DYGABCD - A Program for Calculating Linear A, B, C, and D Matrices From a Nonlinear Dynamic Engine Simulation

Lucille C. Geyser

SEPTEMBER 1978
DYGABCD - A Program for Calculating Linear A, B, C, and D Matrices From a Nonlinear Dynamic Engine Simulation

Lucille C. Geyser
Lewis Research Center
Cleveland, Ohio
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>ANALYTICAL BACKGROUND</td>
<td>2</td>
</tr>
<tr>
<td>Definitions of A, B, C, and D Matrices</td>
<td>5</td>
</tr>
<tr>
<td>Solutions for Full Forms of A, B, C, and D Matrices</td>
<td>8</td>
</tr>
<tr>
<td>Solutions for Reduced Forms of A, B, C, and D Matrices</td>
<td>10</td>
</tr>
<tr>
<td>USERS GUIDE</td>
<td>13</td>
</tr>
<tr>
<td>Changes to DYNGEN</td>
<td>13</td>
</tr>
<tr>
<td>Method</td>
<td>14</td>
</tr>
<tr>
<td>Changes to Subroutine DISTRB</td>
<td>15</td>
</tr>
<tr>
<td>Changes to Function DERIV</td>
<td>16</td>
</tr>
<tr>
<td>Changes to Subroutine PUTIN</td>
<td>17</td>
</tr>
<tr>
<td>Changes to Subroutine ENGBAL</td>
<td>17</td>
</tr>
<tr>
<td>Changes to Main Routine DYGABCD</td>
<td>17</td>
</tr>
<tr>
<td>SAMPLE CASE</td>
<td>17</td>
</tr>
<tr>
<td>CONCLUSIONS</td>
<td>18</td>
</tr>
<tr>
<td>APPENDIXES</td>
<td>19-20</td>
</tr>
<tr>
<td>A - FORTRAN SYMBOLS</td>
<td>20</td>
</tr>
<tr>
<td>B - FORTRAN LISTINGS</td>
<td>40</td>
</tr>
<tr>
<td>C - SAMPLE CASE INPUT AND OUTPUT</td>
<td>136</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>160</td>
</tr>
</tbody>
</table>
DYGABCD – A PROGRAM FOR CALCULATING LINEAR A, B, C, AND D MATRICES FROM A NONLINEAR DYNAMIC ENGINE SIMULATION

by Lucille C. Geyser

Lewis Research Center

SUMMARY

A digital computer program, DYGABCD, has been developed that will generate linearized, dynamic models of simulated turbofan and turbojet engines. DYGABCD is based on an earlier computer program, DYNGEN, that is capable of calculating simulated nonlinear steady-state and transient performance of one- and two-spool turbojet engines or two- and three-spool turbofan engines.

Most control design techniques require linear system descriptions. For multiple-input/multiple-output systems such as turbine engines, state space matrix descriptions of the system are often desirable. DYGABCD computes the state space matrices, commonly referred to as the A, B, C, and D matrices, required for a linear system description. The report discusses the analytical approach and provides a users manual, FORTRAN listings, and a sample case. NASA TN D-7901, describing DYNGEN, is a necessary adjunct to this report.

INTRODUCTION

Digital computers are more frequently being used to accurately simulate the steady-state and transient performance characteristics of turbojet and turbofan engines. These simulations are capable of modeling these engines over the complete range of power settings and flight conditions. DYNGEN (ref. 1) is one such engine simulation that is generalized to permit the user to model many different engine configurations. It is based on the generalized steady-state engine modeling codes GENENG (ref. 2) and GENENG II (ref. 3). These codes are based on a technique called SMOTE (refs. 4 and 5).
To accomplish the modeling accuracy being demonstrated by DYNGEN and other digital computer engine simulation codes, nonlinear steady-state and transient physical relationships must be included. Most of the analytical design approaches used by control designers, however, require linear descriptions of the steady-state and transient characteristics of the engine. Linear approximations to nonlinear systems can be obtained at a particular operating point (a particular equilibrium condition of power setting and flight condition) for small excursions about that particular point.

For a system as complex as an engine, this linear system description usually consists of a multitude of interdependent linear algebraic and differential equations that can be written in a compact form by using matrix notation. A widely accepted linear dynamic system representation (ref. 6) uses a system matrix A, a control matrix B, an output matrix C, and a feed-forward matrix D. To describe an engine on a linearized basis over its complete operating envelope, however, a whole family of suitably selected linear approximations would be required.

This report, then, describes DYGABCD, a digital computer program that calculates the A, B, C, and D matrices for any turbofan or turbojet engine that is capable of being simulated with the DYNGEN program. DYGABCD can be used (1) exactly as DYNGEN to generate engine steady-state and transient performance characteristics; or (2) to generate the linear model A, B, C, and D matrices. The method of calculation employed in DYGABCD to generate these matrices can easily be used in any transient simulation that uses a backward difference integration technique (ref. 2).

