NASA-CR-195253

63 P

# INTEGRATED MODEL DEVELOPMENT FOR LIQUID FUELED ROCKET PROPULSION SYSTEMS

(NASA-CR-195253) INTEGRATED MODEL DEVELOPMENT FOR LIQUID FUELED ROCKET PROPULSION SYSTEMS Final Report (Christian Brothers Coll.) 63 p

N94-27166

Unclas

G3[redacted]/20 0209782

FINAL REPORT

by

L. Michael Santi

Mechanical Engineering Department
Christian Brothers University
650 East Parkway South
Memphis, TN 38104

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

June, 1993

## TABLE OF CONTENTS

## 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

Engine Network Components (continued)

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 (lbm/s)]
2. Pressures at system nodes [P (psia)]
3. Temperatures at system nodes [T (R)]
4. Controllable valve resistances [R ($s^2/in^2$-$ft^3$)]
5. Turbopump shaft speeds [N (rpm)]
6. Heat transfer rates [Q (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

$$\sum_{i} m_{ij} = (\text{mass flow residual})_j \qquad (1)$$

i=1,2,...number of I/O's in mass flow circuit j
j=mass flow circuit number

$$\sum_{i} m_{ij}\, h_{ij} - \sum_{k} m_{kj}\, h_{kj} + Q_j = (\text{energy flow residual})_j \qquad (2)$$

i=1,2,...number of inputs to energy circuit j
k=1,2,...number of outlets from energy circuit j
j=energy flow circuit number

Governing Equations in Residual Form (continued)

$$(P_{in} - P_{out} - R * W^2 / \rho)_j = \text{(pressure drop residual)}_j \quad (3)$$

j=1,2,...number of pressure circuits

$$\{ \rho * N^2 * D^2 * [ C_H - C_H(C_Q) ] \}_j = \text{(pump } \Delta P \text{ residual)}_j \quad (4)$$

j=1,2,...number of pumps

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

j=1,2,...number of pumps

$$[(P_{in}-P_{out}) - (P_{in}-P_{out})_{characteristic}]_j = \text{(turbine } \Delta P \text{ residual)}_j \quad (6)$$

j=1,2,...number of turbines

$(P_{in}-P_{out})_{characteristic} = P_{in} * f(C_Q, Ma, \gamma)$

$$[(T_{in}-T_{out}) - (T_{in}-T_{out})_{characteristic}]_j = \text{(turbine } \Delta T \text{ residual)}_j \quad (7)$$

j=1,2,...number of turbines

$(P_{in}-P_{out})_{characteristic} = P_{in} * f(C_Q, Ma, \gamma)$

$$m_{oxygen}/m_{fuel} - (m_{oxygen}/m_{fuel})_{command} = \text{(mixture ratio residual)} \quad (8)$$

$$\sum_i (m_i) - m_{nozzle} = \text{(nozzle flow residual)} \quad (9)$$

i=1,2,...number of nozzle inlet flows

$m_{nozzle} = f(P_{ns}, T_{ch}, \text{nozzle geometry})$

$$( Q - \Delta T_{driving}/R_{thermal} )_j = \text{(heat transfer residual)}_j \quad (10)$$

j=1,2,...number of unknown heat transfer rates

where

m=mass flow rate
h=specific enthalpy
P=total pressure
Q=heat transfer rate
R=flow resistance

T=temperature
W=weight flow rate
$C_Q$=flow coefficient
$C_H$=head coefficient
Ma=turbine Mach number
$P_{ns}$=nozzle stagnation pressure
$T_{ch}$=main combustion chamber temperature
$\gamma$=specific heat ratio
$\eta$=turbine efficiency
$\rho$=mass density
f=appropriate performance function

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 $R^n$ [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:

$$\text{Minimize} \quad F = \sum_i \left[ \frac{\text{residual } i}{\text{uncertainty of residual } i} \right]^2 \qquad (11)$$

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)

6. Append turbopump performance curves.

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

1. Turton, R. K., Principles of Turbomachinery. E. & F. N. Spon, London, 1984.

2. Dixon, S. L., Fluid Mechanics, Thermodynamics of Turbomachinery. Pergamon Press, Oxford, 1966.

3. Rheinboldt, W. C., Methods for Solving Systems of Nonlinear Equations. CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM/Arrowsmith Ltd., Philadelphia, 1974.

4. Fletcher, R., "A New Approach to Variable Metric Algorithms," Comput. J., Vol. 13, 1970, pp. 317-322.

5. Armijo, L., "Minimization of Functions Having Lipschitz-Continuous First Partial Derivatives," Pacific J. Math., Vol. 16, 1966, pp. 1-3.

**APPENDICES**

# APPENDIX A

# TABLES

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

| UNCERTAINTY ESTIMATE | CASE A | CASE B | CASE C |
|---|---|---|---|
| Preburner power balance (Btu/s) | 100 | 100 | 50 |
| Preburner flow (lbm/s) | 0.1 | 0.01 | 0.01 |
| $O_2$ resistance pressure drop (psi) | 10 | 10 | 10 |
| $H_2$ resistance pressure drop (psi) | 10 | 10 | 10 |
| Pump flow head gain (psi) | 10 | 10 | 50 |
| Pump power requirement (hp) | 100 | 100 | 50 |
| Turbine flow head drop (psi) | 10 | 10 | 50 |
| Turbine flow temp drop (deg R) | 1 | 1 | 10 |
| Turbopump power (Btu/s) | 100 | 100 | 50 |
| Preburner $O_2/H_2$ ratio | 0.001 | 0.001 | 0.001 |

Table A2. Summary of independent variable predictions for high pressure fuel turbopump subsystem analyses

| VARIABLE | INITIAL | CASE A FINAL | CASE B FINAL | CASE C FINAL |
|---|---|---|---|---|
| M1 (lbm/s) | 79.63 | 67.99 | 74.68 | 68.81 |
| M2 | 84.08 | 72.27 | 78.12 | 72.69 |
| M3 | 163.72 | 140.51 | 152.80 | 141.51 |
| M4 | 162.25 | 148.53 | 152.16 | 162.70 |
| P1 (psia) | 7733.1 | (same) | (same) | (same) |
| P2 | 6088.0 | (same) | (same) | (same) |
| P3 | 5516.6 | 5790.5 | 5515.2 | 5847.9 |
| P4 | 3717.7 | 3931.5 | 3717.7 | 3873.4 |
| P5 | 271.7 | (same) | (same) | (same) |
| P6 | 6738.7 | 6405.5 | 6725.6 | 6096.9 |
| T1 (deg R) | 208.3 | (same) | (same) | (same) |
| T2 | 278.4 | (same) | (same) | (same) |
| T3 | 1929.3 | 1897.8 | 1927.8 | 1903.6 |
| T4 | 1777.6 | 1726.8 | 1765.1 | 1728.4 |
| T5 | 42.4 | (same) | (same) | (same) |
| T6 | 96.7 | 110.0 | 105.0 | 104.6 |
| N (rpm) | 36116 | 34991 | 36120 | 36993 |

Table A3. Summary of imbalance predictions for high pressure fuel turbopump subsystem analyses

| IMBALANCE | INITIAL | CASE A FINAL | CASE B FINAL | CASE C FINAL |
|---|---|---|---|---|
| Prebnr Flow (lbm/s) | 0.0004 | -0.2444 | 0.002 | -0.0119 |
| Prebnr Power (Btu/s) | -6493.7 | 113.1 | -1265.2 | 1009.1 |
| $O_2$ Resis $\Delta P$ (psia) | -455.5 | -5.3 | -131.9 | -110.0 |
| $H_2$ Resis $\Delta P$ (psia) | 179.7 | 8.1 | 234.7 | -52.6 |
| HPFP $\Delta P$ (psia) | -34.8 | -14.7 | -104.0 | -1015.6 |
| HPFP Power (hp) | -10997.2 | -218.3 | -5209.2 | -1662.6 |
| HPFT $\Delta P$ (psia) | 12.7 | 11.0 | 8.9 | 47.4 |
| HPFT $\Delta T$ (deg R) | -0.0016 | 24.6 | 10.8 | 22.1 |
| HPFTP Power (Btu/s) | 1006.1 | -794.7 | 420.5 | -347.4 |
| $O_2/H_2$ Ratio | 0.0 | -0.0063 | 0.0089 | -0.0004 |

