J = 3 pump 1 inlet  
| J = 4 pump 1 outlet  
| J = 5 pump 2 inlet  
| J = 6 pump 2 outlet
Program Input File Name = VDAT (continued 3)

<table>
<thead>
<tr>
<th>INPUT VARIABLE NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICURV(I,J)</td>
<td>curve number associated with performance curve J of turbopump I</td>
</tr>
<tr>
<td></td>
<td>J = 1 turbine performance curve 1</td>
</tr>
<tr>
<td></td>
<td>J = 2 turbine performance curve 2</td>
</tr>
<tr>
<td></td>
<td>J = 3 pump 1 performance curve 1</td>
</tr>
<tr>
<td></td>
<td>J = 4 pump 1 performance curve 2</td>
</tr>
<tr>
<td></td>
<td>J = 5 pump 2 performance curve 1</td>
</tr>
<tr>
<td></td>
<td>J = 6 pump 2 performance curve 2</td>
</tr>
</tbody>
</table>
Program Input File Name = TDAT

Actual Access File Name = "named in TM2-IO.DAT"
Data Type = analysis description and solution search initializing values

<table>
<thead>
<tr>
<th>INPUT VARIABLE NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDESC</td>
<td>number of analysis description items</td>
</tr>
<tr>
<td>NTTB</td>
<td>number of elements in the TTB solution initializing array</td>
</tr>
<tr>
<td>DESC(I)</td>
<td>analysis description item I (character variable)</td>
</tr>
<tr>
<td>TTB(I)</td>
<td>value of solution initializing variable assigned to array address I</td>
</tr>
</tbody>
</table>
Program Input File Name = UDAT
Actual Access File Name = "named in TM2-IO.DAT"
Data Type = uncertainty values associated with model relations

<table>
<thead>
<tr>
<th>INPUT VARIABLE NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOF</td>
<td>uncertainty of engine oxygen/fuel ratio</td>
</tr>
<tr>
<td>UEVOL(I)</td>
<td>power imbalance uncertainty associated with energy circuit I</td>
</tr>
<tr>
<td>UMVOL(I)</td>
<td>mass flow rate imbalance uncertainty associated with mass flow circuit I</td>
</tr>
<tr>
<td>UPVOL(I)</td>
<td>pressure imbalance uncertainty associated with pressure circuit I</td>
</tr>
<tr>
<td>UETP(I)</td>
<td>power imbalance uncertainty associated with turbopump circuit I</td>
</tr>
<tr>
<td>UT1TP(I)</td>
<td>turbine characteristic 1 (pressure drop) imbalance uncertainty associated with turbopump circuit I</td>
</tr>
<tr>
<td>UT2TP(I)</td>
<td>turbine characteristic 2 (temperature drop) imbalance uncertainty associated with turbopump circuit I</td>
</tr>
<tr>
<td>UP1TP(I)</td>
<td>pump 1 characteristic 1 (pressure drop) imbalance uncertainty associated with turbopump circuit I</td>
</tr>
<tr>
<td>UP2TP(I)</td>
<td>pump 1 characteristic 2 (power) imbalance uncertainty associated with turbopump circuit I</td>
</tr>
<tr>
<td>UP3TP(I)</td>
<td>pump 2 characteristic 1 (pressure drop) imbalance uncertainty associated with turbopump circuit I</td>
</tr>
<tr>
<td>UP4TP(I)</td>
<td>pump 2 characteristic 2 (power) imbalance uncertainty associated with turbopump circuit I</td>
</tr>
</tbody>
</table>
Program Output File Name = ODAT

Actual Access File Name = "named in TM2.IO.DAT"
Data Type = output defining circuit definitions, solution values, and residuals (both initial and final)

<table>
<thead>
<tr>
<th>INPUT VARIABLE NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEC</td>
<td>number of energy circuits in system</td>
</tr>
<tr>
<td>NMC</td>
<td>number of mass flow circuits in system</td>
</tr>
<tr>
<td>NPC</td>
<td>number of pressure circuits in system</td>
</tr>
<tr>
<td>NTPC</td>
<td>number of turbopump circuits in system</td>
</tr>
<tr>
<td>IEN(I,J)</td>
<td>node number associated with flow J across boundary of energy circuit I</td>
</tr>
<tr>
<td>IMBEC(I,J)</td>
<td>branch number of mass flow J across boundary of energy circuit I</td>
</tr>
<tr>
<td>IMBMC(I,J)</td>
<td>branch number of mass flow J across boundary of mass circuit I</td>
</tr>
<tr>
<td>IPN(I,J)</td>
<td>node number of mass flow J across boundary of pressure circuit I</td>
</tr>
<tr>
<td>IMBPC(I,J)</td>
<td>branch number of mass flow J across boundary of pressure circuit I</td>
</tr>
<tr>
<td>ITPN(I,J)</td>
<td>node number associated with mass flow J crossing boundary of turbopump I</td>
</tr>
<tr>
<td>IMBTP(I,J)</td>
<td>branch number associated with mass flow J crossing boundary of turbopump I</td>
</tr>
<tr>
<td>ETP0(I)</td>
<td>initial power imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>ETP(I)</td>
<td>final power imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>T1TP0(I)</td>
<td>initial turbine characteristic 1 (pressure drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>T1TP(I)</td>
<td>final turbine characteristic 1 (pressure drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>T2TP0(I)</td>
<td>initial turbine characteristic 2 (temperature drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>T2TP(I)</td>
<td>final turbine characteristic 2 (temperature drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>P1TP0(I)</td>
<td>initial pump 1 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>P1TP(I)</td>
<td>final pump 1 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>P2TP0(I)</td>
<td>initial pump 1 characteristic 2 (power) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>P2TP(I)</td>
<td>final pump 1 characteristic 2 (power) imbalance associated with turbopump circuit I</td>
</tr>
</tbody>
</table>
Program Output File Name = ODAT (continued 2)

<table>
<thead>
<tr>
<th>INPUT VARIABLE NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3TP0(I)</td>
<td>initial pump 2 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>P3TP(I)</td>
<td>final pump 2 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>P4TP0(I)</td>
<td>initial pump 2 characteristic 2 (power) imbalance associated with turbopump circuit I</td>
</tr>
<tr>
<td>P4TP(I)</td>
<td>final pump 2 characteristic 2 (power) imbalance associated with turbopump circuit I</td>
</tr>
</tbody>
</table>
C3. SOURCE CODE LISTING
PROGRAM TM-02

CHARACTER*24 DESC

CHARACTER*12 PDAT, TDAT, UDAT, VDAT, ODAT

INTEGER EDIR, PDIR

COMMON /Balc/ EIMB(5), PIMB(6), WIMB(5), WZIMB, OFIMB,

1 EIMBO(5), PIMBO(6), WIMBO(5), WZIMBO, OFIMBO,

3 ETP(4), T1TP(4), T2TP(4),

4 P1TP(4), P2TP(4), P3TP(4), P4TP(4),

6 ETP0(4), T1TP0(4), T2TP0(4),

7 P1TP0(4), P2TP0(4), P3TP0(4), P4TP0(4),

1 REV(20), REV(20),

2 REV(20), REV(20),

COMMON /Vdat/ NEC, NMC, NPC, NTPC, NBRH, NNOD, NHGND, NRPM,

1 INOZBR, INOZEC, INOZMC, INOZN, IHGNZ, IMIXR, IATHRT,

2 IA(20), IM(20), IP(20), IR(6), IS(5), IT(20), MAT(20),

3 NEIO(5), EDIR(5, 20), IEN(5, 20), IMBEC(5, 20),

4 NMIO(5), MDIR(5, 20), IMBMC(5, 20),

5 IPN(6, 2), IMBPC(6),

6 IHGN(5), ICEFF(5), IH2HG(5, 5), I02HG(5, 5),

7 ITPY(4), ITPY(4),

1 ITPD(4, 3), ITPD(4, 3),

2 ITPD(4, 3), ITPD(4, 3),

COMMON /Udat/ UOF, UEVOL(5), UMVOL(5), UPVOL(6),

1 UETP(4), UT1TP(4), UT2TP(4),

3 UP2TP(4), UP2TP(4),

COMMON /H2prp/ H2P1(15), H2T1(11), H2H1(15, 11), H2S1(15, 11), H2D1(15, 11),

* H2P2(20), H2T2(11), H2H2(20, 11), H2S2(20, 11), H2D2(20, 11),

* H2P3(29), H2T3(25), H2H3(29, 25), H2S3(29, 25), H2D3(29, 25),

* H2P4(23), H2T4(25), H2H4(23, 25), H2S4(23, 25), H2D4(23, 25),

COMMON /O2prp/

* O2P1(13), O2T1(16), O2H1(13, 16), O2S1(13, 16), O2D1(13, 16),

* O2P2(13), O2T2(17), O2H2(13, 17), O2S2(13, 17), O2D2(13, 17),

* O2P3(5), O2T3(61), O2H3(5, 61), O2S3(5, 61), O2D3(5, 61),

* H2Oprp/

* H2O1(7), H2OT1(13), H2OH1(7, 13), H2OS1(7, 13), H2OD1(7, 13),

COMMON /Table/

* NH2P(4), NH2T(4), NO2P(3), NO2T(3), NH2OP(1), NH2OT(1)