The report first presents (1) the theoretical or analytical bases for linear model approximations to nonlinear systems, (2) a definition of the compact matrix formulation, and (3) a derivation of the equations necessary for determining matrix element values from the transient simulation. Next presented is a users guide for DYGABCD, including complete descriptions of those subroutines that have computational techniques different from DYNGEN. The FORTRAN symbols used are defined in appendix A, complete listings are shown in appendix B, and a sample case is presented in appendix C.

DYGABCD is written in FORTRAN IV and was used on the IBM 360/67 with the time-sharing system (TSS). The program should run equally well with the operating system (OS) on an IBM 360 or 370 series computer.

**ANALYTICAL BACKGROUND**

The simulation of a complex turbine engine process involves both linear and nonlinear algebraic and differential equations. Multiple equations of this type can generally be expressed in compact vector notation as
\[ \dot{x} = f(x, u) \]  \hspace{1cm} (1) \\
\[ y = g(x, u) \]  \hspace{1cm} (2)

where \( x \) is a vector of the time-varying states of the engine, \( u \) is a vector of the system control inputs to the engine, and \( y \) is a vector of the engine's outputs. The vector \( \dot{x} \) represents the time rate of change of each of the engine's states. The following table lists some typical states, inputs, and outputs for a turbine engine system:

<table>
<thead>
<tr>
<th>States</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperatures</td>
<td>Fuel flows</td>
<td>Engine airflows</td>
</tr>
<tr>
<td>Pressures</td>
<td>Nozzle area</td>
<td>Thrust</td>
</tr>
<tr>
<td>Speeds</td>
<td>Bleeds</td>
<td>Specific fuel consumption</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fan surge margin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compressor surge margin</td>
</tr>
</tbody>
</table>

Since most control design techniques require a linear description of the process to be controlled, a linear approximation to the nonlinear system needs to be developed. At a particular operating point, equations (1) and (2) can be represented by a Taylor series expansion about that point. If the vectors \( x \) and \( u \) are assumed to be single elements (i.e., scalars), equation (1) may be written as follows:

\[ \dot{x}_0 = f(x_0, u_0) \]  \hspace{1cm} (3)

The Taylor series expansion about the point \((x_0, u_0)\) is
\[ \dot{x} = f(x_0, u_0) + \frac{\partial f}{\partial x} \bigg|_{x=x_0, u=u_0} (x - x_0) + \frac{\partial f}{\partial u} \bigg|_{x=x_0, u=u_0} (u - u_0) \]

\[ + \frac{\partial^2 f}{\partial x^2} \bigg|_{x=x_0, u=u_0} (x - x_0)^2 + \frac{\partial^2 f}{\partial u^2} \bigg|_{x=x_0} (u - u_0)^2 \]

\[ + \text{Other higher order terms} \]  

(4)

If \((x = x_0, \ u = u_0)\) is a steady-state operating point, the system is in equilibrium and hence \(\dot{x}_0 = 0\). Thus, by using equations (1) and (3), equation (4) can be rewritten as

\[ \dot{x} = \frac{\partial \dot{x}}{\partial x} \bigg|_{\text{op}} (x - x_0) + \frac{\partial \dot{x}}{\partial u} \bigg|_{\text{op}} (u - u_0) \]

\[ + \frac{\partial^2 \dot{x}}{\partial x^2} \bigg|_{\text{op}} (x - x_0)^2 + \frac{\partial^2 \dot{x}}{\partial u^2} \bigg|_{\text{op}} (u - u_0)^2 \]

\[ + \text{Other higher order terms} \]  

(5)

where \(\text{op}\) denotes operating point.

In a suitable neighborhood of the steady-state operating point \((x_0, u_0)\), the higher order terms of equation (4) are negligible, and equation (1) can be approximated by using only the first partial derivative terms. By letting \(\Delta x = x - x_0\), \(\Delta u = u - u_0\), and \(\Delta \dot{x} = \dot{x} - \dot{x}_0\), equation (5) becomes the linear equation

\[ \Delta \dot{x} = \frac{\partial \dot{x}}{\partial x} \bigg|_{\text{op}} \Delta x + \frac{\partial \dot{x}}{\partial u} \bigg|_{\text{op}} \Delta u \]  

(6)

Equation (2) may be linearized in similar fashion for the output \(y\), resulting in

\[ \Delta y = \frac{\partial y}{\partial x} \bigg|_{\text{op}} \Delta x + \frac{\partial y}{\partial u} \bigg|_{\text{op}} \Delta u \]  