# APPENDIX B

## FIGURES

FIGURE B1. MODEL HIERARCHY DIAGRAM

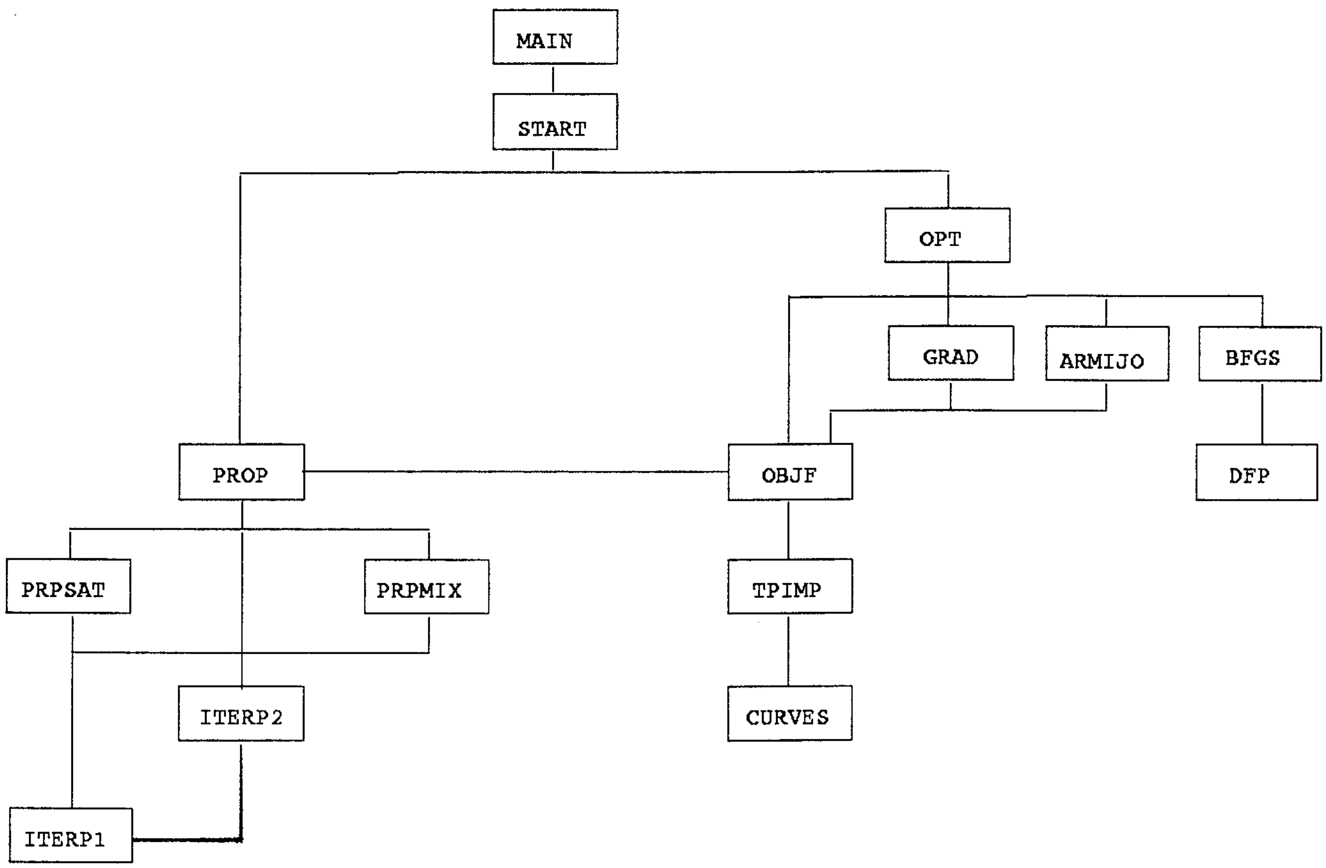
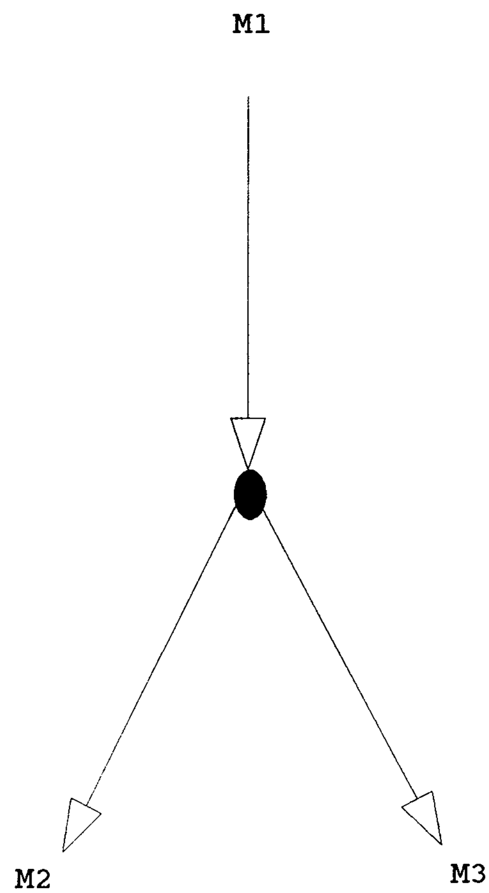
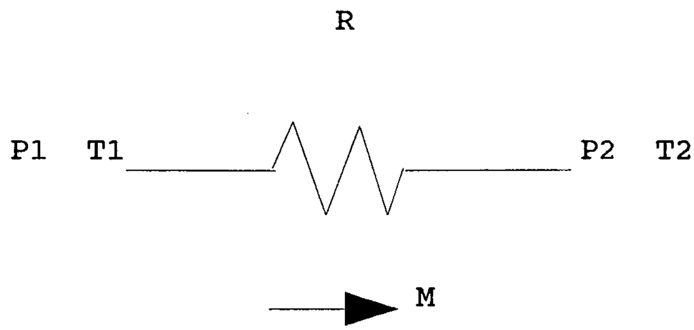


**FIGURE B2. JUNCTION FLOW BALANCE**




M1 - M2 - M3 = FLOW RESIDUAL

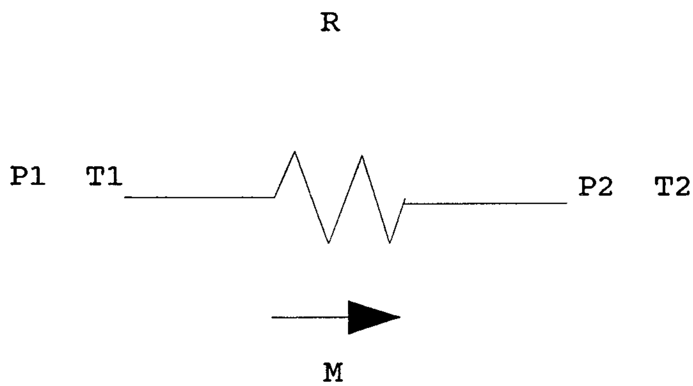
**FIGURE B3. BRANCH PRESSURE LOSS**




(P1-P2) - R * W**2 / Dens(P1,T1) = PRESSURE RESIDUAL

W = M * (gstd/gc)

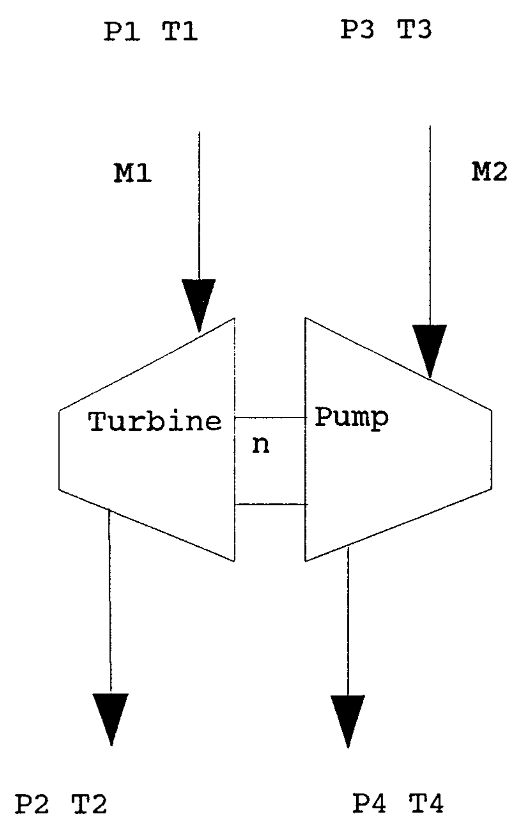
**FIGURE B4. BRANCH ENERGY BALANCE**




h(P1,T1) - h(P2,T2) = ENERGY RESIDUAL

P1 , P2 = TOTAL PRESSURES

**FIGURE B5a. TURBOPUMP BALANCES**




M1 * [h(P1,T1) - h(P2,T2)] + M2*[h(P3,T3) - h(P4,T4)] = POWER RESIDUAL

**FIGURE B5b. TURBOPUMP BALANCES**

Dens(P1,T1) * (n*D)**2 * [CH - CH(CQ)] = PUMP DELTA P RESIDUAL

CH = computed head coefficient
CH(CQ) = pump head coef as a function of flow coef
Eff = computed pump efficiency
Eff(CQ) = pump efficiency as a function of flow coef

$$\frac{M1*(P2 - P1)/Dens(P1,T1)}{Eff} - \frac{M1*(P2-P1)/Dens(P1,T1)}{Eff(CQ)}$$