COMMON /Std/ * HH2ref, HO2ref, Hwaref, SH2ref, SO2ref, SWaref, SH2a, SO2a,

* SWaa

DIMENSION * NH2PA(4), NH2TA(4), NO2PA(3), NO2TA(3), NH2OPA(1), NH2OTA(1)

CHARACTER*70 PTITLE

DATA (NH2PA(I), I=1,4)/15, 20, 29, 23/

DATA (NH2TA(J), J=1,4)/11, 11, 25, 25/

DATA (NO2PA(I), I=1,3)/13, 13, 5/

DATA (NO2TA(J), J=1,3)/16, 17, 61/

DATA (NH2OPA(I), I=1,1)/7/

DATA (NH2OTA(J), J=1,1)/7/

DO 90 I=1,4

NH2P(I)=NH2PA(I)

90 NH2T(I)=NH2TA(I)

DO 91 I=1,3

NO2P(I)=NO2PA(I)

91 NO2T(I)=NO2TA(I)

NH2OP(I)=NH2OPA(I)

NH2OT(I)=NH2OTA(I)

HH2REF = 1790.091

H2REF = 234.681

HWAREF = 1339.990

SH2REF = 15.440

SO2REF = 1.530

SWAREF = 2.294

SH2A = 15.481
OPEN ( 7, FILE = 'TM1-IO.DAT', STATUS = 'OLD' )
READ ( 7, * ) PDAT, TDAT, UDAT, VDAT, ODAT

OPEN ( 8, FILE = PDAT, STATUS = 'OLD' )
OPEN ( 12, FILE = TDAT, STATUS = 'OLD' )
OPEN ( 13, FILE = UDAT, STATUS = 'OLD' )
OPEN ( 14, FILE = VDAT, STATUS = 'OLD' )
OPEN ( 21, FILE = ODAT )

** READ IN H2 PROPERTY TABLE INTO ARRAYS **
DO 10 ITBL=1,4
READ(8,902) PTITLE
DO 10 I=1,NH2P(ITBL)
DO 10 J=1,NH2T(ITBL)
IF(ITBL.EQ.1) READ(8,*) H2P1(I),H2T1(J),
  1 H2H1(I,J),H2S1(I,J),H2D1(I,J)
IF(ITBL.EQ.2) READ(8,*) H2P2(I),H2T2(J),
  1 H2H2(I,J),H2S2(I,J),H2D2(I,J)
IF(ITBL.EQ.3) READ(8,*) H2P3(I),H2T3(J),
  1 H2H3(I,J),H2S3(I,J),H2D3(I,J)
IF(ITBL.EQ.4) READ(8,*) H2P4(I),H2T4(J),
  1 H2H4(I,J),H2S4(I,J),H2D4(I,J)
10 CONTINUE

** READ IN O2 PROPERTY TABLE INTO ARRAYS **
DO 20 ITBL=1,3
READ(8,902) PTITLE
DO 20 I=1,NO2P(ITBL)
DO 20 J=1,NO2T(ITBL)
IF(ITBL.EQ.1) READ(8,*) O2P1(I),O2T1(J),
  1 O2H1(I,J),O2S1(I,J),O2D1(I,J)
IF(ITBL.EQ.2) READ(8,*) O2P2(I),O2T2(J),
  1 O2H2(I,J),O2S2(I,J),O2D2(I,J)
IF(ITBL.EQ.3) READ(8,*) O2P3(I),O2T3(J),
  1 O2H3(I,J),O2S3(I,J),O2D3(I,J)
20 CONTINUE

** READ IN STEAM PROPERTY TABLES INTO ARRAYS **
DO 30 ITBL = 1, 1
READ(8,902) PTITLE
DO 30 I = 1, NH2OP(ITBL)
DO 30 J = 1, NH2OT(ITBL)
IF(ITBL.EQ.1) READ(8,*) H2OP1(I),H2OT1(J),
  1 H2OH1(I,J),H2OS1(I,J),H2OD1(I,J)
30 CONTINUE

READ (12,*) NDESC, NTTB
READ (12,903) ( DESC( I ), I = 1, NDESC )
WRITE (21,901) ( DESC( I ), I = 1, NDESC )
C
DO 40 I = 1, NMC
READ (14,*) ( MDIR(I,J), J = 1, NMC(I) ),
DO 50 I = 1, NEC
READ (14,*) ( EDIR(I,J), J = 1, NEC(I) ),
DO 60 I = 1, NPC
READ (14,*) ( PDIR(I,J), J = 1, 2 ),
IF ( NHGNO .GT. 0 ) THEN
READ (14,*) ( IHGN(I), I = 1, NHGNO ),
DO 70 I = 1, NHGNO
READ (14,*) ( IF2HGN(I,J), J = 1, NO2HGN(I) )
ENDIF
READ (14,*) ( ITFTY(I), I = 1, NTPC ),
DO 80 I = 1, NTPC
IF ( ITFTY(I) .EQ. 1 ) THEN
READ (14,*) ( ITPD(I,J), J = 1, 2 ),
ELSE
READ (14,*) ( ITPD(I,J), J = 1, 3 ),
ENDIF
READ (13,*) ( UEVL(I), I = 1, NEC ),
CALL START
901 FORMAT ( 10 (/ , 1X, A24 ) )
902 FORMAT ( /A70/ )
903 FORMAT ( 8X, 5E13.6 )
STOP
END
C
SUBROUTINE START
C
START ROUTINE
C
CHARACTER*24 DESC
INTEGER EDIR, PDIR

REAL JOULE

DIMENSION X(25), X2(25)

COMMON /BALC/ EIMB(5), PIMB(6), WIMB(5), WZIMB, OFIMB,
1 EIMB(5), PIMB(6), WIMB(5), WZIMB, OFIMB,
3 ETP(4), TTP(4), T2TP(4),
4 P1TP(4), P2TP(4), P3TP(4), P4TP(4),
6 ETP(4), TTP(4), T2TP(4),
7 P1TP(4), P2TP(4), P3TP(4), P4TP(4),
1 REVA(20), REV(20), RVP(20), REV(20),
2 REVR(20), REV(20), RVP(20), REV(20),

COMMON /VDAT/ NEC, NMC, NPC, NTPC, NBRH, NNOD, NHGND, NRPM,
1 INOZBR, INOZEC, INOZMC, INOZN, IHNZN, IMIXR, IATHRT,
2 IA(20), IM(20), IP(20), IR(6), IS(5), IT(20), MAT(20),
3 NEIO(5), EDIR(5, 20), IEI(5, 20), IMBEC(5, 20),
4 NMIO(5), MDIR(5, 20), IMBEC(5, 20), PDIR(6, 2),
5 IPN(6, 2), IMBPC(6), NH2HG(5), NO2HG(5),
6 IHGN(5), ICEFF(5), IH2HG(5, 5), IO2HG(5, 5),
7 ITBTYPE(20), INTYPE(20), INTYPE(20), ITPTY(4),
1 ITPD(4, 3), ITTP(4, 3), ITPS(4), IMBTP(4, 3),
2 ICRV(4, 6),

COMMON /UDAT/ UOF, UEVOL(5), UVMOL(5), UPVOL(6),
1 UETP(4), UT1TP(4), UT2TP(4), UP1TP(4),
3 UP2TP(4), UP3TP(4), UP4TP(4),

COMMON /H2PRP/
1 H2P1(15), H2T1(11), H2H1(15, 11), H2S1(15, 11), H2D1(15, 11),
2 H2P2(20), H2T2(12), H2H2(20, 12), H2S2(20, 12), H2D2(20, 12),
3 H2P3(29), H2T3(25), H2H3(29, 25), H2S3(29, 25), H2D3(29, 25),
4 H2P4(23), H2T4(25), H2H4(23, 25), H2S4(23, 25), H2D4(23, 25),

COMMON /O2PRP/
1 O2P1(13), O2T1(16), O2H1(13, 16), O2S1(13, 16), O2D1(13, 16),
2 O2P2(13), O2T2(17), O2H2(13, 17), O2S2(13, 17), O2D2(13, 17),
3 O2P3(5), O2T3(61), O2H3(5, 61), O2S3(5, 61), O2D3(5, 61),

COMMON /H2OPRP/
1 H2OP1(7), H2OT1(13), H2OH1(7, 13), H2OS1(7, 13), H2OD1(7, 13),

COMMON /STD/
1 HH2REF, HO2REF, HWAREF, SH2REF, SO2REF, SWAREF, SH2A, SO2A,
2 SWAA

COMMON /TABLE/
1 NH2P(4), NH2T(4), NO2P(3), NO2T(3), NH2OP(1), NH2OT(1)

PARAMETER ( JOULE = 778.16, GC = 32.174 )

IFUNC = 0

WRITE (21, 941)
DO 2 IMC = 1, NMC
WRITE (21, 942) IMC, (IMBMC(IMC, IIO), IIO = 1, NMIO(IMC))
2 CONTINUE

WRITE (21, 943)
DO 4 IEC = 1, NEC
WRITE (21, 942) IEC, (IMBEC(IEC, IIO), IIO = 1, NEIO(IEC))
4 CONTINUE

WRITE (21, 945)
DO 6 IPC = 1, NPC
WRITE (21, 942) IPC, IMBPC(IPC), IMBPC(IPC)
WRITE (21, 944) IPC, (IPN(IPC, IIO), IIO = 1, 2)
6 CONTINUE