(7)
Definitions of A, B, C, and D Matrices

An actual system with a multitude of states, inputs, and outputs, when linearized about an operating point, can be described by the following linear vector-matrix equations:

\[ \Delta \dot{x} = A \Delta x + B \Delta u \]  \hspace{1cm} (8)

\[ \Delta y = C \Delta x + D \Delta u \]  \hspace{1cm} (9)

where

\( \Delta x \)  \hspace{0.5cm} n-dimensional vector of state deviations
\( \Delta u \)  \hspace{0.5cm} c-dimensional vector of system control input deviations
\( \Delta y \)  \hspace{0.5cm} o-dimensional vector of system output deviations
\( A \)  \hspace{0.5cm} n-by n-dimensional state system matrix
\( B \)  \hspace{0.5cm} n-by c-dimensional control input matrix
\( C \)  \hspace{0.5cm} o-by n-dimensional output matrix
\( D \)  \hspace{0.5cm} o-by c-dimensional feed-forward matrix
\( n \)  \hspace{0.5cm} number of states
\( c \)  \hspace{0.5cm} number of control inputs
\( o \)  \hspace{0.5cm} number of outputs

The system matrix A is a matrix of numbers that are the values of particular partial derivatives at a specific operating point. When the partial derivative notation of equations (6) and (7) is used, equations (10) to (13) show the elemental arrangements of the matrices A, B, C, and D.
\[
A = \begin{bmatrix}
\frac{\partial \dot{x}_1}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} & \cdots & \frac{\partial \dot{x}_1}{\partial x_n} \\
\frac{\partial \dot{x}_2}{\partial x_1} & \frac{\partial \dot{x}_2}{\partial x_2} & \cdots & \frac{\partial \dot{x}_2}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial \dot{x}_n}{\partial x_1} & \frac{\partial \dot{x}_n}{\partial x_2} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n}
\end{bmatrix}
\]

(10)

\[
B = \begin{bmatrix}
\frac{\partial \dot{x}_1}{\partial u_1} & \frac{\partial \dot{x}_1}{\partial u_2} & \cdots & \frac{\partial \dot{x}_1}{\partial u_c} \\
\frac{\partial \dot{x}_2}{\partial u_1} & \frac{\partial \dot{x}_2}{\partial u_2} & \cdots & \frac{\partial \dot{x}_2}{\partial u_c} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial \dot{x}_n}{\partial u_1} & \frac{\partial \dot{x}_n}{\partial u_2} & \cdots & \frac{\partial \dot{x}_n}{\partial u_c}
\end{bmatrix}
\]

(11)
\[ C = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix} \quad (12) \]

\[ D = \begin{bmatrix} \frac{\partial y_1}{\partial u_1} & \frac{\partial y_1}{\partial u_2} & \cdots & \frac{\partial y_1}{\partial u_n} \\ \frac{\partial y_2}{\partial u_1} & \frac{\partial y_2}{\partial u_2} & \cdots & \frac{\partial y_2}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial u_1} & \frac{\partial y_n}{\partial u_2} & \cdots & \frac{\partial y_n}{\partial u_n} \end{bmatrix} \quad (13) \]
Solutions for Full Forms of A, B, C, and D Matrices

It is our intent to take a DYNGEN digital computer simulation with its nonlinear dynamic equations and to develop a method for extracting from it the elements of the A, B, C, and D matrices at particular engine operating points.

Because of the backward finite-difference integration technique required by DYNGEN (SMOTE technique), it is necessary to calculate the inverse of the state system matrix A. To accomplish this, equation (8) can be rewritten as

\[ \Delta x = A^{-1} \Delta \dot{x} - A^{-1} B \Delta u \]  

(14)

To solve for \( A^{-1} \), all system control input deviations \( \Delta u \) from the operating point are set to zero. Thus,

\[ \Delta x = A^{-1} \Delta \dot{x} \]  

(15)

Each element of \( A^{-1} \) is then approximated as

\[ (A^{-1})_{ij} \approx \frac{\Delta x_i}{\Delta \dot{x}_j} \]  

(16)

where the state derivatives \( \dot{x}_j \) are perturbed, one at a time, as described in the following paragraph.

A steady-state vector of the states \( x \) is calculated. The state derivatives \( \dot{x} \) at this steady-state point are therefore zero. The first \( \dot{x} \) is set to some suitable value, \( \Delta \dot{x} \), with the rest of the \( \dot{x} \)'s held to zero. A new state vector \( x^* \) is calculated. The difference between \( x^* \) and \( x \) is the vector \( \Delta x \) associated with this first \( \Delta \dot{x} \). Each of the other \( \dot{x} \)'s is then set to some suitable value, \( \Delta \dot{x} \), one at a time, with all other \( \dot{x} \)'s set to zero each time. A new vector \( x^* \) and a new vector \( \Delta x \) are calculated for each \( \Delta \dot{x} \). A matrix of these \( \Delta x \) column vectors is formed. Each column vector is divided by the \( \Delta \dot{x} \) that produced it, thereby forming the \( A^{-1} \) matrix referred to in equation (16). The \( A^{-1} \) matrix is then inverted to obtain the \( A \) matrix.