= PUMP POWER RESIDUAL

**FIGURE B5c. TURBINE BALANCES**

```
[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**

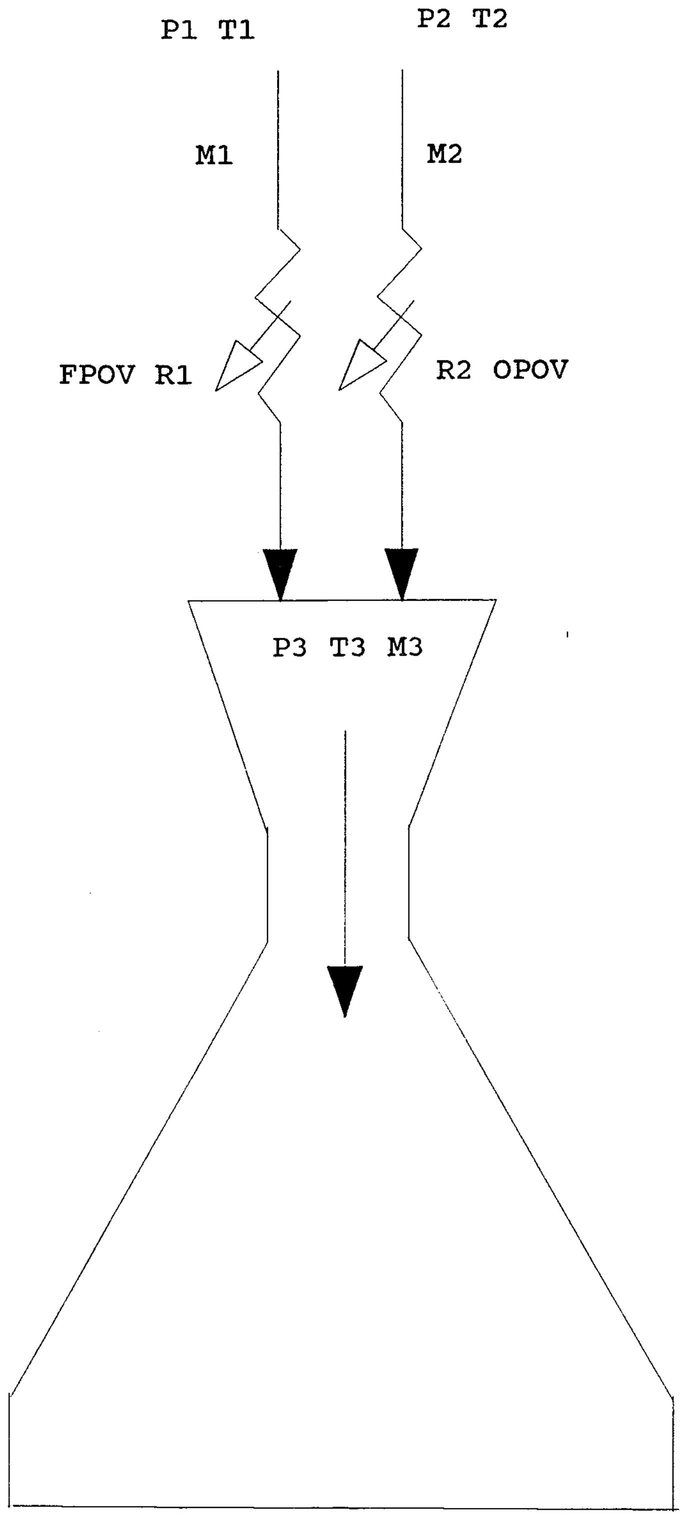
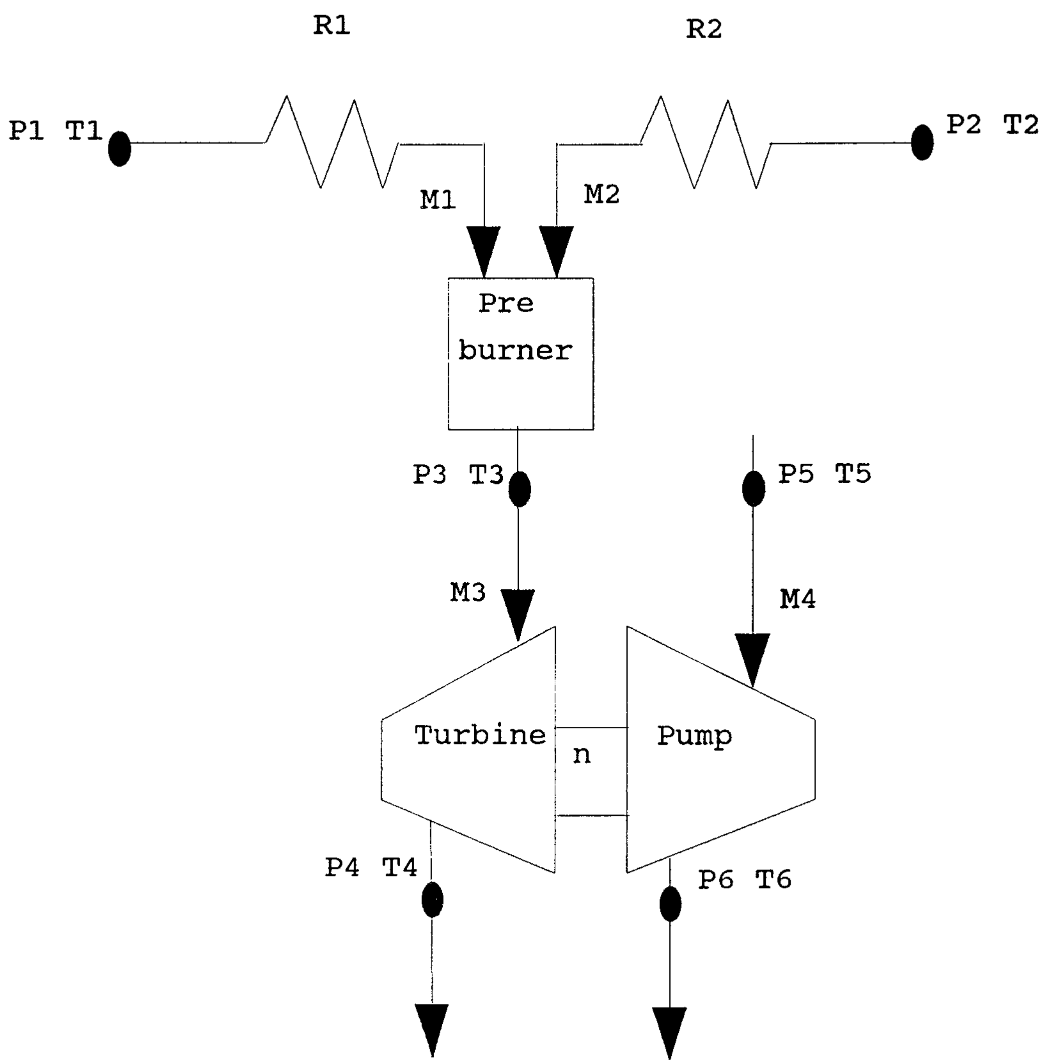


**FIGURE B6b. NOZZLE AND OVERALL BALANCES**

```
M2/M1 - COMMANDED RATIO  =  MIXTURE RATIO RESIDUAL

M1 + M2 - M3                    =  FLOW RESIDUAL 1

M3 - M3(P3,T3,nozzle geometry)  =  FLOW RESIDUAL 2

M1*h(P1,T1) + M2*h(P2,T2) - M3*h(P3,T3)  =  ENERGY RESIDUAL

(P1-P3) - R1*W1**2/Dens(P1,T1) = PRESSURE RESIDUAL 1

(P2-P3) - R2*W2**2/Dens(P2,T2) = PRESSURE RESIDUAL 2

Thrust(M3,P3,T3,nozzle geometry) - COMMAND THRUST =  THRUST RESIDUAL
```

**FIGURE B7. TEST CONFIGURATION**



# APPENDIX C

# DOCUMENTATION

## C1. DESCRIPTION OF ROUTINES

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

## C2. INPUT/OUTPUT VARIABLE DEFINITIONS

Program Input File Name = **TM2-IO.DAT**

Data Type = I/O data file identification

| INPUT VARIABLE NAME | | DEFINITION |
|---|---|---|
| PDAT | - | name of file containing fluid property data |
| TDAT | - | name of file containing solution initiation data |
| UDAT | - | name of file containing residual uncertainty estimates |
| VDAT | - | name of file containing volume definition and connection information |
| ODAT | - | name of file containing solution output |

Program Input File Name = **VDAT**

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

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

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<br>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<br>0 = fixed >0 = variable |
| INTYPE(I) - | type number of node I<br>0 = fixed pressure and temperature<br>1 = variable pressure, fixed temperature<br>>1 = variable pressure and temperature |
| IRTYPE(I) - | type number of resistance in pressure circuit I<br>0 = fixed resistance<br>>0 = variable resistance |
| ITPTY(I) - | type number of turbopump I<br>1 = one turbine, one pump<br>>1 = one turbine, two pumps |
| ITPD(I,J) - | TTB array location containing the impeller diameter of turbomachine J in turbopump circuit I<br>J = 1 turbine J = 2 or 3 pump |
| ITPN(I,J) - | node number associated with mass flow J crossing boundary of turbopump I<br>J = 1 turbine inlet J = 2 turbine outlet<br>J = 3 pump 1 inlet J = 4 pump 1 outlet<br>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<br>J = 1 turbine inlet J = 2 turbine outlet<br>J = 3 pump 1 inlet J = 4 pump 1 outlet<br>J = 5 pump 2 inlet J = 6 pump 2 outlet |

Program Input File Name = **VDAT** (continued 3)

INPUT VARIABLE NAME — DEFINITION

ICURV(I,J)- curve number associated with performance curve J of turbopump I

- J = 1 turbine performance curve 1
- J = 2 turbine performance curve 2
- J = 3 pump 1 performance curve 1
- J = 4 pump 1 performance curve 2
- J = 5 pump 2 performance curve 1
- J = 6 pump 2 performance curve 2

Program Input File Name = **TDAT**

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

| INPUT VARIABLE NAME | | DEFINITION |
|---|---|---|
| NDESC | - | number of analysis description items |
| NTTB | - | number of elements in the TTB solution initializing array |
| DESC(I) | - | analysis description item I (character variable) |
| TTB(I) | - | value of solution initializing variable assigned to array address I |

Program Input File Name = **UDAT**

Actual Access File Name = "named in TM2-IO.DAT"
Data Type = uncertainty values associated with model relations

| INPUT VARIABLE NAME | | DEFINITION |
|---|---|---|
| UOF | - | uncertainty of engine oxygen/fuel ratio |
| UEVOL(I) | - | power imbalance uncertainty associated with energy circuit I |
| UMVOL(I) | - | mass flow rate imbalance uncertainty associated with mass flow circuit I |
| UPVOL(I) | - | pressure imbalance uncertainty associated with pressure circuit I |
| UETP(I) | - | power imbalance uncertainty associated with turbopump circuit I |
| UT1TP(I) | - | turbine characteristic 1 (pressure drop) imbalance uncertainty associated with turbopump circuit I |
| UT2TP(I) | - | turbine characteristic 2 (temperature drop) imbalance uncertainty associated with turbopump circuit I |
| UP1TP(I) | - | pump 1 characteristic 1 (pressure drop) imbalance uncertainty associated with turbopump circuit I |
| UP2TP(I) | - | pump 1 characteristic 2 (power) imbalance uncertainty associated with turbopump circuit I |
| UP3TP(I) | - | pump 2 characteristic 1 (pressure drop) imbalance uncertainty associated with turbopump circuit I |
| UP4TP(I) | - | pump 2 characteristic 2 (power) imbalance uncertainty associated with turbopump circuit I |

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)

| INPUT VARIABLE NAME | | DEFINITION |
|---|---|---|
| NEC | - | number of energy circuits in system |
| NMC | - | number of mass flow circuits in system |
| NPC | - | number of pressure circuits in system |
| NTPC | - | number of turbopump circuits in system |
| IEN(I,J) | - | node number associated with flow J across boundary of energy circuit I |
| IMBEC(I,J) | - | branch number of mass flow J across boundary of energy circuit I |
| IMBMC(I,J) | - | branch number of mass flow J across boundary of mass circuit I |
| 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 |
| ITPN(I,J) | - | node number associated with mass flow J crossing boundary of turbopump I<br>J = 1 turbine inlet  J = 2 turbine outlet<br>J = 3 pump 1 inlet  J = 4 pump 1 outlet<br>J = 5 pump 2 inlet  J = 6 pump 2 outlet |
| IMBTP(I,J) | - | branch number associated with mass flow J crossing boundary of turbopump I<br>J = 1 turbine inlet  J = 2 turbine outlet<br>J = 3 pump 1 inlet  J = 4 pump 1 outlet<br>J = 5 pump 2 inlet  J = 6 pump 2 outlet |
| ETPO(I) | - | initial power imbalance associated with turbopump circuit I |
| ETP(I) | - | final power imbalance associated with turbopump circuit I |
| T1TPO(I) | - | initial turbine characteristic 1 (pressure drop) imbalance associated with turbopump circuit I |
| T1TP(I) | - | final turbine characteristic 1 (pressure drop) imbalance associated with turbopump circuit I |
| T2TPO(I) | - | initial turbine characteristic 2 (temperature drop) imbalance associated with turbopump circuit I |
| T2TP(I) | - | final turbine characteristic 2 (temperature drop) imbalance associated with turbopump circuit I |
| P1TPO(I) | - | initial pump 1 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I |
| P1TP(I) | - | final pump 1 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I |
| P2TPO(I) | - | initial pump 1 characteristic 2 (power) imbalance associated with turbopump circuit I |
| P2TP(I) | - | final pump 1 characteristic 2 (power) imbalance associated with turbopump circuit I |

Program Output File Name = **ODAT** (continued 2)

| INPUT VARIABLE NAME | | DEFINITION |
|---|---|---|
| P3TPO(I) | - | initial pump 2 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I |
| P3TP(I) | - | final pump 2 characteristic 1 (pressure drop) imbalance associated with turbopump circuit I |
| P4TPO(I) | - | initial pump 2 characteristic 2 (power) imbalance associated with turbopump circuit I |
| P4TP(I) | - | final pump 2 characteristic 2 (power) imbalance associated with turbopump circuit I |

## C3. SOURCE CODE LISTING