WRITE (21, 946)
DO 7 ITPC = 1, NTPC
WRITE (21, 942) ITPC, (IMBTP(ITPC, J), J = 1, 3)
WRITE (21, 944) ITPC, (ITPN(ITPC, J), J = 1, 6)
7 CONTINUE

****** INITIALIZE ALGORITHM *************
NMAX = 100
EPSC=1.E-6
EPST=1.E-6
RHO=1.0
ALPHA=0.0
BETA=0.4

C ********** INITIALIZE INDEPENDENT VARIABLES X(I) **********

K=1
DO 10 I=1,NBRH
  REVM(I)=TTB(IM(I))
  IF (IBTYPE(I).EQ.0) THEN
    GO TO 10
  ELSE
    X(K)=SQRT(REVM(I))
    K=K+1
  ENDIF
10 CONTINUE

DO 20 I=1,NNOD
  TTB(IP(I))
  REVT(I)=TTB(IT(I))
  IF (INTYPE(I).EQ.0) THEN
    GO TO 20
  ELSE
    IF (I.EQ.INOZN) THEN
      X(K)=SQRT(REVT(I))
      K=K+1
    ELSE
      X(K)=SQRT(REVP(I))
      X(K+1)=SQRT(REVT(I))
      K=K+2
    ENDIF
  ENDIF
20 CONTINUE

DO 40 I=1,NPC
  TTB(IR(I))
  IF (IRTYPE(I).EQ.0) THEN
    GO TO 40
  ELSE
    X(K)=SQRT(REVR(I))
    K=K+1
  ENDIF
40 CONTINUE

DO 50 I=1,NRPM
  TTB(IS(I))
  X(K)=SQRT(REVS(I))
  K=K+1
50 CONTINUE

N=K-1

DO 100 I=1,NNOD
  P=REVP(I)
  T=REVT(I)
  IF (MAT(I).GE.3) GO TO 70
  IF (MAT(I).GE.2) GO TO 60
  CALL PROP(1,P,T,0.0,0.0,H,S,RHO)
  REVD(I)=RHO
  REVH(I)=H-HH2REF
  GO TO 100

60 CALL PROP(2,P,T,0.0,0.0,H,S,RHO)
  REVD(I)=RHO
  REVH(I)=H-HO2REF
  GO TO 100

70 IDHG=1
72 IF (IHGN(IDHG).EQ.1) THEN
  IHG=IDHG
  GO TO 74
ELSE
  IDHG=IDHG+1
  IF (IDHG.GT.NHGNOD) THEN

GO TO 74
ELSE
GO TO 72
ENDIF
ENDIF

74 WH2=0.0
DO 80 IH2IN=1,NH2HG(IHG)
IBRH=IH2HG(IHG,IH2IN)
WH2=WH2+REVM(IBRH)
80 CONTINUE

90 WO2=0.0
DO 90 IO2IN=1,NO2HG(IHG)
IBRH=IO2HG(IHG,IO2IN)
WO2=WO2+REVM(IBRH)
90 CONTINUE

CEFF=TTB(ICEFF(IHG))
OF=WO2/WH2
CALL PROP ( 4, P, T, OF, CEFF, H, S, RHO)
REVD(I)=RHO
REVH(I)=H

100 CONTINUE

******* CALL OPTIMIZATION STRATEGY *********
CALL OPT (N,NMAX,EPSC,EPST,RHO,ALPHA,BETA,ISTAGE,F2,GNORM,
1 DXNORM,X,X2)

WRITE (21,801)
DO 200 I=1,NBRH
WRITE (21,802) I,TTB(IM(I)),REVM(I)
200 CONTINUE

WRITE (21,803) I,TTB(IP(I)),REVP(I)
210 CONTINUE

WRITE (21,804) I,TTB(IT(I)),REVT(I)
220 CONTINUE

WRITE (21,805) I,TTB(IS(I)),REVS(I)
230 CONTINUE

WRITE (21,806)
DO 240 I=1,NMC
WRITE (21,807) I,WIMB0(I),WIMB(I)
240 CONTINUE

WRITE (21,808) I,EIMB0(I),EIMB(I)
250 CONTINUE

WRITE (21,809) I,PIMB0(I),PIMB(I)
260 CONTINUE

DO 270 I=1,NTPC
IF (ITPTY(I).EQ.1) THEN
WRITE (21,810) I,ETP0(I),ETP(I)
1 I,T1TP0(I),T1TP(I)
2 I,T2TP0(I),T2TP(I)
3 I,P1TP0(I),P1TP(I)
4 I,P2TP0(I),P2TP(I)
ELSE
WRITE (21,811) I,ETP0(I),ETP(I)
1 I,T1TP0(I),T1TP(I)
2 I,T2TP0(I),T2TP(I)
3 I,P1TP0(I),P1TP(I)
4 I,P2TP0(I),P2TP(I)
5 I,P3TP0(I),P3TP(I)
6 I,P4TP0(I),P4TP(I)
ENDIF
270 CONTINUE
SUBROUTINE OBJF (IFUNC,X,F)

CHARACTER*24 DESC
INTEGER EDIR, PDIR
REAL JOULE

DIMENSION X(25)

COMMON /Balc/ EIMB(5), PIMB(6), WIMB(5), WZIMB, OFIMB
1 EIMBO(5), PMIBO(6), WIMBO(5), WZIMBO, OFIMBO,
3 ETP(4), T1TP(4), T2TP(4), P4P(4)
4 ETP0(4), T1TP0(4), T2TP0(4), P4P0(4)
6 P1TP(4), P2TP(4), P3TP(4), P4TP(4)
7 REV(20), REV(20), REVP(20), REV(20), REV(20)
2 REVR(20), REVS(20), REVD(20), REVH(20)

COMMON /Vdat/ NEC, NMC, NPC, NTPC, NBRH, NNOD, NHG, NHG=false, NRPM,
1 INOZB, INOZC, INO2MC, INO2N, NHG, IMIXR, IATHRT,
2 IA(20), IM(20), IP(20), IR(6), IS(5), IT(20), MAT(20),
3 NEIO(5), EDIR(5,20), IEN(5,20), IMBEC(5,20),
4 NMIO(5), MDIR(5,20), IMBMC(5,20), PDIR(6,2),
6 ICEFF 5, IH2HG 5, IO2HG 5, IBTYPE 20, INTYP(0),
7 ITPD(4), ITPN(4,6), ITPS(4), IMBTP(4,3),
2 ICURV(4,6)
COMMON /TDAT/ NDESC, NTTB, DESC(5), TTB(1350)

COMMON /UDAT/ UOF, UEVOL(5), UMVOL(5), UPVOL(6),
1 UETP(4), UT1TP(4), UT2TP(4), UP1TP(4),
3 UP2TP(4), UP3TP(4), UP4TP(4)

COMMOM /H2PRP/
1 H2P1(15), H2T1(11), H2H1(15,11), H2S1(15,11), H2D1(15,11),
2 H2P2(20), H2T2(11), H2H2(20,11), H2S2(20,11), H2D2(20,11),
3 H2P3(29), H2T3(25), H2H3(29,25), H2S3(29,25), H2D3(29,25),
4 H2P4(23), H2T4(25), H2H4(23,25), H2S4(23,25), H2D4(23,25)

COMMOM /O2PRP/
1 O2P1(13), O2T1(16), O2H1(13,16), O2S1(13,16), O2D1(13,16),
2 O2P2(13), O2T2(17), O2H2(13,17), O2S2(13,17), O2D2(13,17),
3 O2P3(5), O2T3(61), O2H3(5,61), O2S3(5,61), O2D3(5,61)

COMMOM /H2OPR/
1 H2OP1(7), H2OT1(13), H2OH1(7,13), H2OS1(7,13), H2OD1(7,13)

COMMOM /STD/
1 HH2REF, H2O2REF, HWAREF, SH2REF, SO2REF, SWAREF, SH2A, SO2A,
2 SWAA

COMMOM /TABLE/
1 NH2P(4), NH2T(4), NO2P(3), NO2T(3), NH2OP(1), NH2OT(1)

PARAMETER ( JOULE = 778.16, GC = 32.174 )

IFUNC=IFUNC+1

K=1
DO 10 I=1, NBRH
IF (IBTYPE(I).EQ.0) THEN
GO TO 10
ELSE
  REVM(I)=X(K)**2
  K=K+1
ENDIF

CONTINUE

10 DO 20 I=1, NNOD
IF (INTYPE(I).EQ.0) THEN
GO TO 20
ELSE
  IF (I.EQ.INOZN) THEN
    REVT(I)=X(K)**2
    K=K+1
  ELSE
    REVP(I)=X(K)**2
    REVT(I)=X(K+1)**2
    K=K+2
  ENDIF
ENDIF

CONTINUE

20 CONTINUE

DO 40 I=1, NPC
IF (IRTYPE(I).EQ.0) THEN
GO TO 40
ELSE
  REVR(I)=X(K)**2
  K=K+1
ENDIF

CONTINUE

40 CONTINUE

DO 50 I=1, NRPM
REVS(I)=X(K)**2
K=K+1

50 CONTINUE

DO 100 I=1, NNOD
IF (INTYPE(I).EQ.0) THEN
GO TO 100
ELSE
  P=REVP(I)
  T=REVT(I)
  IF (MAT(I) .GE. 3) GO TO 70
  IF (MAT(I) .GE. 2) GO TO 60