Solve for the control matrix B as follows: Under equilibrium conditions, \( \dot{x} = 0 \), equation (8) becomes

\[ \Delta x = -A^{-1} B \Delta u \]  

(17)
Each element of $-A^{-1}B$ is then approximated as

$$
(-A^{-1}B)_{ij} \approx \frac{\Delta x_i}{\Delta u_j}
$$

where the system control inputs $u_j$ are perturbed, one at a time, as described in the following paragraph.

A steady-state vector of the states $x$ is calculated. The first system control input $u$ is changed by some suitable value $\Delta u$ with the remainder of the $u$'s held to their steady-state values. A new state vector $x^*$ is calculated. The difference between $x^*$ and $x$ is the vector $\Delta x$ associated with this first $\Delta u$. Each of the other $u$'s is then perturbed by some suitable $\Delta u$, one at a time, with all other $u$'s remaining constant each time. A new vector $x^*$ and a new vector $\Delta x$ are calculated for each $\Delta u$. A matrix of these $\Delta x$ column vectors is then formed. Each column vector is divided by the $\Delta u$ that produced it, thereby forming the $-A^{-1}B$ matrix approximated by equation (18). This matrix is then premultiplied by $-A$ to obtain the $B$ matrix.

To solve for the output matrix $C$, substitute the equation for $\Delta x$ (eq. (14)) in the equation for $\Delta y$ (eq. (9)), yielding

$$
\Delta y = CA^{-1} \Delta x - CA^{-1}B \Delta u + D \Delta u
$$

When the system control inputs are held fixed, equation (19) becomes

$$
\Delta y = CA^{-1} \Delta \dot{x}
$$

Each element of $CA^{-1}$ is then approximated as

$$
(CA^{-1})_{ij} \approx \frac{\Delta y_i}{\Delta \dot{x}_j}
$$

where the state derivatives $\dot{x}_j$ are perturbed one at a time, as was done in obtaining $A^{-1}$.

A steady-state output vector $y$ is calculated. The state derivatives $\dot{x}$ at this steady-state point are therefore zero. The first $\dot{x}$ is set to some suitable value, $\Delta \dot{x}$, with the rest of the $\dot{x}$'s held to zero. A new output vector $y^*$ is calculated. The difference between $y^*$ and $y$ is the vector $\Delta y$ associated with this first $\Delta \dot{x}$. Each of the other $\dot{x}$'s is then set to some suitable value $\Delta \dot{x}$, one at a time, with all other $\dot{x}$'s
set to zero each time. A new vector \( y^* \) and a new vector \( \Delta y \) are calculated for each \( \Delta x \). A matrix of these \( \Delta y \) column vectors is then formed. Each column vector is divided by the \( \Delta x \) that produced it, thereby forming the \( CA^{-1} \) matrix approximated by equation (21). This matrix is then postmultiplied by \( A \) to obtain the \( C \) matrix.

Solving equation (9) for the feed-forward matrix \( D \) yields the approximation

\[
D_{ij} \approx \frac{\Delta y_i}{\Delta u_j} - C \frac{\Delta x_i}{\Delta u_j}
\]  

(22)

where the system control inputs \( u_j \) are perturbed, one at a time, as described in the following paragraph.

Two steady-state vectors, \( x \) (the state vector) and \( y \) (the output vector), are calculated. The first system control input \( u \) is changed by some suitable value \( \Delta u \), with the remainder of the \( u \)'s held to their steady-state value. A new state vector \( x^* \) is calculated. And a new output vector \( y^* \) is calculated. The difference between \( x^* \) and \( x \) is the vector \( \Delta x \) associated with this first \( \Delta u \). The difference between \( y^* \) and \( y \) is the vector \( \Delta y \) associated with this first \( \Delta u \). Each of the other \( u \)'s is then perturbed by some suitable value \( \Delta u \), one at a time, with all other \( u \)'s remaining constant each time. A new vector \( x^* \) and a new vector \( \Delta x \) are calculated for each \( \Delta u \). A new vector \( y^* \) and a new vector \( \Delta y \) are also calculated for each \( \Delta u \). A matrix of the \( \Delta x \) column vectors is then formed, and each column vector is divided by the \( \Delta u \) that produced it. A matrix of the \( \Delta y \) column vectors is also formed, and each column vector is again divided by the \( \Delta u \) that produced it. By substituting these two matrices in equation (22), we can solve for \( D \).

Solutions for Reduced Forms of \( A \), \( B \), \( C \), and \( D \) Matrices

Once the complete linear model of the dynamic system is obtained, it may be desirable, for certain reasons, to reduce the complexity of the linear description. For example, the eigenvalues of the full system either may be beyond the frequency range of interest or may have little overall effect on the system dynamic characteristics. This reduction can be done by several methods.