```
C     PROGRAM TM-02
C
      CHARACTER*24 DESC
      CHARACTER*12 PDAT,TDAT,UDAT,VDAT,ODAT
      INTEGER      EDIR,PDIR
C
      COMMON /BALC/ EIMB(5)   , PIMB(6)   , WIMB(5)   , WZIMB   , OFIMB ,
     1              EIMB0(5)  , PIMB0(6)  , WIMB0(5)  , WZIMB0  , OFIMB0,
     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              REVA(20)  , REVM(20)  , REVP(20)  , REVT(20),
     2              REVR(20)  , REVS(20)  , REVD(20)  , REVH(20)
C
      COMMON /VDAT/ NEC, NMC, NPC, NTPC, NBRH, NNOD, NHGNOD, NRPM,
     1              INOZBR,INOZEC,INOZMC,INOZN,IHGNOZ,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)  ,
     5              IPN(6,2)   , IMBPC(6)  , NH2HG(5)   , NO2HG(5)   ,
     6              IHGN(5)    , ICEFF(5)  , IH2HG(5,5) , IO2HG(5,5) ,
     7              IBTYPE(20) , INTYPE(20), IRTYPE(20) , ITPTY(4)   ,
     1              ITPD(4,3)  , ITPN(4,6) , ITPS(4)    , IMBTP(4,3) ,
     2              ICURV(4,6)
C
      COMMON /TDAT/ NDESC, NTTB, DESC(5), TTB(1350)
C
      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)
C
      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)
      COMMON /H2OPRP/
     * H2OP1(7),H2OT1(13),H2OH1(7,13),H2OS1(7,13),H2OD1(7,13)
C
      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
C
      DIMENSION
     * NH2PA(4),NH2TA(4),NO2PA(3),NO2TA(3),NH2OPA(1),NH2OTA(1)
C
      CHARACTER*70 PTITLE
C
      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)/13/
C
      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(1)=NH2OPA(1)
      NH2OT(1)=NH2OTA(1)
C
      HH2REF =  1790.091
      HO2REF =   234.681
      HWAREF =  1339.990
      SH2REF =    15.440
      SO2REF =     1.530
      SWAREF =     2.294
      SH2A   =    15.481
```

```
      SO2A   =      1.531
      SWAA   =      0.928
C
      OPEN ( 7, FILE = 'TM1-IO.DAT' , STATUS = 'OLD' )
      READ ( 7, * )  PDAT, TDAT, UDAT, VDAT, ODAT
C
      OPEN (  8, FILE = PDAT , STATUS = 'OLD' )
      OPEN ( 12, FILE = TDAT , STATUS = 'OLD' )
      OPEN ( 13, FILE = UDAT , STATUS = 'OLD' )
      OPEN ( 14, FILE = VDAT , STATUS = 'OLD' )
C
      OPEN ( 21, FILE = ODAT  )
C
C     **  READ IN H2 PROPERTY TABLE INTO ARRAYS  **
C
      DO 10 ITBL=1,4
C
      READ(8,902) PTITLE
      DO 10 I=1,NH2P(ITBL)
      DO 10 J=1,NH2T(ITBL)
C
      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)
C
   10 CONTINUE
C
C     **  READ IN O2 PROPERTY TABLE INTO ARRAYS  **
C
      DO 20 ITBL=1,3
C
      READ(8,902) PTITLE
      DO 20 I=1,NO2P(ITBL)
      DO 20 J=1,NO2T(ITBL)
C
      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)
C
   20 CONTINUE
C
C     **  READ IN STEAM PROPERTY TABLES INTO ARRAYS  **
C
      DO 30  ITBL = 1, 1
C
      READ(8,902) PTITLE
      DO 30  I = 1, NH2OP(ITBL)
      DO 30  J = 1, NH2OT(ITBL)
C
      IF( ITBL .EQ. 1 )  READ(8,*) H2OP1(I),H2OT1(J),
     1   H2OH1(I,J),H2OS1(I,J),H2OD1(I,J)
C
   30 CONTINUE
C
      READ  (12,*)     NDESC, NTTB
      READ  (12,*)     ( DESC( I ),  I = 1, NDESC )
      READ  (12,903)   ( TTB( I ) ,  I = 1, NTTB )
C
      WRITE (21,901)   ( DESC( I ),  I = 1, NDESC )
C
      READ (14,*)  NEC, NMC, NPC, NTPC, NBRH, NNOD, NHGNOD, NRPM,
     1             INOZBR,INOZEC,INOZMC,INOZN,IHGNOZ,IMIXR,IATHRT
      READ (14,*)  ( IA(I)   ,  I = 1, NNOD ),
     1             ( IP(I)   ,  I = 1, NNOD ),
     2             ( IT(I)   ,  I = 1, NNOD ),
     3             ( MAT(I)  ,  I = 1, NNOD ),
     4             ( IM(I)   ,  I = 1, NBRH ),
     5             ( IR(I)   ,  I = 1, NPC  ),
     6             ( IS(I)   ,  I = 1, NRPM ),
     7             ( NEIO(I) ,  I = 1, NEC  ),
```

```
     1                 ( NMIO(I)  ,   I = 1, NMC   ),
     2                 ( IBTYPE(I), I = 1, NBRH  ),
     3                 ( INTYPE(I), I = 1, NNOD  ),
     4                 ( IRTYPE(I), I = 1, NPC   )
C
      DO 40  I = 1, NMC
      READ (14,*)     ( MDIR(I,J) ,   J = 1, NMIO(I) ),
     1                 ( IMBMC(I,J),   J = 1, NMIO(I) )
   40 CONTINUE
C
      DO 50  I = 1, NEC
      READ (14,*)     ( EDIR(I,J) ,   J = 1, NEIO(I) ),
     1                 ( IMBEC(I,J),   J = 1, NEIO(I) ),
     2                 ( IEN(I,J)  ,   J = 1, NEIO(I) )
   50 CONTINUE
C
      DO 60  I = 1, NPC
      READ (14,*)     ( PDIR(I,J)  ,  J = 1, 2 ),
     1                   IMBPC(I),
     2                 ( IPN(I,J)   ,  J = 1, 2 )
   60 CONTINUE
C
      IF ( NHGNOD .GT. 0 )  THEN
      READ (14,*)     ( IHGN(I)  ,   I = 1, NHGNOD ),
     1                 ( NH2HG(I),   I = 1, NHGNOD ),
     2                 ( NO2HG(I),   I = 1, NHGNOD ),
     3                 ( ICEFF(I),   I = 1, NHGNOD )
         DO 70  I = 1, NHGNOD
         READ (14,*)     ( IH2HG(I,J),   J = 1, NH2HG(I) ),
     1                    ( IO2HG(I,J),   J = 1, NO2HG(I) )
   70    CONTINUE
      ENDIF
C
      READ (14,*)     ( ITPTY(I),   I = 1, NTPC ),
     1                 ( ITPS(I) ,   I = 1, NTPC )
C
      DO 80  I = 1, NTPC
      IF (ITPTY(I).EQ.1) THEN
         READ (14,*)     ( ITPD(I,J) ,   J = 1, 2 ),
     1                    ( IMBTP(I,J),   J = 1, 2 ),
     2                    ( ITPN(I,J) ,   J = 1, 4 ),
     3                    ( ICURV(I,J),   J = 1, 4 )
      ELSE
         READ (14,*)     ( ITPD(I,J) ,   J = 1, 3 ),
     1                    ( IMBTP(I,J),   J = 1, 3 ),
     2                    ( ITPN(I,J) ,   J = 1, 6 ),
     3                    ( ICURV(I,J),   J = 1, 6 )
      ENDIF
   80 CONTINUE
C
      READ (13,*)     ( UEVOL(I),   I = 1, NEC    ),
     1                 ( UMVOL(I),   I = 1, NMC    ),
     2                 ( UPVOL(I),   I = 1, NPC    ),
     3                 ( UETP(I) ,   I = 1, NTPC   ),
     4                 ( UT1TP(I),   I = 1, NTPC   ),
     5                 ( UT2TP(I),   I = 1, NTPC   ),
     6                 ( UP1TP(I),   I = 1, NTPC   ),
     7                 ( UP2TP(I),   I = 1, NTPC   ),
     1                 ( UP3TP(I),   I = 1, NTPC   ),
     2                 ( UP4TP(I),   I = 1, NTPC   ),
     3                   UOF
C
      CALL START
C
  901 FORMAT ( 10 ( /, 1X, A24 ) )
  902 FORMAT ( /A70/ )
  903 FORMAT ( 8X, 5E13.6 )
C
C
      STOP
      END
C**********************************************************************
      SUBROUTINE START
C
C     START ROUTINE
C
      CHARACTER*24 DESC
      INTEGER      EDIR, PDIR
```

```
      REAL         JOULE
C
      DIMENSION X(25),X2(25)
C
      COMMON /BALC/ EIMB(5)   , PIMB(6)   , WIMB(5)   , WZIMB    , OFIMB ,
     1              EIMB0(5)  , PIMB0(6)  , WIMB0(5)  , WZIMB0   , OFIMB0,
     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              REVA(20)  , REVM(20)  , REVP(20)  , REVT(20),
     2              REVR(20)  , REVS(20)  , REVD(20)  , REVH(20)
C
      COMMON /VDAT/ NEC, NMC, NPC, NTPC, NBRH, NNOD, NHGNOD, NRPM,
     1              INOZBR,INOZEC,INOZMC,INOZN,IHGNOZ,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)  ,
     5              IPN(6,2)    , IMBPC(6)  , NH2HG(5)    , NO2HG(5)   ,
     6              IHGN(5)     , ICEFF(5)  , IH2HG(5,5)  , IO2HG(5,5) ,
     7              IBTYPE(20)  , INTYPE(20), IRTYPE(20)  , ITPTY(4)   ,
     1              ITPD(4,3)   , ITPN(4,6) , ITPS(4)     , IMBTP(4,3) ,
     2              ICURV(4,6)
C
      COMMON /TDAT/ NDESC, NTTB, DESC(5), TTB(1350)
C
      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)
C
      COMMON /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)
      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)
C
      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)
C
      PARAMETER ( JOULE = 778.16, GC = 32.174 )
C
      IFUNC=0
C
      WRITE ( 21, 941 )
      DO  2  IMC = 1, NMC
      WRITE ( 21, 942 )  IMC, ( IMBMC(IMC,IIO), IIO = 1, NMIO( IMC ) )
    2 CONTINUE
C
      WRITE ( 21, 943 )
      DO  4  IEC = 1, NEC
      WRITE ( 21, 942 )  IEC, ( IMBEC(IEC,IIO), IIO = 1, NEIO( IEC ) )
      WRITE ( 21, 944 )  IEC, (    IEN(IEC,IIO), IIO = 1, NEIO( IEC ) )
    4 CONTINUE
C
      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
C
      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
C
C ********** INITIALIZE ALGORITHM ***************************************
C
      NMAX=100
```