C
CALL PROP ( 1, P, T, 0.0, 0.0, H, S, RHO)
REVH(I) = RHO
GO TO 100

CALL PROP ( 2, P, T, 0.0, 0.0, H, S, RHO)
REVH(I) = RHO
GO TO 100

IDHG = 1
IF (IHGN(IDHG).EQ.I) THEN
   IHG = IDHG
   GO TO 74
ELSE
   IDHG = IDHG + 1
   IF (IDHG.GT.NHGNO) THEN
      GO TO 74
   ELSE
      GO TO 72
   ENDIF
ENDIF

WH2 = 0.0
DO 80 IH2IN = 1, NH2HG(IHG)
   IBRH = IH2HG(IHG, IH2IN)
   WH2 = WH2 + REVM(IBRH)
CONTINUE

WO2 = 0.0
DO 90 IO2IN = 1, NO2HG(IHG)
   IBRH = IO2HG(IHG, IO2IN)
   WO2 = WO2 + REVM(IBRH)
CONTINUE

CEFF = TTB(ICEFF(IHG))
OF = WO2 / WH2
CALL PROP ( 4, P, T, 0.0, OF, CEFF, H, S, RHO)
REVH(I) = H

ENDIF

CONTINUE

F = 0.0
DO 120 IMC = 1, NMC
   WIMB(IMC) = 0.0
   DO 110 IIO = 1, NMIO(IMC)
      IOD = MDIR(IMC, IIO)
      IBRH = IMBMC(IMC, IIO)
      WIMB(IMC) = WIMB(IMC) + IOD*REVM(IBRH)
   CONTINUE
   F = F + (WIMB(IMC)/UMVOL(IMC))**2
CONTINUE

DO 140 IEC = 1, NEC
   EIMB(IEC) = 0.0
   DO 130 IIO = 1, NEIO(IEC)
      IOD = EDIR(IEC, IIO)
      INOD = IEN(IEC, IIO)
      IBRH = IMBEC(IEC, IIO)
      ENTH = REVM(INOD)
      W = REVM(IBRH)
      EIMB(IEC) = EIMB(IEC) + IOD*W*ENTH
   CONTINUE
   HEAT = f(RefTi, RefT2, RefMi, RefM2, BoundaryProps)
   EIMB(IEC) = EIMB(IEC) + HEAT
   F = F + (EIMB(IEC)/UEVOL(IEC))**2
CONTINUE

DO 160 IPC = 1, NPC
   IBRH = IMBPC(IPC)
   RES = REVR(IPC)
   CONTINUE
w=REVW(IBM)H
DO 150 I10=1,2
I0D=PDfT(IPC,I10)
INOD=ITP(N(IPC,110)
IF (I0D.GE.0) THEN
   PIN=REVP(INOD)
   RHO=REVD(INOD)
ELSE
   POUT=REVP(INOD)
ENDIF
150 CONTINUE
PIMB(IPC)=PIN-POUT-RES*W**2/RHO
C
F=F+(PIMB(IPC)/UPOVOL(IPC))**2
C
160 CONTINUE
C
DO 170 I=1,NTPC
IF (ITPTY(I).EQ.1) THEN
   IDIA1=ITPD(I,1)
   IDIA2=ITPD(I,2)
   IBRH1=MBTP(I,1)
   IBRH2=MBTP(I,2)
   INOD1=ITPN(I,1)
   INOD2=ITPN(I,2)
   INOD3=ITPN(I,3)
   INOD4=ITPN(I,4)
   ICUR1=ICURV(I,1)
   ICUR2=ICURV(I,2)
   ICUR3=ICURV(I,3)
   ICUR4=ICURV(I,4)
   ISPEED=ITPS(I)
C
   DIA1=TTB(IDIA1)
   DIA2=TTB(IDIA2)
   W1=REVW(IBM)
   W2=REVW(IBM)
   P1=REVP(INOD1)
   P2=REVP(INOD2)
   P3=REVP(INOD3)
   P4=REVP(INOD4)
   T1=REVW(INOD1)
   T2=REVW(INOD2)
   T3=REVW(INOD3)
   T4=REVW(INOD4)
   D1=REVD(INOD1)
   D2=REVD(INOD2)
   D3=REVD(INOD3)
   D4=REVD(INOD4)
   H1=REVU(INOD1)
   H2=REVU(INOD2)
   H3=REVU(INOD3)
   H4=REVU(INOD4)
   SPEDE=REV(RSPEED)
C
   ETPl(I)=W1*H1+W2*H3-W1*H2-W2*H4
   CALL TPIMB (I,ICUR1,ICUR2,DIA1,W1,SPEED,D1,D2,P1,P2,T1,T2,H1,
   1   H2,T1TP(I),T2TP(I))
   CALL TPIMB (I,ICUR3,ICUR4,DIA2,W2,SPEED,D3,D4,P3,P4,T3,T4,H3,
   1   H4,P1TP(I),P2TP(I))
C
   F = F + (ETP(I) / UETP(I)) **2 +
   1   T1TP(I) / U1TP(I) **2 +
   1   T2TP(I) / U2TP(I) **2 +
   1   P1TP(I) / UP1TP(I) **2 +
   4   P2TP(I) / UP2TP(I) **2
C
ELSE
   IDIA1=ITPD(I,1)
   IDIA2=ITPD(I,2)
   IDIA3=ITPD(I,3)
   IBRH1=MBTP(I,1)
   IBRH2=MBTP(I,2)
   IBRH3=MBTP(I,3)
   INOD1=ITPN(I,1)
   INOD2=ITPN(I,2)
   INOD3=ITPN(I,3)
   INOD4=ITPN(I,4)
INOD5=ITPN(I,5)
INOD6=ITPN(I,6)
ICUR1=ICURV(I,1)
ICUR2=ICURV(I,2)
ICUR3=ICURV(I,3)
ICUR4=ICURV(I,4)
ICUR5=ICURV(I,5)
ICUR6=ICURV(I,6)
ISPEED=ITPS(I)

C

DIA1=TTB(IDIA1)
DIA2=TTB(IDIA2)
DIA3=TTB(IDIA3)
W1=REVM(IBRH1)
W2=REVM(IBRH2)
W3=REVM(IBRH3)
P1=REVP(INOD1)
P2=REVP(INOD2)
P3=REVP(INOD3)
P4=REVP(INOD4)
P5=REVP(INOD5)
P6=REVP(INOD6)
T1=REVT(INOD1)
T2=REVT(INOD2)
T3=REVT(INOD3)
T4=REVT(INOD4)
T5=REVT(INOD5)
T6=REVT(INOD6)
D1=REVD(INOD1)
D2=REVD(INOD2)
D3=REVD(INOD3)
D4=REVD(INOD4)
D5=REVD(INOD5)
D6=REVD(INOD6)
H1=REVH(INOD1)
H2=REVH(INOD2)
H3=REVH(INOD3)
H4=REVH(INOD4)
H5=REVH(INOD5)
H6=REVH(INOD6)

SPEED=REV(SISPEED)

C

ETP(I)=W1*H1+W2*H3+W3*H5-W1*H2-W2*H4-W3*H6
CALL TPIMB(I,ICUR1,ICUR2,DIA1,W1,SPEED,D1,D2,P1,P2,T1,T2,H1,
H2,UT1TP(I),T2TP(I))
CALL TPIMB(I,ICUR3,ICUR4,DIA2,W2,SPEED,D3,D4,P3,P4,T3,T4,H3,
H4,PITP(I),P2TP(I))
CALL TPIMB(I,ICUR5,ICUR6,DIA3,W3,SPEED,D5,D6,P5,P6,T5,T6,H5,
H6,P3TP(I),P4TP(I))

F=F+(ETP(I)/UETP(I))**2+
(T1TP(I)/UT1TP(I))**2+
(T2TP(I)/UT2TP(I))**2+
(P1TP(I)/UP1TP(I))**2+
(P2TP(I)/UP2TP(I))**2+
(P3TP(I)/UP3TP(I))**2+
(P4TP(I)/UP4TP(I))**2

ENDIF

C

170 CONTINUE

C

WH2=0.0
WO2=0.0

C

DO 180 IH2IN=1,NH2HG(IHGNZ)
IBRH2=IH2HG(IHGNZ,IH2IN)
WH2=WH2+REVM(IBRH2)
180 CONTINUE

C

DO 190 IO2IN=1,NO2HG(IHGNZ)
IBRH2=IO2HG(IHGNZ,IO2IN)
WO2=WO2+REVM(IBRH2)
190 CONTINUE

C

ATHROT=TTB(IATHRT)
OFCOM=TTB(IMIXR)
WTOTAL=WO2+WH2
OFCAL=WO2/WH2
OFIMB = OFCAL - OFCOM

\[ F = F + \left( \frac{OFIMB}{UOF} \right)^2 \]