By the method of Weinberg and Adams (ref. 7), equation (14) is rewritten in partitioned form as

Solutions for Reduced Forms of \( A \), \( B \), \( C \), and \( D \) Matrices
\[
\frac{\Delta x_1}{\Delta x_2} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} \frac{\Delta \hat{x}_1}{\Delta \hat{x}_2} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{21} \\ \hat{A}_{12} & \hat{A}_{22} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \Delta u
\] (23)

where

\( \Delta x_1 \) \text{ r-dimensional part of state deviation vector}

\( \Delta x_2 \) \text{ (n-r)-dimensioned part of state deviation vector}

\( \Delta u \) \text{ c-dimensional system control input deviation vector}

\( \hat{A}_{11} \) \text{ r- by r-dimensional upper left partition of } A^{-1} \text{ matrix}

\( \hat{A}_{12} \) \text{ r- by (n-r)-dimensioned upper right partition of } A^{-1} \text{ matrix}

\( \hat{A}_{21} \) \text{ (n-r)- by r-dimensional lower left partition of } A^{-1} \text{ matrix}

\( \hat{A}_{22} \) \text{ (n-r)- by (n-r)-dimensioned lower right partition of } A^{-1} \text{ matrix}

\( B_1 \) \text{ r- by c-dimensional part of control input matrix}

\( B_2 \) \text{ (n-r)- by c-dimensional part of control input matrix}

n \text{ total number of states}

r \text{ number of states desired}

c \text{ number of control inputs}

Before calculating the partitioned \( \hat{A} \) matrix, permute the order of the \( \Delta x \)'s and the corresponding \( \Delta \hat{x} \)'s, so that the \( \Delta x \)'s to be retained are all included in \( \Delta x_1 \). Those remaining make up the elements of \( \Delta x_2 \). The elements of \( \Delta x_2 \) are assumed to be negligible in determining the system transient response. Hence, setting \( \Delta \hat{x}_2 = 0 \) in equation (23) yields

\[
\Delta x_1 = \hat{A}_{11} \Delta \hat{x}_1 - \left( \hat{A}_{11} B_1 + \hat{A}_{12} B_2 \right) \Delta u \] (24)

Solving for \( \Delta \hat{x}_1 \) gives

\[
\Delta \hat{x}_1 = \left( \hat{A}_{11} \right)^{-1} \Delta x_1 + \left( \hat{A}_{11} \right)^{-1} \left( \hat{A}_{11} B_1 + \hat{A}_{12} B_2 \right) \Delta u \] (25)

or

\[
\Delta \hat{x}_1 = A_R \Delta x_1 + B_R \Delta u \] (26)
where

$$A_R = \left(\hat{A}_{11}\right)^{-1}$$  \hfill (27)

and

$$B_R = \left(\hat{A}_{11}\right)^{-1} \left(\hat{A}_{11}B_1 + \hat{A}_{12}B_2\right)$$  \hfill (28)

Note that it is necessary to compute only the upper left quadrant of the $A^{-1}$ matrix (the $r$-by-$r$ $\hat{A}_{11}$) and then to invert it to obtain $A_R$. The quantity $\hat{A}_{11}B_1 + \hat{A}_{12}B_2$ consists of the first $r$ rows of the $A^{-1}B$ matrix from equation (18). Thus, only the first $r$ rows of $-A^{-1}B$ need to be calculated. These rows are then premultiplied by $-A_R$ to obtain $B_R$.

In solving for a reduced $C$ matrix and the $D$ matrix used with the reduced $A$, $B$, and $C$ matrices, there is no simple discarding of rows and columns. Starting with equation (23), if $\Delta\hat{x}_2$ is zero, then

$$\Delta x_2 = A_{21} \Delta\hat{x}_1 - \left(\hat{A}_{21}B_1 + \hat{A}_{22}B_2\right)\Delta u$$  \hfill (29)

Substituting equations (25) and (27) into equation (29) yields

$$\Delta x_2 = \hat{A}_{21} A_R \Delta x_1 + \left(\hat{A}_{21}A_R\hat{A}_{12} - \hat{A}_{22}\right)B_2 \Delta u$$  \hfill (30)

The partitioned form of equation (9) is

$$\Delta y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \frac{\Delta x_1}{\Delta x_2} + D \Delta u$$  \hfill (31)

where

$\Delta y$  o-dimensioned output deviation vector
$\Delta x_1$  r-dimensioned part of state deviation vector
$\Delta x_2$  (n-r)-dimensioned part of state deviation vector
$\Delta u$  c-dimensioned system control input deviation vector
$C_1$  o- by r-dimensioned part of output matrix
Equation (31) yields

\[ \Delta y = C_1 \Delta x_1 + C_2 \Delta x_2 + D \Delta u \] (32)