```
      EPSC=1.E-6
      EPST=1.E-6
      RHO=1.0
      ALPHA=0.0
      BETA=0.4
C
C ********** INITIALIZE INDEPENDENT VARIABLES X(I) *******************
C
      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
C
      DO 20 I=1,NNOD
      REVP(I)=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
C
      DO 40 I=1,NPC
      REVR(I)=TTB(IR(I))
      IF (IRTYPE(I).EQ.0) THEN
         GO TO 40
      ELSE
         X(K)=SQRT(REVR(I))
         K=K+1
      ENDIF
   40 CONTINUE
C
      DO 50 I=1,NRPM
      REVS(I)=TTB(IS(I))
      X(K)=SQRT(REVS(I))
      K=K+1
   50 CONTINUE
C
      N=K-1
C
      DO 100  I = 1, NNOD
      P=REVP( I )
      T=REVT( I )
C
      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)
      REVD(I)=RHO
      REVH(I)=H-HH2REF
      GO TO 100
C
   60 CALL PROP ( 2, P, T, 0.0, 0.0, H, S, RHO)
      REVD(I)=RHO
      REVH(I)=H-HO2REF
      GO TO 100
C
   70 IDHG=1
   72 IF (IHGN(IDHG).EQ.I) THEN
         IHG=IDHG
         GO TO 74
      ELSE
         IDHG=IDHG+1
         IF (IDHG.GT.NHGNOD) THEN
```

```
            GO TO 74
         ELSE
            GO TO 72
         ENDIF
      ENDIF
C
   74 WH2=0.0
      DO 80 IH2IN=1,NH2HG(IHG)
      IBRH=IH2HG(IHG,IH2IN)
      WH2=WH2+REVM(IBRH)
   80 CONTINUE
C
      WO2=0.0
      DO 90 IO2IN=1,NO2HG(IHG)
      IBRH=IO2HG(IHG,IO2IN)
      WO2=WO2+REVM(IBRH)
   90 CONTINUE
C
      CEFF=TTB(ICEFF(IHG))
      OF=WO2/WH2
      CALL PROP ( 4, P, T, OF, CEFF, H, S, RHO)
      REVD(I)=RHO
      REVH(I)=H
C
  100 CONTINUE
C
C ********** CALL OPTIMIZATION STRATEGY ******************************
C
      CALL OPT (N,NMAX,EPSC,EPST,RHO,ALPHA,BETA,ISTAGE,F2,GNORM,
     1          DXNORM,X,X2)
C
      WRITE (21,801)
      DO 200 I=1,NBRH
      WRITE (21,802) I,TTB(IM(I)),REVM(I)
  200 CONTINUE
C
      DO 210 I=1,NNOD
      WRITE (21,803) I,TTB(IP(I)),REVP(I)
  210 CONTINUE
C
      DO 220 I=1,NNOD
      WRITE (21,804) I,TTB(IT(I)),REVT(I)
  220 CONTINUE
C
      DO 230 I=1,NRPM
      WRITE (21,805) I,TTB(IS(I)),REVS(I)
  230 CONTINUE
C
      WRITE (21,806)
      DO 240 I=1,NMC
      WRITE (21,807) I,WIMB0(I),WIMB(I)
  240 CONTINUE
C
      DO 250 I=1,NEC
      WRITE (21,808) I,EIMB0(I),EIMB(I)
  250 CONTINUE
C
      DO 260 I=1,NPC
      WRITE (21,809) I,PIMB0(I),PIMB(I)
  260 CONTINUE
C
      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
```

```
C
      WRITE (21,812) I,OFIMB0,OFIMB
      WRITE (21,813) I,WZIMB0,WZIMB
C
 801  FORMAT (/12X,'ANALYSIS RESULTS',/4X,'VARIABLE',15X,
     1         'INITIAL',15X,'FINAL')
 802  FORMAT (1X,'    FLOW',I3,7X,F15.4,5X,F15.4,5X,'(LB/S)')
 803  FORMAT (1X,'PRESSURE',I3,7X,F15.4,5X,F15.4,5X,'(PSI)')
 804  FORMAT (1X,'    TEMP',I3,7X,F15.4,5X,F15.4,5X,'(DEG R)')
 805  FORMAT (1X,'   SPEED',I3,7X,F15.4,5X,F15.4,5X,'(RPM)')
 806  FORMAT (/16X,'BALANCES',/8X,'CIRCUIT',12X,'INITIAL',12X,
     1         'FINAL')
 807  FORMAT (5X,'    MASS',I2,7X,F12.4,5X,F12.4,5X,'(LB/S)')
 808  FORMAT (5X,'  ENERGY',I2,7X,F12.4,5X,F12.4,5X,'(BTU/S)')
 809  FORMAT (5X,'PRESSURE',I2,7X,F12.4,5X,F12.4,5X,'(PSI)')
 810  FORMAT ( 1X,'  ENERGY STP',I2,7X,F12.4,5X,F12.4,5X,'(BTU/S)',
     1         /1X,' TURB DP STP',I2,7X,F12.4,5X,F12.4,5X,'(PSI)',
     2         /1X,' TURB DT STP',I2,7X,F12.4,5X,F12.4,5X,'(R)',
     3         /1X,' PUMP DP STP',I2,7X,F12.4,5X,F12.4,5X,'(PSI)',
     4         /1X,'PUMP PWR STP',I2,7X,F15.4,5X,F15.4,5X,'(BTU/S)')
C
 811  FORMAT ( 1X,'  ENERGY BTP',I2,7X,F12.4,5X,F12.4,5X,'(BTU/S)',
     1         /1X,' TURB DP BTP',I2,7X,F12.4,5X,F12.4,5X,'(PSI)',
     2         /1X,' TURB DT BTP',I2,7X,F12.4,5X,F12.4,5X,'(R)',
     3         /1X,'PUMP1 DP BTP',I2,7X,F12.4,5X,F12.4,5X,'(PSI)',
     4         /1X,'PUMP1 PW BTP',I2,7X,F15.4,5X,F15.4,5X,'(BTU/S)',
     5         /1X,'PUMP2 DP BTP',I2,7X,F12.4,5X,F12.4,5X,'(PSI)',
     6         /1X,'PUMP2 PW BTP',I2,7X,F15.4,5X,F15.4,5X,'(BTU/S)')
C
 812  FORMAT (/1X,'O/F - O/F COMMANDED',5X,'INITIAL',F7.3,5X,'FINAL',
     1        F7.3)
 813  FORMAT (/1X,'NOZZLE FLOW IMBALNC',5X,'INITIAL',F7.3,5X,'FINAL',
     1        F7.3)
C
 941  FORMAT ( /4X, 'FLOW SUBSYSTEM NO', 5X, 'I/O NUMBER', 6X,
     1        '    1    2    3    4    5    6' )
 942  FORMAT ( 8X, I5, 12X, 'I/O FLOW BRANCHES', 6I5 )
 943  FORMAT ( /2X, 'ENERGY SUBSYSTEM NO', 5X, 'I/O NUMBER', 6X,
     1        '    1    2    3    4    5    6' )
 944  FORMAT ( 8X, I5, 12X, 'I/O ENERGY NODES ', 6I5 )
 945  FORMAT ( /1X, 'DELTA P SUBSYSTEM NO', 5X, 'I/O NUMBER', 6X,
     1        '    1    2' )
 946  FORMAT ( /9X, 'TURBOPUMP NO', 5X, 'I/O NUMBER', 6X,
     1        '    1    2    3    4    5    6' )
C
      RETURN
      END
C
C
C ********************************************************************
C
      SUBROUTINE OBJF (IFUNC,X,F)
C
      CHARACTER*24 DESC
      INTEGER      EDIR, PDIR
      REAL         JOULE
C
      DIMENSION X(25)
C
      COMMON /BALC/ EIMB(5)  , PIMB(6)  , WIMB(5)  , WZIMB   , OFIMB ,
     1              EIMB0(5) , PIMB0(6) , WIMB0(5) , WZIMB0  , OFIMB0,
     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              REVA(20) , REVM(20) , REVP(20) , REVT(20),
     2              REVR(20) , REVS(20) , REVD(20) , REVH(20)
C
      COMMON /VDAT/ NEC, NMC, NPC, NTPC, NBRH, NNOD, NHGNOD, NRPM,
     1              INOZBR,INOZEC,INOZMC,INOZN,IHGNOZ,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)  ,
     5              IPN(6,2)   , IMBPC(6)  , NH2HG(5)   , NO2HG(5)   ,
     6              IHGN(5)    , ICEFF(5)  , IH2HG(5,5) , IO2HG(5,5) ,
     7              IBTYPE(20) , INTYPE(20), IRTYPE(20) , ITPTY(4)   ,
     1              ITPD(4,3)  , ITPN(4,6) , ITPS(4)    , IMBTP(4,3) ,
     2              ICURV(4,6)
```