GAMH2N = 1.3
GAMO2N = 1.3
GAMH2O = 1.3
P TOTAL = REVP (INOZN)
T TOTAL = REV (INOZN)
CEFF = TTB (ICEFF (IHNOZ))

\[ XF = 1.0 \div (1.0 + OFCAL) \]
\[ XO = 1.0 - XF \]
\[ XH2 = XF - XO \times 2.0 \times CEFF \times 2.016 / 31.9988 \]
\[ XH2O = XO \times 2.0 \times CEFF \times 18.0153 / 31.9988 \]
\[ XO2 = 1.0 - XH2 - XH2O \]
\[ \Gamma = XH2 \times \Gamma_{H2N} + XH2O \times \Gamma_{H2O} + XO2 \times \Gamma_{O2N} \]

\[ \text{GASCON} = \left( \frac{XH2}{2.016} + \frac{XH2O}{18.0153} + \frac{XO2}{31.9988} \right) \times 1545.3 \]
\[ T_{THROT} = T_{TOTAL} / (1 + (\Gamma / (\Gamma - 1)) / 2) \]
\[ V_{THROT} = \sqrt{\Gamma \times \text{GASCON} \times T_{THROT} \times GC} \]
\[ \rho = 144.0 \times V_{THROT} / (\text{GASCON} \times T_{THROT}) \]
\[ W_{NOZ} = \rho \times T_{THROT} \times \sqrt{\rho} \]
\[ W_{CAL} = \text{REVM} (INOZBR) \]
\[ W_{ZIMB} = W_{CAL} - W_{NOZ} \]

\[ F = F + \left( \frac{W_{ZIMB}}{UMVOL (INOZMC)} \right)^2 \]

IF (IFUNC.EQ.1) THEN
DO 210 I = 1, NMC
WIMBO (I) = WIMB (I)
CONTINUE
DO 220 I = 1, NEC
EIMBO (I) = EIMB (I)
CONTINUE
DO 230 I = 1, NPC
PIMBO (I) = PIMB (I)
CONTINUE
DO 240 I = 1, NTPC
IF (ITPTY (I).EQ.1) THEN
ETP0 (I) = ETP (I)
T1TP0 (I) = T1TP (I)
T2TP0 (I) = T2TP (I)
P1TP0 (I) = P1TP (I)
P2TP0 (I) = P2TP (I)
ELSE
ETP0 (I) = ETP (I)
T1TP0 (I) = T1TP (I)
T2TP0 (I) = T2TP (I)
P1TP0 (I) = P1TP (I)
P2TP0 (I) = P2TP (I)
P3TP0 (I) = P3TP (I)
P4TP0 (I) = P4TP (I)
ENDIF
CONTINUE
WZIMBO = WZIMB
OFIMBO = OFIMB
ELSE
ENDIF
RETURN
END

**********************************************************************
SUBROUTINE PROP (MAT, PRSI, TMP1, OF, CEFF, ZENTH, ZENTR, ZDENS)

PROP - PROPERTY PROGRAM CALCULATING HYDROGEN, OXYGEN, STEAM AND HOT GAS PROPERTIES

COMMON /H2PRP/
* H2P1(15), H2T1(11), H2H1(15, 11), H2S1(15, 11), H2D1(15, 11),
* H2P2(20), H2T2(11), H2H2(20, 11), H2S2(20, 11), H2D2(20, 11),
* H2P3(29), H2T3(25), H2H3(29, 25), H2S3(29, 25), H2D3(29, 25),
COMMON /O2PRP/
* O2P1(13), O2T1(16), O2H1(13, 16), O2S1(13, 16), O2D1(13, 16),
* O2P2(13), O2T2(17), O2H2(13, 17), O2S2(13, 17), O2D2(13, 17),
* O2P3(5), O2T3(56), O2H3(56), O2S3(56), O2D3(56)
* COMMON /H2OPRP/
* H2OP1(7), H2OT1(13), H2OH1(7,13), H2OS1(7,13), H2OD1(7,13)

COMMON /TABLE/
* NH2P(4), NH2T(4), NO2P(3), NO2T(3), NH2OP(1), NH2OT(1)

DIMENSION
* TSH2(11), PSH2(11), HLH2(11), HVH2(11), SLH2(11), SVH2(11),
* DLH2(11), DVLH2(11), TS02(16), PS02(16), HLO2(16), HVO2(16), SLO2(16), SVO2(16),
* DLO2(16), DV02(16)

TSH2 - H2 SATURATION TEMPERATURE
PSH2 - H2 SATURATION PRESSURE
HLH2 - H2 SATURATION ENTHALPY - LIQUID
HVH2 - H2 SATURATION ENTHALPY - VAPOR
SLH2 - H2 SATURATION ENTROPY - LIQUID
SVH2 - H2 SATURATION ENTROPY - VAPOR
DLH2 - H2 SATURATION DENSITY - LIQUID
DVH2 - H2 SATURATION DENSITY - VAPOR

DATA (TSH2(J), J=1,11)/
* 30.0, 32.0, 34.0, 36.0, 38.0, 40.0, 42.0, 44.0, 46.0, 48.0, 50.0/

DATA (PSH2(J), J=1,11)/
* 4.170, 6.446, 9.527, 13.561, 18.694, 25.089, 32.915, 42.334,
* 53.514, 66.625, 81.838/

DATA (HLH2(J), J=1,11)/
* -123.995, -120.090, -115.893, -111.380, -106.524, -101.289,
* -95.636, -89.513, -82.850, -75.556, -67.493/

DATA (HVH2(J), J=1,11)/
* 70.977, 74.584, 77.848, 80.729, 83.256, 85.199, 86.614, 87.431,
* 87.546, 86.817, 85.043/

DATA (SLH2(J), J=1,11)/
* 1.506, 1.629, 1.752, 1.876, 2.002, 2.129, 2.259, 2.391, 2.528,
* 2.670, 2.819/

DATA (SVH2(J), J=1,11)/
* 8.005, 7.713, 7.451, 7.214, 6.998, 6.794, 6.601, 6.415, 6.234,
* 6.054, 5.871/

DATA (DLH2(J), J=1,11)/
* 4.6500, 4.5832, 4.5127, 4.4378, 4.3580, 4.2724, 4.1801, 4.0798,
* 3.9698, 3.8479, 3.7108/

DATA (DVH2(J), J=1,11)/
* 0.0272, 0.0401, 0.0568, 0.0779, 0.1039, 0.1363, 0.1757, 0.2234,
* 0.2809, 0.3508, 0.4362/

TS02 - O2 SATURATION TEMPERATURE
PS02 - O2 SATURATION PRESSURE
HL02 - O2 SATURATION ENTHALPY - LIQUID
HV02 - O2 SATURATION ENTHALPY - VAPOR
SLO2 - O2 SATURATION ENTROPY - LIQUID
SVO2 - O2 SATURATION ENTROPY - VAPOR
DLO2 - O2 SATURATION DENSITY - LIQUID
DV02 - O2 SATURATION DENSITY - VAPOR

DATA (TS02(J), J=1,16)/
* 160.0, 164.0, 168.0, 172.0, 176.0, 180.0, 184.0, 188.0, 192.0,
* 196.0, 200.0, 204.0, 208.0, 212.0, 216.0, 220.0/

DATA (PS02(J), J=1,16)/
* 12.810, 16.183, 20.200, 24.935, 30.467, 36.876, 44.243, 52.654,
* 62.194, 72.951, 85.013, 98.473, 113.421, 129.952, 148.162,
* 168.146/

DATA (HL02(J), J=1,16)/
* -46.790, -45.093, -43.380, -41.650, -39.901, -38.130, -36.334,
* -34.511, -32.657/

DATA (HV02(J), J=1,16)/
* 33.777, 34.457, 35.110, 35.734, 36.326, 36.884, 37.408, 37.894,

C

DATA (SLO2(J), J=1,16)/
* 0.698, 0.708, 0.717, 0.727, 0.736, 0.746, 0.755, 0.764, 0.772,
* 0.781, 0.790, 0.798, 0.806, 0.815, 0.823, 0.831/

C

DATA (SVO2(J), J=1,16)/
* 1.273, 1.263, 1.254, 1.245, 1.237, 1.229, 1.221, 1.214, 1.207,
* 1.200, 1.193, 1.187, 1.180, 1.174, 1.168, 1.162/

C

DATA (DLO2(J), J=1,16)/
* 71.630, 70.941, 70.243, 69.536, 68.818, 68.089, 67.347, 66.593,
* 65.823, 65.037, 64.234, 63.412, 62.567, 61.699, 60.804, 59.880/

C

DATA (DVO2(J), J=1,16)/
* 0.246, 0.305, 0.374, 0.455, 0.547, 0.653, 0.774, 0.911, 1.065,
* 1.239, 1.433, 1.650, 1.893, 2.162, 2.461, 2.794/

C

C

* 0.698, 0.708, 0.717, 0.727, 0.736, 0.746, 0.755, 0.764, 0.772,
* 0.781, 0.790, 0.798, 0.806, 0.815, 0.823, 0.831/

C

DATA (SLO2(J), J=1,16)/
* 1.273, 1.263, 1.254, 1.245, 1.237, 1.229, 1.221, 1.214, 1.207,
* 1.200, 1.193, 1.187, 1.180, 1.174, 1.168, 1.162/

C

DATA (SVO2(J), J=1,16)/
* 71.630, 70.941, 70.243, 69.536, 68.818, 68.089, 67.347, 66.593,
* 65.823, 65.037, 64.234, 63.412, 62.567, 61.699, 60.804, 59.880/