Using equation (30) to eliminate \( \Delta x_2 \) gives

\[ \Delta y = C_1 \Delta x_1 + C_2 \hat{A}_{21} A_R \Delta x_1 + \left[ C_2 \left( \hat{A}_{21} A_R \hat{A}_{12} - \hat{A}_{22} \right) B_2 + D \right] \Delta u \] (33)

or

\[ \Delta y = C_R \Delta x_1 + D_R \Delta u \] (34)

where

\[ C_R = C_1 + C_2 \hat{A}_{21} A_R \] (35)

and

\[ D_R = C_2 \left( \hat{A}_{21} A_R \hat{A}_{12} - \hat{A}_{22} \right) B_2 + D \] (36)

**Users Guide**

Changes to DYNGEN

The IBM 360/67 computer with TSS was chosen because of the storage size of the DYGABCD program. Certain changes have been made in the conversion of DYNGEN to DYGABCD for accuracy and speed of calculation. All real variables are now double precision. Each convergence loop now uses a convergence criterion of TOLALL multiplied by some constant, where TOLALL is input in the namelist data as before. Although the number of passes through some loops has been increased, the relationships among the loops have been kept constant, so that the accuracy of the convergence criteria can be increased. All BLOCK DATA subroutines are combined into one subroutine with the named common COMMON/COMDAT/COMD(5423). All program variables are com-
bined into one named common COMMON/COMALL/COM(1062). Within each subroutine, only those variables actually referenced are equivalenced to either COMD and/or COM, with one exception. In the main routine DYGABCD, each variable referenced anywhere in the program is equivalenced, so that all may be seen at a glance.

The DYGABCD program provides a maximum of 23 possible derivative variables for use in calculating the A and C matrices. The specific number of possible variables for any particular engine depends on the configuration of that engine. In calculating the B matrix, only two control variables are used. More parameters could be chosen. In calculating the C matrix, any variables calculated in the program, including the states and/or inputs, may be used. Variables that are both states and outputs are used in calculating the D matrix.

Minor changes have been made in the main routine DYGABCD and the subroutines ENGBAL and PUTIN (underlined in the listings) to accommodate the addition of the A, B, C, and D matrix calculations. An addition has been made to function DERIV to allow either the original calculations of DYNGEN or the calculations necessary for the A and/or C matrices to be performed. Since DISTRB was already a user-written subroutine in DYNGEN, the bulk of the A, B, C, and D matrix calculations are done in it.

Method

DYGABCD may be run exactly as DYNGEN if DYNGEN data are used. It may also be used to generate the A, B, C, and D matrices when it is used with a different set of data and a new subroutine DISTRB that is modified by the user. To use the A, B, C, and D matrix capability, the following tasks must be performed:

(1) A steady-state point is run and the resulting main fuel flow noted. This is the steady-state point about which the small perturbations (generating A, B, C, and D) will be made. When the A, B, C, and D matrices are generated, they will define a linear model that is applicable in the region of this steady-state point.

(2) A new run is made with the addition of one data card that includes the following parameters: MODE=2, WFB=(the steady-state value noted previously in task 1), ITRAN=1, and IAMTRX=1.

IAMTRX is a new variable that is input as 1 if the A, B, C, and D matrices are to be calculated. The default value is 0, which is the value when DYGABCD is used as DYNGEN. If IAMTRX is 1, the program runs a "pseudo-transient" to calculate the A and C matrices, a set of steady-state points to calculate the B matrix, and a set of steady-state points plus the C matrix to calculate the D matrix.

The term "pseudo-transient" is used to describe the following procedure: since ITRAN = 1 on the data card in task 2, the program runs in the transient mode. However, during every time step, when the time index, NSTEP, is increased by 1 in sub-
routine ENGINE, it is promptly decreased by 1 back to 0 by the next statement, if
IAMTRX is 1. This 'fool the program' pseudo-transient is necessary to calculate the
A and C matrices in subroutine DISTRB. Just before calculating the B and D matrices,
which depend on a set of steady-state points, DISTRB sets ITRAN to 0.

Changes to Subroutine DISTRB

DATA statements and program statements in subroutine DISTRB control the cal-
culations of the A, B, C, and D matrices. DYGABCD provides a maximum of 23 pos-
sible time-varying state variables. These variables and their original order are as
follows:

<table>
<thead>
<tr>
<th>List number</th>
<th>Variable name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XNHP</td>
</tr>
<tr>
<td>2</td>
<td>XNIP</td>
</tr>
<tr>
<td>3</td>
<td>XNLP</td>
</tr>
<tr>
<td>4</td>
<td>P22</td>
</tr>
<tr>
<td>5</td>
<td>U22</td>
</tr>
<tr>
<td>6</td>
<td>P21</td>
</tr>
<tr>
<td>7</td>
<td>U21</td>
</tr>
<tr>
<td>8</td>
<td>P3</td>
</tr>
<tr>
<td>9</td>
<td>U3</td>
</tr>
<tr>
<td>10</td>
<td>P4</td>
</tr>
<tr>
<td>11</td>
<td>U4</td>
</tr>
<tr>
<td>12</td>
<td>P50</td>
</tr>
<tr>
<td>13</td>
<td>U50</td>
</tr>
<tr>
<td>14</td>
<td>P5</td>
</tr>
<tr>
<td>15</td>
<td>U5</td>
</tr>
<tr>
<td>16</td>
<td>P55</td>
</tr>
<tr>
<td>17</td>
<td>U55</td>
</tr>
<tr>
<td>18</td>
<td>P7</td>
</tr>
<tr>
<td>19</td>
<td>U7</td>
</tr>
<tr>
<td>20</td>
<td>P24</td>
</tr>
<tr>
<td>21</td>
<td>U24</td>
</tr>
<tr>
<td>22</td>
<td>P37</td>
</tr>
<tr>
<td>23</td>
<td>U37</td>
</tr>
</tbody>
</table>

The DATA statements and program statements include parameters that describe the
particular case the user wishes to run.
INV total number of states possible for particular engine being modeled, less than or equal to 23

INVRED total number of states actually used in model (full or reduced), less than or equal to INV

IVARB order in which states are to be done (If this is a reduced model, the states to be included are listed before those not included.)

INB number of system control inputs desired, less than or equal to 5

BPER percent (BPER=.02 means 2 percent) (The steady-state value of each input will be multiplied by this percent to form the \( \Delta u \) of each input in order to calculate the B and D matrices.)

INC number of outputs desired, less than or equal to 50

NUCOM list of equivalenced 'COM' numbers of outputs desired (All of the equivalences are in the main routine DYGABCD.)

PVRDOT vector of percents (PVRDOT=.02 means 2 percent) in list number order (The steady-state value of each state will be multiplied by the percent associated with that state to form the \( \Delta \dot{x} \) for that particular state in order to calculate the A and C matrices.)

There is one other necessary parameter that is set by a program statement - ICHOIC. If ICHOIC is 0, subroutine DISTRB calculates the A and C matrices by using the percentages of the states as set by the initial conditions of the PVRDOT's. If ICHOIC is set to 1, the percentages will be calculated and reset by the program to form an \( A^{-1} \) matrix with no zero columns or rows. A run with ICHOIC set to 1 should always be made first to find values for the PVRDOT's that do not result in an \( A^{-1} \) matrix containing any zero columns or rows. Obviously, an \( A^{-1} \) matrix that has zero columns or rows cannot be inverted.

Subroutine DISTRB is programmed to use inputs WFB (main fuel flow) and A8 (nozzle area). However, more and/or other inputs could be used. The number of inputs is limited to 5, but that limit could be changed by the user.

A line-by-line study of subroutine DISTRB should be made before attempting to make any runs other than the sample case. Subroutine DISTRB includes many comment cards that can be helpful. The listing starts on page 98.

Changes to Function DERIV

Function DERIV has been changed so that, if IAMTRX=1, a different set of statements is used. These statements set all derivatives to zero except for the particular
Changes to Subroutine PUTIN

Subroutine PUTIN has been changed so that it will accept one more variable in the namelist DATAIN. This variable is IAMTRY, which is defaulted to zero if DYGABCD is to be used as DYNGEN and is set to 1 if DYGABCD is to be used to generate the A, B, C, and D matrices.

Changes to Subroutine ENGBAL

Two changes have been made to subroutine ENGBAL. One change resets the time index if IAMTRX=1. This has been explained in the section Method. The other change causes subroutines DISTRB and COINLT to be called during the steady-state calculations for the B and D matrices if IAMTRX=1. If IAMTRX=0, subroutine ENGBAL behaves in the same manner as it did in DYNGEN.

Changes to Main Routine DYGABCD

The routine DYGABCD sets the default value of IAMTRX to zero and initializes the parameter IDOT to zero. IDOT is used internally in subroutine DISTRB to keep a record on the states being used to generate the A and C matrices and later to keep a record on the inputs used in the calculations for the B and D matrices.

SAMPLE CASE

The sample case is a two-spool, two-stream engine simulation that has a maximum of 16 states. The system control inputs used were WFB (main fuel flow) and A8 (nozzle area). The outputs specified were SFC (specific fuel consumption), FG (gross thrust), and FN (net thrust).

A steady-state point was run and the main fuel flow was noted to be 2.75 lbm/sec. The data card for the A, B, C, and D matrix calculations was inserted. In subroutine DISTRB, the following parameters were set in DATA statements: the number of states possible for this engine (16), in INV; the number of states for a reduced model (9), in INVRED; the order of states, in IVARB; the number of system control inputs (2), in
INB; the percent delta of the inputs for the calculation of the B and D matrices (0.01), in BPER; the number of outputs (3), in INC; and the list of the equivalenced 'COM' numbers of the outputs, in NUCom. Subroutine statements were as follows: ICHOIC was set to 1 so that subroutine DISTRB would calculate the PVRDOT's. These PVRDOT's were all initialized to 0.02. The A, B, C, and D matrices generated were used to form a linear model.