```
C
      COMMON /TDAT/ NDESC, NTTB, DESC(5), TTB(1350)
C
      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)
C
      COMMON /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)
      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)
C
      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)
C
      PARAMETER ( JOULE = 778.16, GC = 32.174 )
C
      IFUNC=IFUNC+1
C
      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
   10 CONTINUE
C
      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
   20 CONTINUE
C
      DO 40 I=1,NPC
      IF (IRTYPE(I).EQ.0) THEN
         GO TO 40
      ELSE
         REVR(I)=X(K)**2
         K=K+1
      ENDIF
   40 CONTINUE
C
      DO 50 I=1,NRPM
      REVS(I)=X(K)**2
      K=K+1
   50 CONTINUE
C
      DO 100  I = 1, NNOD
      IF (INTYPE(I).EQ.0) THEN
         GO TO 100
      ELSE
         P=REVP( I )
         T=REVT( I )
C
         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)
         REVD(I)=RHO
         REVH(I)=H-HH2REF
         GO TO 100
C
   60    CALL PROP ( 2, P, T, 0.0, 0.0, H, S, RHO)
         REVD(I)=RHO
         REVH(I)=H-HO2REF
         GO TO 100
C
   70    IDHG=1
   72    IF (IHGN(IDHG).EQ.I) THEN
            IHG=IDHG
            GO TO 74
         ELSE
            IDHG=IDHG+1
            IF (IDHG.GT.NHGNOD) THEN
               GO TO 74
            ELSE
               GO TO 72
            ENDIF
         ENDIF
C
   74    WH2=0.0
         DO 80 IH2IN=1,NH2HG(IHG)
         IBRH=IH2HG(IHG,IH2IN)
         WH2=WH2+REVM(IBRH)
   80    CONTINUE
C
         WO2=0.0
         DO 90 IO2IN=1,NO2HG(IHG)
         IBRH=IO2HG(IHG,IO2IN)
         WO2=WO2+REVM(IBRH)
   90    CONTINUE
C
         CEFF=TTB(ICEFF(IHG))
         OF=WO2/WH2
         CALL PROP ( 4, P, T, OF, CEFF, H, S, RHO)
         REVD(I)=RHO
         REVH(I)=H
      ENDIF
C
  100 CONTINUE
C
      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)
  110 CONTINUE
C
      F=F+(WIMB(IMC)/UMVOL(IMC))**2
C
  120 CONTINUE
C
      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=REVH(INOD)
      W=REVM(IBRH)
      EIMB(IEC)=EIMB(IEC)+IOD*W*ENTH
  130 CONTINUE
C
C     HEAT=f(RefT1,RefT2,RefM1,RefM2,BoundaryProps)
C     EIMB(IEC)=EIMB(IEC)+HEAT
C
      F=F+(EIMB(IEC)/UEVOL(IEC))**2
C
  140 CONTINUE
C
      DO 160 IPC=1,NPC
      IBRH=IMBPC(IPC)
      RES=REVR(IPC)
```

```
      W=REVM(IBRH)
      DO 150 IIO=1,2
      IOD=PDIR(IPC,IIO)
      INOD=IPN(IPC,IIO)
      IF (IOD.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)/UPVOL(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=IMBTP(I,1)
         IBRH2=IMBTP(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=REVM(IBRH1)
         W2=REVM(IBRH2)
         P1=REVP(INOD1)
         P2=REVP(INOD2)
         P3=REVP(INOD3)
         P4=REVP(INOD4)
         T1=REVT(INOD1)
         T2=REVT(INOD2)
         T3=REVT(INOD3)
         T4=REVT(INOD4)
         D1=REVD(INOD1)
         D2=REVD(INOD2)
         D3=REVD(INOD3)
         D4=REVD(INOD4)
         H1=REVH(INOD1)
         H2=REVH(INOD2)
         H3=REVH(INOD3)
         H4=REVH(INOD4)
         SPEED=REVS(ISPEED)
C
         ETP(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)/UT1TP(I))**2 +
     2             (T2TP(I)/UT2TP(I))**2 +
     3             (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=IMBTP(I,1)
         IBRH2=IMBTP(I,2)
         IBRH3=IMBTP(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=REVS(ISPEED)
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,
     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))
          CALL TPIMB (I,ICUR5,ICUR6,DIA3,W3,SPEED,D5,D6,P5,P6,T5,T6,H5,
     1                H6,P3TP(I),P4TP(I))
C
          F  =  F + (ETP(I) /UETP(I)) **2 +
     1              (T1TP(I)/UT1TP(I))**2 +
     2              (T2TP(I)/UT2TP(I))**2 +
     3              (P1TP(I)/UP1TP(I))**2 +
     4              (P2TP(I)/UP2TP(I))**2 +
     5              (P3TP(I)/UP3TP(I))**2 +
     6              (P4TP(I)/UP4TP(I))**2
        ENDIF
C
 170    CONTINUE
C
        WH2=0.0
        WO2=0.0
C
        DO 180 IH2IN=1,NH2HG(IHGNOZ)
        IBRH2=IH2HG(IHGNOZ,IH2IN)
        WH2=WH2+REVM(IBRH2)
 180    CONTINUE
C
        DO 190 IO2IN=1,NO2HG(IHGNOZ)
        IBRH2=IO2HG(IHGNOZ,IO2IN)
        WO2=WO2+REVM(IBRH2)
 190    CONTINUE
C
        ATHROT=TTB(IATHRT)
        OFCOM =TTB(IMIXR)
        WTOTAL=WO2+WH2
        OFCAL=WO2/WH2
```

```
      OFIMB=OFCAL-OFCOM
C
      F=F+(OFIMB/UOF)**2
C
      GAMH2N=1.3
      GAMO2N=1.3
      GAMH2O=1.3
      PTOTAL=REVP(INOZN)
      TTOTAL=REVT(INOZN)
      CEFF=TTB(ICEFF(IHGNOZ))
      XF   = 1.0 / (1.0 + OFCAL)
      XO   = 1.0 - XF
      XH2  = XF - XO * 2.0 * CEFF * 2.016   / 31.9988
      XH2O =      XO * 2.0 * CEFF * 18.0153 / 31.9988
      XO2  = 1.0 - XH2 - XH2O
      GAMMA=XH2*GAMH2N+XH2O*GAMH2O+XO2*GAMO2N
      GASCON=(XH2/2.016+XH2O/18.0153+XO2/31.9988)*1545.3
      TTHROT=TTOTAL/(1+(GAMMA-1)/2)
      PTHROT=PTOTAL/(1+(GAMMA-1)/2)**(GAMMA/(GAMMA-1))
      VTHROT=SQRT(GAMMA*GASCON*TTHROT*GC)
      RHO=144.0*PTHROT/(GASCON*TTHROT)
      WNOZ=RHO*ATHROT*VTHROT
      WCAL=REVM(INOZBR)
      WZIMB=WCAL-WNOZ
C
      F=F+(WZIMB/UMVOL(INOZMC))**2
C
      IF (IFUNC.EQ.1) THEN
         DO 210 I=1,NMC
         WIMB0(I)=WIMB(I)
 210     CONTINUE
         DO 220 I=1,NEC
         EIMB0(I)=EIMB(I)
 220     CONTINUE
         DO 230 I=1,NPC
         PIMB0(I)=PIMB(I)
 230     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
 240     CONTINUE
         WZIMB0=WZIMB
         OFIMB0=OFIMB
C
      ELSE
      ENDIF
C
      RETURN
      END
C
C
C*****************************************************************
      SUBROUTINE PROP (MAT,PRSI,TMPI,OF,CEFF,ZENTH,ZENTR,ZDENS)
C
C     PROP - PROPERTY PROGRAM CALCULATING HYDROGEN,
C              OXYGEN, STEAM AND HOT GAS PROPERTIES
C
      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)
       COMMON /H2OPRP/
     * H2OP1(7),H2OT1(13),H2OH1(7,13),H2OS1(7,13),H2OD1(7,13)
C
       COMMON /TABLE/
     * NH2P(4),NH2T(4),NO2P(3),NO2T(3),NH2OP(1),NH2OT(1)
C
       DIMENSION
     * TSH2(11),PSH2(11),HLH2(11),HVH2(11),SLH2(11),SVH2(11),
     * DLH2(11),DVH2(11),
     * TSO2(16),PSO2(16),HLO2(16),HVO2(16),SLO2(16),SVO2(16),
     * DLO2(16),DVO2(16)
C
C      TSH2  -  H2 SATURATION TEMPERATURE
C      PSH2  -  H2 SATURATION PRESSURE
C      HLH2  -  H2 SATURATION ENTHALPY - LIQUID
C      HVH2  -  H2 SATURATION ENTHALPY - VAPOR
C      SLH2  -  H2 SATURATION ENTROPY  - LIQUID
C      SVH2  -  H2 SATURATION ENTROPY  - VAPOR
C      DLH2  -  H2 SATURATION DENSITY  - LIQUID
C      DVH2  -  H2 SATURATION DENSITY  - VAPOR
C
           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/
C
           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/
C
           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/
C
           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/
C
           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/
C
           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/
C
           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/
C
           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/
C
C      TSO2  -  O2 SATURATION TEMPERATURE
C      PSO2  -  O2 SATURATION PRESSURE
C      HLO2  -  O2 SATURATION ENTHALPY - LIQUID
C      HVO2  -  O2 SATURATION ENTHALPY - VAPOR
C      SLO2  -  O2 SATURATION ENTROPY  - LIQUID
C      SVO2  -  O2 SATURATION ENTROPY  - VAPOR
C      DLO2  -  O2 SATURATION DENSITY  - LIQUID
C      DVO2  -  O2 SATURATION DENSITY  - VAPOR
C
           DATA (TSO2(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/
C
           DATA (PSO2(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/
C
           DATA (HLO2(J),J=1,16)/
     * -58.356,-56.730,-55.096,-53.455,-51.804,-50.144,-48.473,
     * -46.790,-45.093,-43.380,-41.650,-39.901,-38.130,-36.334,
     * -34.511,-32.657/
C
           DATA (HVO2(J),J=1,16)/
     * 33.777,34.457,35.110,35.734,36.326,36.884,37.408,37.894,
```

```
     * 38.340,38.745,39.105,39.419,39.683,39.894,40.049,40.144/
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
   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')
C
C     **  INTERPOLATE RESULTS FROM SINGLE ARRAY  **
C
      IPRP=0
      NPX1=2
      NPY1=2
      ZENTH=0.0
      ZENTR=0.0
      ZDENS=0.0
C
      GO TO (10,20,30,40) MAT
C
   10 IF(TMPI.GT.  30.0.AND.TMPI.LT.  50.0) IPRP=1
      IF(TMPI.GT.  70.0.AND.TMPI.LT. 110.0) IPRP=2
      IF(TMPI.GT. 240.0.AND.TMPI.LT. 720.0) IPRP=3
      IF(TMPI.GT.1400.0.AND.TMPI.LT.2000.0) IPRP=4
      GO TO (11,12,13,14) IPRP
C
   11 IF(PRSI.LT.  20.0.OR.PRSI.GT. 370.0) GO TO 50
      CALL PRPSAT(PRSI,TMPI,ZENTH,
     *  TSH2(11),NH2P(1),NH2T(1),11,29.95,50.05,
     *  H2P1,H2T1,H2H1,PSH2,TSH2,HLH2,HVH2)
      CALL PRPSAT(PRSI,TMPI,ZENTR,
     *  TSH2(11),NH2P(1),NH2T(1),11,29.95,50.05,
     *  H2P1,H2T1,H2S1,PSH2,TSH2,SLH2,SVH2)
      CALL PRPSAT(PRSI,TMPI,ZDENS,
     *  TSH2(11),NH2P(1),NH2T(1),11,29.95,50.05,
     *  H2P1,H2T1,H2D1,PSH2,TSH2,DLH2,DVH2)
      RETURN
C
   12 IF(PRSI.LT.3400.0.OR.PRSI.GT.7200.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2P2,H2T2,H2H2,
     *  NH2P(2),NH2T(2),NPX1,NPY1,NH2P(2),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2P2,H2T2,H2S2,
     *  NH2P(2),NH2T(2),NPX1,NPY1,NH2P(2),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,H2P2,H2T2,H2D2,
     *  NH2P(2),NH2T(2),NPX1,NPY1,NH2P(2),ZDENS,N1)
      RETURN
C
   13 IF(PRSI.LT.1400.0.OR.PRSI.GT.7000.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2P3,H2T3,H2H3,
     *  NH2P(3),NH2T(3),NPX1,NPY1,NH2P(3),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2P3,H2T3,H2S3,
     *  NH2P(3),NH2T(3),NPX1,NPY1,NH2P(3),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,H2P3,H2T3,H2D3,
     *  NH2P(3),NH2T(3),NPX1,NPY1,NH2P(3),ZDENS,N1)
      RETURN
C
   14 IF(PRSI.LT.1400.0.OR.PRSI.GT.5800.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2P4,H2T4,H2H4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2P4,H2T4,H2S4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),ZENTR,N1)
```