C

DATA (DLO2(J), J=1,16)/
* 0.246, 0.305, 0.374, 0.455, 0.547, 0.653, 0.774, 0.911, 1.065,
* 1.239, 1.433, 1.650, 1.893, 2.162, 2.461, 2.794/

C

C

51 FORMAT(/3X,'PROP - REQUESTED PRS > ',F7.2,2X,
* 'AND TMP > ',F7.2,2X,'FOR H2 IS OUT OF RANGE')
52 FORMAT(/3X,'PROP - REQUESTED PRS > ',F7.2,2X,
* 'AND TMP > ',F7.2,2X,'FOR O2 IS OUT OF RANGE')
53 FORMAT(/3X,'PROP - REQUESTED PRS > ',F7.2,2X,
* 'AND TMP > ',F7.2,2X,'FOR STEAM IS OUT OF RANGE')

** INTERPOLATE RESULTS FROM SINGLE ARRAY **

IPRP=0
NPXI=2
NPIY=2
ZENTH=0.0
ZENTR=0.0
ZDENS=0.0

GO TO (10,20,30,40) MAT

10 IF(TMPL.GT. 30.0.AND.TMPI.LT. 50.0) IPRP=1
IF(TMPL.GT. 70.0.AND.TMPI.LT. 110.0) IPRP=2
IF(TMPL.GT. 240.0.AND.TMPI.LT. 720.0) IPRP=3
IF(TMPL.GT.1400.0.AND.TMPI.LT.2000.0) IPRP=4
GO TO (11,12,13,14) IPRP

11 IF(PRSL.LT. 20.0.OR.PRSL.GT. 370.0) GO TO 50
CALL PRPSAT(PRSL,TPL, ZENTH,
* TSH2(11), NH2P(1), NH2T(1), 11, 29.95, 50.05,
* H2P1, H2T1, H2H1, PSH2, TSH2, H2H2, H2H2)
CALL PRPSAT(PRSL,TPL, ZENTR,
* TSH2(11), NH2P(1), NH2T(1), 11, 29.95, 50.05,
* H2P1, H2T1, H2S1, PSH2, TSH2, SLH2, SVH2)
CALL PRPSAT(PRSL,TPL, ZDENS,
* TSH2(11), NH2P(1), NH2T(1), 11, 29.95, 50.05,
* H2P1, H2T1, H2D1, PSH2, TSH2, DLH2, DVH2)
RETURN

12 IF(PRSL.LT.3400.0.OR.PRSL.GT.7200.0) GO TO 50
CALL ITERP2(PRSL,TPL,H2P1,H2T2,H2H2,
* NH2P(2), NH2T(2), NPX1, NPIY, NH2P(2), ZENTH,N1)
CALL ITERP2(PRSL,TPL,H2P2,H2T2,H2H2,
* NH2P(2), NH2T(2), NPX1, NPIY, NH2P(2), ZENTR,N1)
CALL ITERP2(PRSL,TPL,H2P3,H2T2,H2H2,
* NH2P(3), NH2T(3), NPX1, NPIY, NH2P(3), ZDENS,N1)
RETURN

13 IF(PRSL.LT.1400.0.OR.PRSL.GT.7000.0) GO TO 50
CALL ITERP2(PRSL,TPL,H2P3,H2T3,H2H3,
* NH2P(3), NH2T(3), NPX1, NPIY, NH2P(3), ZENTH,N1)
CALL ITERP2(PRSL,TPL,H2P4,H2T4,H2H4,
* NH2P(4), NH2T(4), NPX1, NPIY, NH2P(4), ZENTR,N1)
RETURN

14 IF(PRSL.LT.1400.0.OR.PRSL.GT.5800.0) GO TO 50
CALL ITERP2(PRSL,TPL,H2P4,H2T4,H2H4,
* NH2P(4), NH2T(4), NPX1, NPIY, NH2P(4), ZENTR,N1)
CALL ITERP2(PRSI, TMPI, H2P4, H2T4, H2D4,  
* NH2P(4), NH2T(4), NPX1, NPY1, NH2P(4), ZDENS, N1)  
RETURN

20 IF(TMPI.GT.160.0.AND.TMPI.LT.240.0) IPRP=1  
IF(TMPI.LT.1.0.AND.PRSI.GT.650.0) IPRP=1  
IF(IPRP.EQ.1.AND.PRSI.GT.650.0) IPRP=2  
IF(TMPI.GT.600.0.AND.TMPI.LT.1500.0) IPRP=3  
GO TO (21,22,23) IPRP

21 IF(PRSI.LT.30.0.OR.PRSI.GT.630.0) GO TO 50  
IF(TMPI.LT.1.0.AND.TMPI.GT.219.9) GO TO 50  
CALL PRPSAT(PRSI, TMPI, ZENTH,  
* TSO2(16), NO2P(1), NO2T(1), 16, 159.95, 220.05,  
* O2P1, O2T1, O2H1, PSO2, TSO2, HLO2, HVO2)  
CALL PRPSAT(PRSI, TMPI, ZENTR,  
* TSO2(16), NO2P(1), NO2T(1), 16, 159.95, 220.05,  
* O2P1, O2T1, O2S1, PSO2, TSO2, SLO2, SVO2)  
CALL PRPSAT(PRSI, TMPI, ZDENS,  
* TSO2(16), NO2P(1), NO2T(1), 16, 159.95, 220.05,  
* O2P1, O2T1, O2D1, PSO2, TSO2, DLO2, DVO2)  
RETURN

22 IF(PRSI.LT.2000.0.OR.PRSI.GT.8000.0) GO TO 50  
CALL ITERP2(PRSI, TMPI, O2P2, O2T2, O2H2,  
* NO2P(2), NO2T(2), NPX1, NPY1, NO2P(2), ZENTH, N1)  
CALL ITERP2(PRSI, TMPI, O2P2, O2T2, O2S2,  
* NO2P(2), NO2T(2), NPX1, NPY1, NO2P(2), ZENTR, N1)  
CALL ITERP2(PRSI, TMPI, O2P2, O2T2, O2D2,  
* NO2P(2), NO2T(2), NPX1, NPY1, NO2P(2), ZDENS, N1)  
RETURN

23 IF(PRSI.LT.2000.0.OR.PRSI.GT.4000.0) GO TO 50  
CALL ITERP2(PRSI, TMPI, O2P3, O2T3, O2H3,  
* NO2P(3), NO2T(3), NPX1, NPY1, NO2P(3), ZENTH, N1)  
CALL ITERP2(PRSI, TMPI, O2P3, O2T3, O2S3,  
* NO2P(3), NO2T(3), NPX1, NPY1, NO2P(3), ZENTR, N1)  
CALL ITERP2(PRSI, TMPI, O2P3, O2T3, O2D3,  
* NO2P(3), NO2T(3), NPX1, NPY1, NO2P(3), ZDENS, N1)  
RETURN

30 IF(TMPI.LT.1400.0.OR.TMPI.GT.2200.0) GO TO 50  
IF(PRSI.LT.100.0.OR.PRSI.GT.700.0) GO TO 50  
CALL ITERP2(PRSI, TMPI, H2OPI, H2OTI, H2OHI,  
* NH2OP(1), NH2OT(1), NPX1, NPY1, NH2OP(1), ZENTH, N1)  
CALL ITERP2(PRSI, TMPI, H2OPI, H2OTI, H2OS1,  
* NH2OP(1), NH2OT(1), NPX1, NPY1, NH2OP(1), ZENTR, N1)  
CALL ITERP2(PRSI, TMPI, H2OPI, H2OTI, H2OD1,  
* NH2OP(1), NH2OT(1), NPX1, NPY1, NH2OP(1), ZDENS, N1)  
RETURN

40 CALL PRPMIX(PRSI, TMPI, OF, CEFF, HMIX, SMIX)  
** MODIFIED 2/19/93 **
40 CALL PRPMIX(PRSI, TMPI, OF, CEFF, HMIX, SMIX, DMIX)  
ZENTH=HMIX  
ZENTR=SMIX  
ZDENS=0.0  
** MODIFIED 2/19/93 **  
ZDENS=DMIX  
RETURN

50 IF(MAT.EQ.1) WRITE(21,51) PRSI, TMPI  
IF(MAT.EQ.2) WRITE(21,52) PRSI, TMPI  
IF(MAT.EQ.3) WRITE(21,53) PRSI, TMPI  
RETURN

END

** PRPMIX - CALCULATES HOT GAS MIXTURE PROPERTIES. **
```plaintext
COMMON /H2PRP/
* H2P1(15), H2T1(11), H2H1(15,11), H2S1(15,11), H2D1(15,11),
* H2P2(20), H2T2(11), H2H2(20,11), H2S2(20,11), H2D2(20,11),
* H2P3(29), H2T3(25), H2H3(29,25), H2S3(29,25), H2D3(29,25),
COMMON /O2PRP/
* O2P1(13), O2T1(16), O2H1(13,16), O2S1(13,16), O2D1(13,16),
* O2P2(13), O2T2(17), O2H2(13,17), O2S2(13,17), O2D2(13,17),
* O2P3(5), O2T3(61), O2H3(5,61), O2S3(5,61), O2D3(5,61)
COMMON /H2OPRP/
* H2O1(7), H2OT1(13), H2OH1(7,13), H2OS1(7,13), H2OD1(7,13)

COMMON /TABLE/
* NH2P(4), NH2T(4), NO2P(3), NO2T(3), NH2OP(1), NH2OT(1)
COMMON /STD/
* HH2REF, HO2REF, HWAREF, SH2REF, SO2REF, SWAREF, SH2A, SO2A,
* SWAA

XMWH2 = 2.0160
XMWO2 = 31.9988
XMWH2O = 18.0153
HCOMB = -6825.6550

NPIX1 = 2
NPy1 = 2
ITST1 = 0
ITST2 = 0
ITST3 = 0
ITST4 = 0
ITST5 = 0
ITST6 = 0

XF = 1.0 / (1.0 + OF)
XO = 1.0 - XF
XH2 = XF - XO * 2.0 * CEFF * XMWH2 / XMWO2
XH2O = XO * 2.0 * CEFF * XMWH2O / XMWO2
XO2 = 1.0 - XH2 - XH2O

EH2 = XH2 / XMWH2
EH2O = XH2O / XMWH2O
EO2 = XO2 / XMWO2
ET = EH2 + EH2O + EO2

YH2 = EH2 / ET
YH2O = EH2O / ET
YO2 = 1.0 - YH2 - YH2O

PH2 = P * YH2
PH2O = P * YH2O
PO2 = P * YO2

IF (TMP1.LT.1000.0.OR.TMPI.GT.2000.0) ITST1=1
IF (PH2 .LT.1400.0.OR.PH2 .GT.5800.0) ITST2=1
CALL ITERP2(PH2, TMP1, H2P4, H2T4, H2H4, H2S4, H2D4, NI)
* NH2P(4), NH2T(4), NPIX1, NPy1, NH2P(4), HH2, N1
CALL ITERP2(PH2, TMP1, H2P4, H2T4, H2H4, H2S4, H2D4, NI)
* NH2P(4), NH2T(4), NPIX1, NPy1, NH2P(4), HH2, N1

IF (TMP1.LT.1000.0.OR.TMPI.GT.2000.0) ITST3=1
IF (PH20.LT.1000.0.OR.PH20.GT.7000.0) ITST4=1
CALL ITERP2(PH20, TMP1, H2P4, H2T4, H2H4, H2S4, H2D4, NI)
* NH2P(4), NH2T(4), NPIX1, NPy1, NH2P(4), HH2, N1
CALL ITERP2(PH20, TMP1, H2P4, H2T4, H2H4, H2S4, H2D4, NI)
* NH2P(4), NH2T(4), NPIX1, NPy1, NH2P(4), HH2, N1

IF (YO2.LT.0.001) THEN
DHO2 = 0.0
DSO2 = 0.0
ELSE
IF (TMP1.GT. 600.0.AND.TMPI.LT.1500.0) ITST5=1
IF (PO2 .LT.2000.0.OR.PO2 .GT.4000.0) ITST6=1
CALL ITERP2(PO2, TMP1, O2P3, O2T3, O2H3, O2S3, O2D3, NI)
* NO2P(3), NO2T(3), NPIX1, NPy1, NO2P(3), HO2, N1
CALL ITERP2(PO2, TMP1, O2P3, O2T3, O2H3, O2S3, O2D3, NI)
* NO2P(3), NO2T(3), NPIX1, NPy1, NO2P(3), SO2, N1
DHO2 = HO2 - H2O2REF
DSO2 = SO2 - SO2REF + SO2A
```
C ENDIF

10 DH2 = HH2 - HH2REF
DHH2M = (HH2O - HWAREF) + HCOMB
DSH2 = SH2 - SH2REF + SH2A
DSH2O = SH2O - SWAREF + SWAA

C HMX = XH2*DHH2 + XH20*DHH20M + XO2*DHO2
SMIX = XH2*DSH2 + XH2O*DSH2O + XO2*DSO2

C ************ MODIFIED 2/19/93 ************
XMMIX = XMWH20*YH20 + XMWH2*YH2 + XMWO2*YO2
DMIX = 144.*XMMIX/P/(1545.3*TMPI)

C IF {ITST1 EQ.I OR. ITST2 EQ.I) WRITE(21,511 PH2, TMP1)
C IF {ITST3 EQ.I OR. ITST4 EQ.I) WRITE(21,52 PH20, TMP1)
C IF {ITST5 EQ.I OR. ITST6 EQ.I) WRITE(21,53 PO2, TMP1)

51 FORMAT(/3X, 'PRPMIX - REQUESTED PH2 PRS > ',F7.2,2X, * AND TMP > ',F7.2,2X,'FOR " H2" IS OUT OF RANGE')
52 FORMAT(/3X, 'PRPMIX - REQUESTED PH20 PRS > ',F7.2,2X, * AND TMP > ',F7.2,2X,'FOR " H20" IS OUT OF RANGE')
53 FORMAT(/3X, 'PRPMIX - REQUESTED PO2 PRS > ',F7.2,2X,'FOR " O2" IS OUT OF RANGE')

C RETURN
END

SUBROUTINE PRPSAT (X,Y,FPROP,TCRT,NXI,NYI,NX2,YL,YH, * PRSI,TMPI,PROP,PRSI,TMP2,PROPL,PROPV)

C PRPSAT - CALCULATES NBS PROPERTIES NEAR SATURATION CURVE
C
DIMENSION PRSI(1),TMP1(1)

NR1=NX1
NPX1=2
NPY1=2
NPX2=2

ZPLGAS=0.0
ZPHGAS=0.0
ZPLLIQ=0.0
ZPHLIQ=0.0
ZPROPI=0.0
ZPROP=0.0
ZTSAT=0.0
ARGA=0.0
ARGB=0.0
ZTSAT=0.0

CALL ITERP2(X,Y,PRSI,TMPI,PROP,NXI,NYI,NPX1,NPY1,NRI,ZPROPI,NI)
FPROP=ZPROPI
IF(Y.GT.TCRT) GO TO 70
CALL ITERPI(X,PRSI,TMP2,NX2,NPX2,ZTSAT,N2)
IF(Y.LT.ZTSAT) GO TO 61

* * GAS CALCULATIONS * *
CALL ITERPI(X,PRSI,PROP,NX1,NPY1,NRI,ZTSAT,NI)
DTST=ZTSAT-ZPGAS
IF(DTST.GT.0.0001) GO TO 50
ZPGAS=ZTSAT-ZPGAS
IF(ZPROPI.LT.ZPGAS) GO TO 70
GO TO 51
50 ZPGAS=ZPGAS
IF(ZPROPI.GT.ZPGAS) GO TO 70
GO TO 51
51 LPR=1
52 PRSD=PRSI(LPR)-0.0001
IF(PRSD.GT.X) GO TO 52
LPR=LPR+1
GO TO 53
52 ARGA=PRSI(LPR)
CALL ITERP1(ARGA,PRS2,TMP2,NX2,NPX2,ZTSATT,N2)

LTP=1
54 TMPD=TMP1(LTP)-0.0001
IF(TMPD.GT.ZTSATT) GO TO 55
LTP=LTP+1
GO TO 54

55 ARG=TMP1(LTP)
YY=ARG
IF(DTST.GT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,* NPX1,NPY1,NR1,ZPLGAS,N1)
IF(DTST.LT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,* NPX1,NPY1,NR1,ZPHGAS,N1)
ZPROP=ZPHGAS-(ZPHGAS-ZPLGAS)*((ARG-Y)/(ARG-ZTSAT))
FPROP=ZPROP
GO TO 70

* * LIQ CALCULATIONS * *
61 CALL ITERP1(X,PRS2,PROP,L,NX2,NPX2,ZPLIQ,N2)
CALL ITERP2(X,YL,PRS1,TMP1,PROP,NX1,NY1,NPX1,NPY1,NR1,ZTST,N1)
DTST=ZTST-ZPLIQ
IF(DTST.GT.0.0001) GO TO 59
ZPLIQ=ZPLIQ
IF(ZPROP.LT.ZPLIQ) GO TO 70
GO TO 60
59 ZPHLIQ=ZPLIQ
IF(ZPROP.GT.ZPLIQ) GO TO 70
GO TO 60

62 ARG=PRS1(LPR-1)
CALL ITERP1(ARGA,PRS2,TMP2,NX2,NPX2,ZTSATT,N2)

LTP=1
64 TMPD=TMP1(LTP)-0.0001
IF(TMPD.GT.ZTSATT) GO TO 65
LTP=LTP+1
GO TO 64

65 ARG=TMP1(LTP-1)
YY=ARG
IF(DTST.GT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,* NPX1,NPY1,NR1,ZPHLIQ,N1)
IF(DTST.LT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,* NPX1,NPY1,NR1,ZPHLIQ,N1)
ZPROP=ZPHLIQ-(ZPHLIQ-ZPLLIQ)*((ZTSAT-Y)/(ZTSAT-ARG))
FPROP=ZPROP
GO TO 70