```
      CALL ITERP2(PRSI,TMPI,H2P4,H2T4,H2D4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),ZDENS,N1)
      RETURN
C
C
   20 IF(TMPI.GT. 160.0.AND.TMPI.LT. 240.0) IPRP=1
      IF(IPRP.EQ.1.AND.PRSI.LT.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
C
   21 IF(PRSI.LT.  30.0.OR.PRSI.GT. 630.0) GO TO 50
      IF(TMPI.LT. 160.0.OR.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
C
   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
C
   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
C
C
   30 IF(TMPI.LT.1400.0.OR.TMPI.GT.2000.0) GO TO 50
      IF(PRSI.LT. 100.0.OR.PRSI.GT. 700.0) GO TO 50
      CALL ITERP2(PRSI,TMPI,H2OP1,H2OT1,H2OH1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),ZENTH,N1)
      CALL ITERP2(PRSI,TMPI,H2OP1,H2OT1,H2OS1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),ZENTR,N1)
      CALL ITERP2(PRSI,TMPI,H2OP1,H2OT1,H2OD1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),ZDENS,N1)
      RETURN
C
C
C 40  CALL PRPMIX(PRSI,TMPI,OF,CEFF,HMIX,SMIX)
C ************ MODIFIED 2/19/93 **********
   40 CALL PRPMIX(PRSI,TMPI,OF,CEFF,HMIX,SMIX,DMIX)
      ZENTH=HMIX
      ZENTR=SMIX
C     ZDENS=0.0
C ************ MODIFIED 2/19/93 ************
      ZDENS=DMIX
      RETURN
C
C
   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
C
      END
C*****************************************************************
C     SUBROUTINE PRPMIX (P,TMPI,OF,CEFF,HMIX,SMIX)
C ************ MODIFIED 2/19/93 ************
      SUBROUTINE PRPMIX (P,TMPI,OF,CEFF,HMIX,SMIX,DMIX)
C
C     PRPMIX - CALCULATES HOT GAS MIXTURE PROPERTIES.
C
```

```
      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)
      COMMON /H2OPRP/
     * H2OP1(7),H2OT1(13),H2OH1(7,13),H2OS1(7,13),H2OD1(7,13)
C
      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
C
C
      XMWH2  =     2.0160
      XMWO2  =    31.9988
      XMWH2O =    18.0153
      HCOMB  = -6825.6550
C
      NPX1 = 2
      NPY1 = 2
      ITST1= 0
      ITST2= 0
      ITST3= 0
      ITST4= 0
      ITST5= 0
      ITST6= 0
C
      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
C
      EH2  = XH2  / XMWH2
      EH2O = XH2O / XMWH2O
      EO2  = XO2  / XMWO2
      ET   = EH2 + EH2O + EO2
C
      YH2  = EH2  / ET
      YH2O = EH2O / ET
      YO2  = 1.0 - YH2 - YH2O
C
      PH2  = P * YH2
      PH2O = P * YH2O
      PO2  = P * YO2
C
      IF(TMPI.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,TMPI,H2P4,H2T4,H2H4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),HH2,N1)
      CALL ITERP2(PH2,TMPI,H2P4,H2T4,H2S4,
     *  NH2P(4),NH2T(4),NPX1,NPY1,NH2P(4),SH2,N1)
C
      IF(TMPI.LT.1400.0.OR.TMPI.GT.2000.0) ITST3=1
      IF(PH2O.LT. 100.0.OR.PH2O.GT. 700.0) ITST4=1
      CALL ITERP2(PH2O,TMPI,H2OP1,H2OT1,H2OH1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),HH2O,N1)
      CALL ITERP2(PH2O,TMPI,H2OP1,H2OT1,H2OS1,
     *  NH2OP(1),NH2OT(1),NPX1,NPY1,NH2OP(1),SH2O,N1)
C
      IF(YO2.LT.0.001) THEN
       DHO2 = 0.0
       DSO2 = 0.0
      ELSE
      IF(TMPI.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,TMPI,O2P3,O2T3,O2H3,
     *  NO2P(3),NO2T(3),NPX1,NPY1,NO2P(3),HO2,N1)
      CALL ITERP2(PO2,TMPI,O2P3,O2T3,O2S3,
     *  NO2P(3),NO2T(3),NPX1,NPY1,NO2P(3),SO2,N1)
       DHO2 = HO2 - HO2REF
       DSO2 = SO2 - SO2REF + SO2A
```

```
      ENDIF
C
  10  DHH2   = HH2 - HH2REF
      DHH2OM = (HH2O - HWAREF) + HCOMB
      DSH2   = SH2 - SH2REF + SH2A
      DSH2O  = SH2O - SWAREF + SWAA
C
      HMIX   = XH2*DHH2 + XH2O*DHH2OM + XO2*DHO2
      SMIX   = XH2*DSH2 + XH2O*DSH2O  + XO2*DSO2
C
C ************ MODIFIED 2/19/93 ************
      XMWMIX = XMWH2O*YH2O + XMWH2*YH2 + XMWO2*YO2
      DMIX   = 144.*XMWMIX*P/(1545.3*TMPI)
C
      IF (ITST1.EQ.1.OR.ITST2.EQ.1) WRITE(21,51) PH2,TMPI
      IF (ITST3.EQ.1.OR.ITST4.EQ.1) WRITE(21,52) PH2O,TMPI
      IF (ITST5.EQ.1.OR.ITST6.EQ.1) WRITE(21,53) PO2,TMPI
C
  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 PH2O PRS > ',F7.2,2X,
     *  'AND TMP > ',F7.2,2X,'FOR "H2O" IS OUT OF RANGE')
  53  FORMAT(/3X,'PRPMIX - REQUESTED  PO2 PRS > ',F7.2,2X,
     *  'AND TMP > ',F7.2,2X,'FOR " O2" IS OUT OF RANGE')
C
      RETURN
      END
C*****************************************************************
      SUBROUTINE PRPSAT (X,Y,FPROP,TCRT,NX1,NY1,NX2,YL,YH,
     *  PRS1,TMP1,PROP,PRS2,TMP2,PROPL,PROPV)
C
C     PRPSAT - CALCULATES NBS PROPERTIES NEAR SATURATION CURVE
C
      DIMENSION PRS1(1),TMP1(1)
C
      NR1=NX1
      NPX1=2
      NPY1=2
      NPX2=2
C
      ZPLGAS=0.0
      ZPHGAS=0.0
      ZPLLIQ=0.0
      ZPHLIQ=0.0
      ZPROP1=0.0
      ZPROP=0.0
      FPROP=0.0
      ZTSAT=0.0
      ARGA=0.0
      ARGB=0.0
      ZTSATT=0.0
C
C
      CALL ITERP2(X,Y,PRS1,TMP1,PROP,NX1,NY1,NPX1,NPY1,NR1,ZPROP1,N1)
      FPROP=ZPROP1
      IF(Y.GT.TCRT) GO TO 70
      CALL ITERP1(X,PRS2,TMP2,NX2,NPX2,ZTSAT,N2)
      IF(Y.LT.ZTSAT) GO TO 61
C
C        * *  GAS CALCULATIONS   * *
C
      CALL ITERP1(X,PRS2,PROPV,NX2,NPX2,ZPGAS,N2)
      CALL ITERP2(X,YH,PRS1,TMP1,PROP,NX1,NY1,NPX1,NPY1,NR1,ZTST,N1)
      DTST=ZTST-ZPGAS
      IF(DTST.GT.0.0001) GO TO 50
      ZPLGAS=ZPGAS
      IF(ZPROP1.LT.ZPGAS) GO TO 70
      GO TO 51
   50 ZPHGAS=ZPGAS
      IF(ZPROP1.GT.ZPGAS) GO TO 70
C
   51 LPR=1
   53 PRSD=PRS1(LPR)-0.0001
      IF(PRSD.GT.X) GO TO 52
      LPR=LPR+1
      GO TO 53
C
   52 ARGA=PRS1(LPR)
```

```
      CALL ITERP1(ARGA,PRS2,TMP2,NX2,NPX2,ZTSATT,N2)
C
      LTP=1
   54 TMPD=TMP1(LTP)-0.0001
      IF(TMPD.GT.ZTSATT) GO TO 55
      LTP=LTP+1
      GO TO 54
C
   55 ARGB=TMP1(LTP)
      YY=ARGB
      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)*((ARGB-Y)/(ARGB-ZTSAT))
      FPROP=ZPROP
C
      GO TO 70
C
C        * *  LIQ CALCULATIONS  * *
C
   61 CALL ITERP1(X,PRS2,PROPL,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
      ZPLLIQ=ZPLIQ
      IF(ZPROP1.LT.ZPLIQ) GO TO 70
      GO TO 60
   59 ZPHLIQ=ZPLIQ
      IF(ZPROP1.GT.ZPLIQ) GO TO 70
C
   60 LPR=1
   63 PRSD=PRS1(LPR)-0.0001
      IF(PRSD.GT.X) GO TO 62
      LPR=LPR+1
      GO TO 63
C
   62 ARGA=PRS1(LPR-1)
      CALL ITERP1(ARGA,PRS2,TMP2,NX2,NPX2,ZTSATT,N2)
C
      LTP=1
   64 TMPD=TMP1(LTP)-0.0001
      IF(TMPD.GT.ZTSATT) GO TO 65
      LTP=LTP+1
      GO TO 64
C
   65 ARGB=TMP1(LTP-1)
      YY=ARGB
      IF(DTST.GT.0.0001) CALL ITERP2(X,YY,PRS1,TMP1,PROP,NX1,NY1,
     *  NPX1,NPY1,NR1,ZPLLIQ,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-ARGB))
      FPROP=ZPROP
C
   70 CONTINUE
C
      RETURN
      END
C*******************************************************************
      SUBROUTINE ITERP1 (X,XT,YT,NX,NPX,Y,NERR)
C
C     ITERP1 - SINGLE INTERPOLATION ROUTINE.
C
      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
```