70 CONTINUE

RETURN
END

********************************************************************
SUBROUTINE ITERP1 (X,XT,YT,NX,NPX,Y,NERR)
********************************************************************
DIMENSION XT(1),YT(1)
NERR=0
INTER=1
NP=NPX
IF(NX.LT.NP) NP=NX
IH=NP/2
I=1
IF(XT(I)-X)30,20,10
10 IH=0
12 NERR=1
GO TO 70
13 NERR=2
GO TO 70
20 INTER=2
Y = YT(I)
GO TO 999
I = NX
IF (XT(I) - X) 13, 20, 40
N1 = I
N2 = NX
MP = (N1 + N2) / 2
IF (XT(MP) - X) 52, 54, 56
N1 = MP
GO TO 60
I = MP
GO TO 20
N2 = MP
IF ((N2 - N1) .NE. 1) GO TO 45
IF (N2 .GT. (IH + I)) GO TO 65
I = IH + I
GO TO 70
I = N2
IF (N2 .GT. I) I = N2
K = 1 - IH
N = K + NP - 1
Y = 0.
IF (N - NX) 90, 90, 80
N = NX
K = NX - NP + 1
DO 120 J = K, NP = I, 0
DO 110 I = K, N
IF (I - J) 100, II0 (XT(J) - XT(I))
100 CONTINUE
Y = Y + YT(J) * P
110 CONTINUE
Y = Y + YT(J) * P
120 CONTINUE
ENTRY ENTERP (X, XT, YT, Y)
GO TO 999
N = K + NPYY - I
IF (N - NY) 90, 90, 80
GO TO 45
I = IH + I
SUBROUTINE OPT (N,NMAX,EPSC,EPST,RHO,ALPHA,BETA,ISTAGE,F2,GNORM,DXNORM,X,X2)
DIMENSION X(25),X2(25),G(25),G2(25),DX(25),H(25,25)

ISTAGE=STAGE COUNTER
ISTAGE=1
IFUNC=0
CALL OBJF (IFUNC,X,F)
CALL GRAD (IFUNC,N,X,G,F)

SET INITIAL METRIC H TO THE IDENTITY MATRIX
DO 12 I=1,N
DO 12 J=1,N
IF (J .NE. 1) THEN
   H(I,J)=0.0
ELSE
   H(I,J)=1.0
ENDIF
12 CONTINUE

KC=STEP DIRECTION PARAMETER
KC=1 NEGATIVE GRADIENT STEP DIRECTION
KC=0 QUASI-NEWTON STEP DIRECTION

KC=1
CALL UNIVARIATE SEARCH ROUTINE
CALL ARMijo (IFUNC,IFLAG,N,RHO,ALPHA,BETA,DMAx,
   DX,X,X2,F2,G)
CALL GRAD (IFUNC,N,X2,G2,F2)

DO 50 I=1,N
DOXDG=DXDG+(X2(I)-X(I))*(G2(I)-G(I))
IF (DXDG .LT. 0.0 .AND. KC .EQ. 0) IFLAG=1
IF (IFLAG .NE. 1) GO TO 90
76 IF (KC .EQ. 1 .AND. IFLAG .EQ. 1) GO TO 200
GO TO 10

CALL UNIVARIATE SEARCH ROUTINE
CALL ARMijo (IFUNC,IFLAG,N,RHO,ALPHA,BETA,DMAx,
   DX,X,X2,F2,G)
CALL GRAD (IFUNC,N,X2,G2,F2)

DO 50 I=1,N
DOXDG=DXDG+(X2(I)-X(I))*(G2(I)-G(I))
IF (DXDG .LT. 0.0 .AND. KC .EQ. 0) IFLAG=1
IF (IFLAG .NE. 1) GO TO 90
76 IF (KC .EQ. 1 .AND. IFLAG .EQ. 1) GO TO 200
GO TO 10

GRADIENT NORM**2 < EPSC IMPLIES CONVERGENCE
GNORM=0.0
DO 92 I=1,N
GNORM=GNORM+G2(I)**2
IF (GNORM-EPSC) 200,110,110

DELTA X NORM**2 < EPST IMPLIES TERMINATION
DXNORM=0.0
DO 112 I=1,N
DXNORM=DXNORM+(X2(I)-X(I))**2
IF (DXNORM-EPST) 200,120,120

INCREMENT STAGE COUNTER
ISTAGE=ISTAGE+1
CHECK THAT MAXIMUM NUMBER OF STAGES NOT EXCEEDED
IF (ISTAGE .GT. NMAX) GO TO 200

UPDATE METRIC H
CALL BFGS (IFLAG,N,X,X2,G,G2,H)
IF (IFLAG.EQ.1) GO TO 76
KC=0

REINITIALIZE
F=F2
DO 125 I=1,N
X(I)=X2(I)
G(I)=G2(I)
125 CONTINUE
GO TO 15

RETURN
END

SUBROUTINE ARMijo (IFUNC,IFLAG,N,RHO,ALPHA,BETA,DMAX,DX,X,X2,F,F2,G)
DIMENSION X(25),X2(25),DX(25),G(25)
DMAX=0.05
IFLAG=0
ICOUNT=1
GDX=0.0

DO 110 I=1,N
GDX=GDX+G(I)*DX(I)
110 CONTINUE
RATIO=0.0
DO 111 I=1,N
RATIO2=ABS(DX(I)/X(I))
111 CONTINUE
RATIO=MAX(RATIO,RATIO2)
RMU=RHO
SCALE=RMU*RATIO
IF (SCALE .GT. DMAX) RMU=DMAX/RATIO

DO 112 I=1,N
X2(I)=X(I)+RMU*DX(I)
112 CONTINUE
CALL OBJF (IFUNC,X2,F2)
TBAR=F2-F-RMU*ALPHA*GDX
IF (TBAR .GT. 0.0) GO TO 120
IF (F-F2 .GT. 180,180,200)
RMU=RMU*BETA
ICOUNT=ICOUNT+1
IF (ICOUNT .LE. 12) GO TO 112
180 CONTINUE
IFLAG=1
RETURN
END

SUBROUTINE BFGS (IFLAG,N,X,X2,G,G2,H)
DIMENSION X(25),X2(25),G(25)
V(25),H(25,25)
IFLAG=0

CALL DFP METRIC UPDATE
CALL DFP (N,DXDG,DGHDG,X,X2,G,G2,DX,DG,HDG,H)
IF (DGHDG .GT. 0.0) GO TO 10
IFLAG=1
GO TO 400

DETERMINE BFGS METRIC UPDATE
10 DO 310 I=1,N
310 V(I)=DGHDG**0.5*(DX(I)/DXDG-HD(I)/DGHDG)
DO 320 I=1,N
DO 320 J=1,N
320 H(I,J)=V(I)*V(J)+H(I,J)

RETURN
END
SUBROUTINE DFP (N, DXDG, DGHDG, X, X2, G, G2, DX, DG, HDG, H)
DIMENSION X(25), X2(25), G(25), G2(25), DX(25), DG(25), HDG(25), H(25, 25)
DO 100 I = 1, N
   DX(I) = X2(I) - X(I)
   DG(I) = G2(I) - G(I)
   DXDG = 0.0
   DGHDG = 0.0
DO 410 I = 1, N
   HDG(I) = 0.0
   DGH(I) = 0.0
DO 400 J = 1, N
   HDG(I) = HDG(I) + H(I, J) * DG(J)
   DGH(I) = DGH(I) + DG(J) * H(J, I)
   DXDG = DXDG + DX(I) * DG(I)
   DGHDG = DGHDG + DGH(I) * DG(I)
410 DO 420 I = 1, N
420 H(I, J) = H(I, J) + DX(I) * DX(J) / DXDG - HDG(I) * DGH(J) / DGHDG
RETURN
END

SUBROUTINE GRAD (IFUNC, N, X, G, F)
DIMENSION X(25), X2(25), G(25)
DX = 0.001
DO 10 I = 1, N
   X2(I) = X(I)
DO 20 I = 1, N
   X2(I) = X(I) + DX
   CALL OBJF (IFUNC, X2, F2)
   G(I) = (F2 - F) / DX
X2(I) = X(I)
RETURN
END

SUBROUTINE TPIMB (I, ICURA, ICURB, DIA, W, SPEED, D1, D2, P1, P2, T1, T2, H1, H2, ZIMB1, ZIMB2)

***** TURBINE CURVES REQUIRE ICURA & ICURB < 20 ***********************
IF (ICURA.LT.20) THEN
   FLOWC = W * SQRT(T1) / P1
   TMACH = SPEED / SQRT(T1)
   CALL CURVES (ICURA, FLOWC, TMACH, CHR3, CHR4, PRATIO)
   CALL CURVES (ICURB, FLOWC, TMACH, CHR3, CHR4, TRATIO)
   DPCHR = P1 - P2
   DPCAL = P1 - P2
   ZIMB1 = DPCAL - DPCHR
   DTCHR = T1 - (1.0 - TRATIO) * T1
   DTCAL = T1 - T2
   ZIMB2 = DTCAL - DTCHR
ELSE
***** PUMP CURVES REQUIRE ICURA & ICURB >= 20 ***********************
FLOWC = W / (D1 * SPEED * DIA ** 3)
CALL CURVES (ICURA, FLOWC, CHR2, CHR3, CHR4, HEADC)
CALL CURVES (ICURB, FLOWC, CHR2, CHR3, CHR4, EFFCHR)
DPCHR = HEADC * D1 * SPEED ** 2 * DIA ** 2
DPCAL = P2 - P1
ZIMB1 = DPCAL - DPCHR
PWRI = W * (H2 - H1)
PWRID = (144.0 / 778.16) * (W * (P2 - P1) / D1)
EFFCAL = PWRI * PWR
ZIMB2 = (EFFCAL - EFFCHR) * PWR
ENDIF
10 CONTINUE
RETURN
END