```
  22  Y=YT(I)
      GO TO 999
  30  I=NX
      IF(XT(I)-X)13,20,40
  40  N1=1
      N2=NX
  45  MP=(N1+N2)/2
  50  IF(XT(MP)-X)52,54,56
  52  N1=MP
      GO TO 60
  54  I=MP
      GO TO 20
  56  N2=MP
  60  IF((N2-N1) .NE. 1) GO TO 45
C
      IF (N2.GT.(IH+1)) GO TO 65
      I=IH+1
      GO TO 70
  65  I=N2
C
      IF(N2 .GT. I) I=N2
  70  K=I-IH
      N=K+NP-1
      Y=0.
      IF(N-NX)90,90,80
  80  N=NX
      K=NX-NP+1
  90  DO 120 J=K, N
      P=1.0
      DO 110 I=K, N
      IF(I-J)100,110,100
 100  P=P*(X-XT(I))/(XT(J)-XT(I))
 110  CONTINUE
      Y=Y+YT(J)*P
 120  CONTINUE
      GO TO 999
      ENTRY ENTERP (X,XT,YT,Y)
      Y=0.
      GO TO (90,22),INTER
 999  CONTINUE
      RETURN
      END
C*******************************************************************
      SUBROUTINE ITERP2 (X,Y,XT,YT,ZT,NX,NY,NPX,NPY,NR,Z,NERR)
C
C     ITERP2 - DOUBLE INTERPOLATION ROUTINE.
C
      DIMENSION XT(1),YT(1),ZT(NR,1),ZC(15)
      NERRB=0
      NPYY=NPY
      IF(NY .LT. NPY) NPYY=NY
      IH=NPYY/2
      I=1
      IF(YT(I)-Y)30,20,10
   10 IH=0
   12 NERRB=201
      GO TO 70
   13 NERRB=204
      GO TO 70
   20 CALL ITERP1(X,XT,ZT(1,I),NX,NPX,Z,NERRA)
      GO TO 999
   30 I=NY
      IF(YT(I)-Y)13,20,40
   40 N1=1
      N2=NY
   45 MP=(N1+N2)/2
   50 IF(YT(MP)-Y)52,54,56
   52 N1=MP
      GO TO 60
   54 I=MP
      GO TO 20
   56 N2=MP
   60 IF((N2-N1) .NE. 1) GO TO 45
      I=N2
      IF(I .LT. (IH+1)) I=IH+1
   70 K=I-IH
      N=K+NPYY-1
      IF(N-NY)90,90,80
```

```
   80 N=NY
      K=NY-NPYY+1
   90 J=0
      DO 100 I=K, N
      J=J+1
      IF(J .NE. 1) GO TO 95
      CALL ITERP1(X,XT,ZT(1,I),NX,NPX,ZC(J),NERRA)
      GO TO 100
   95 CALL ENTERP(X,XT,ZT(1,I),ZC(J))
  100 CONTINUE
      CALL ITERP1(Y,YT(K),ZC,NPYY,NPYY,Z,NERRC)
  999 NERR=NERRA+NERRB
      RETURN
      END
C
C
      SUBROUTINE OPT (N,NMAX,EPSC,EPST,RHO,ALPHA,BETA,
     1                ISTAGE,F2,GNORM,DXNORM,X,X2)
      DIMENSION X(25),X2(25),G(25),G2(25),DX(25),H(25,25)
C
C     ISTAGE=STAGE COUNTER
      ISTAGE=1
      IFUNC=0
C
      CALL OBJF (IFUNC,X,F)
      CALL GRAD (IFUNC,N,X,G,F)
C
C     SET INITIAL METRIC H TO THE IDENTITY MATRIX
   10 DO 12 I=1,N
      DO 12 J=1,N
      IF (J.NE.I) THEN
          H(I,J)=0.0
      ELSE
          H(I,J)=1.0
      ENDIF
   12 CONTINUE
C
C     KC=STEP DIRECTION PARAMETER
C        KC=1  NEGATIVE GRADIENT STEP DIRECTION
C        KC=0  QUASI-NEWTON STEP DIRECTION
C
      KC=1
C
C     CALCULATE SEARCH DIRECTION VECTOR DX=-H*G
   15 DO 20 I=1,N
      DX(I)=0.0
      DO 18 J=1,N
   18 DX(I)=DX(I)-H(I,J)*G(J)
   20 CONTINUE
C
C     CALL UNIVARIATE SEARCH ROUTIVE
   70 CALL ARMIJO (IFUNC,IFLAG,N,RHO,ALPHA,BETA,DMAX,
     1             DX,X,X2,F,F2,G)
      CALL GRAD (IFUNC,N,X2,G2,F2)
C
      DO 50 I=1,N
   50 DXDG=DXDG+(X2(I)-X(I))*(G2(I)-G(I))
C
      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
C
C     GRADIENT NORM**2 < EPSC IMPLIES CONVERGENCE
   90 GNORM=0.0
      DO 92 I=1,N
   92 GNORM=GNORM+G2(I)**2
      IF (GNORM-EPSC) 200,110,110
C
C     DELTA X NORM**2 < EPST IMPLIES TERMINATION
  110 DXNORM=0.0
      DO 112 I=1,N
  112 DXNORM=DXNORM+(X2(I)-X(I))**2
      IF (DXNORM-EPST) 200,120,120
C
C     INCREMENT STAGE COUNTER
  120 ISTAGE=ISTAGE+1
C
```

```
C     CHECK THAT MAXIMUM NUMBER OF STAGES NOT EXCEEDED
      IF (ISTAGE .GT. NMAX) GO TO 200
C
C     UPDATE METRIC H
      CALL BFGS (IFLAG,N,X,X2,G,G2,H)
      IF (IFLAG.EQ.1) GO TO 76
      KC=0
C
C     REINITIALIZE
      F=F2
      DO 125 I=1,N
      X(I)=X2(I)
  125 G(I)=G2(I)
C
      GO TO 15
C
  200 RETURN
C
      END
C
C
      SUBROUTINE ARMIJO (IFUNC,IFLAG,N,RHO,ALPHA,BETA,DMAX,
     1                   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
C
      DO 110 I=1,N
  110 GDX=GDX+G(I)*DX(I)
C
      RATIO=0.0
      DO 111 I=1,N
      RATIO2=ABS(DX(I)/X(I))
  111 RATIO=MAX(RATIO,RATIO2)
C
      RMU=RHO
      SCALE=RMU*RATIO
      IF (SCALE .GT. DMAX) RMU=DMAX/RATIO
C
  112 DO 115 I=1,N
  115 X2(I)=X(I)+RMU*DX(I)
C
      CALL OBJF (IFUNC,X2,F2)
      TBAR=F2-F-RMU*ALPHA*GDX
      IF (TBAR .GT. 0.0) GO TO 120
      IF (F-F2) 180,180,200
  120 RMU=RMU*BETA
      ICOUNT=ICOUNT+1
      IF (ICOUNT .LE. 12) GO TO 112
  180 IFLAG=1
  200 RETURN
      END
C
C
      SUBROUTINE BFGS (IFLAG,N,X,X2,G,G2,H)
      DIMENSION X(25),X2(25),G(25),G2(25),DX(25),DG(25),HDG(25),
     1          V(25),H(25,25)
      IFLAG=0
C
C     CALL DFP METRIC UPDATE
      CALL DFP (N,DXDG,DGHDG,X,X2,G,G2,DX,DG,HDG,H)
C
      IF (DGHDG .GT. 0.0) GO TO 10
      IFLAG=1
      GO TO 400
C
C     DETERMINE BFGS METRIC UPDATE
   10 DO 310 I=1,N
  310 V(I)=DGHDG**0.5*(DX(I)/DXDG-HDG(I)/DGHDG)
      DO 320 I=1,N
      DO 320 J=1,N
  320 H(I,J)=V(I)*V(J)+H(I,J)
C
  400 RETURN
      END
C
```

```
C
      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),
     1          DGH(25),H(25,25)
      DO 100 I=1,N
      DX(I)=X2(I)-X(I)
 100  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)
 400  DGH(I)=DGH(I)+DG(J)*H(J,I)
      DXDG=DXDG+DX(I)*DG(I)
 410  DGHDG=DGHDG+DGH(I)*DG(I)
      DO 420 I=1,N
      DO 420 J=1,N
 420  H(I,J)=H(I,J)+DX(I)*DX(J)/DXDG-HDG(I)*DGH(J)/DGHDG
      RETURN
      END
C
C
      SUBROUTINE GRAD (IFUNC,N,X,G,F)
      DIMENSION X(25),X2(25),G(25)
      DX=0.001
      DO 10 I=1,N
  10  X2(I)=X(I)
      DO 20 I=1,N
      X2(I)=X(I)+DX
      CALL OBJF (IFUNC,X2,F2)
      G(I)=(F2-F)/DX
  20  X2(I)=X(I)
      RETURN
      END
C
C
C
      SUBROUTINE TPIMB (I,ICURA,ICURB,DIA,W,SPEED,D1,D2,P1,P2,T1,T2,
     1                  H1,H2,ZIMB1,ZIMB2)
C
C ***** TURBINE CURVES REQUIRE ICURA & ICURB < 20 ******************
C
      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-P1/PRATIO
         DPCAL=P1-P2
         ZIMB1=DPCAL-DPCHR
         DTCHR=T1-(1.0-TRATIO)*T1
         DTCAL=T1-T2
         ZIMB2=DTCAL-DTCHR
      ELSE
C
C ***** PUMP CURVES REQUIRE ICURA & ICURB >= 20 ********************
C
         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
         PWR=W*(H2-H1)
         PWRID=(144.0/778.16)*(W*(P2-P1)/D1)
         EFFCAL=PWRID/PWR
         ZIMB2=(EFFCAL-EFFCHR)*PWR
      ENDIF
C
  10  CONTINUE
C
      RETURN
      END
```