Two transient runs were made using DYGABCD as DYNGEN. The first run had a 3-percent step change in main fuel flow, and the other had a 3-percent step change in nozzle area. The results from these two full-state, nonlinear transients were the time histories of the states, inputs, and outputs. These nonlinear results (squares on the graphs) were then used as the standard against which the results from the various runs using the linear model (circles on the graphs) were plotted.

Since one of the state eigenvalues was extremely large, 15th order (instead of 16th) was the largest order linear model that could be used to generate time responses. A run was made using the 15th-order (full) linear model with a 3-percent step change in main fuel flow. Another run was made using a second-order (reduced) linear model with the same step change. These two runs were then repeated with a 3-percent step change in nozzle area.

The responses of the states and outputs from these four linear cases are compared with the responses from the nonlinear cases in figures 1 to 19. The results of the full-order linear runs and the nonlinear runs for the two step inputs are shown in the upper quadrants of each figure. The agreement is very good. The reduced (low order) linear runs and the nonlinear runs for the two step inputs are shown in the lower quadrants of each figure. Here, there is disagreement, especially in the first 0.1 second of the transient, because those parameters that cause the high-frequency dynamics are missing.

The FORTRAN symbols used are defined in appendix A. The program is listed in appendix B. Sample case input data and output are listed in appendix C.

CONCLUSIONS

DYGABCD is a computer program that produces the A, B, C, and D matrices from a nonlinear, generalized, transient engine simulation. These matrices are used to generate a linear model. The results from the nonlinear simulations and the full-order linear simulations show good agreement. This method also permits reduced-order
models to be formed that produce varying degrees of agreement. The method of calculating these matrices could be applied to many other dynamic engine simulations.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, April 21, 1978,
505-05.
APPENDIX A

FORTTRAN SYMBOLS

An asterisk denotes that the variable can be input.

A24  cross-sectional area at station 24, m² (ft²)
A25  cross-sectional area at station 25, m² (ft²)
* A28  area of fan duct nozzle throat, station 28, m² (ft²)
A28SAV  saved area of fan duct nozzle throat, station 28, at design conditions, m² (ft²)
A29  cross-sectional area at station 29, m² (ft²)
A29SAV  saved cross-sectional area at station 29 at design conditions, m² (ft²)
* A38  area of wing duct nozzle throat, station 38, m² (ft²)
A39  cross-sectional area at station 39, m² (ft²)
A55  cross-sectional area at station 55, m² (ft²)
* A6  cross-sectional area of afterburner entrance, station 6, calculated from AM6, m² (ft²)
A7  cross-sectional area at station 7, m² (ft²)
* A9  area of main nozzle throat, station 8, (can be changed at off-design), m² (ft²)
A8SAV  saved area of main nozzle throat, station 8, at design conditions, m² (ft²)
A9  cross-sectional area at station 9, m² (ft²)
A9SAV  saved cross-sectional area at station 9 at design conditions, m² (ft²)
* AFTPAN  (logical) control for an aft-fan engine
* ALTP  altitude of aircraft, m (ft)
* AM Mach number of aircraft

* AM23 Input: design Mach number of ductburner entrance, station 23
  Output: Mach number at station 23

AM25 Mach number at station 25
AM28 Mach number at station 28
AM29 Mach number at station 29
AM38 Mach number at station 38
AM39 Mach number at station 39

* AM55 Input: design Mach number at low-pressure turbine exit, station 55
  Output: Mach number at station 55

* AM6 Input: design Mach number at afterburner entrance, station 6
  Output: Mach number at station 6

AM6DSV saved Mach number at afterburner entrance at design conditions, station 6

AM7 Mach number at station 7
AM8 Mach number at station 8
AM9 Mach number at station 9

BLC bleed flow out of compressor, kg/sec (lbf/sec)
BLDU bleed flow into fan duct, kg/sec (lbf/sec)
BLF bleed flow out of fan (dumped overboard), kg/sec (lbf/sec)
BLHP bleed flow into high-pressure turbine, kg/sec (lbf/sec)
BLI airflow into third stream, kg/sec (lbf/sec)
BLIP bleed flow into intermediate-pressure turbine, kg/sec (lbf/sec)
BLLP bleed flow into low-pressure turbine, kg/sec (lbf/sec)
BLEED bleed flow lost overboard (customer bleed), kg/sec (lbm/sec)

BPRINT ratio of airflow into wing duct to airflow into core

BYPASS ratio of airflow into fan duct to airflow into intermediate compressor

CNC corrected shaft speed of core compressor

CNF corrected shaft speed of fan compressor

CNHP corrected shaft speed of high-pressure turbine

CNHPCF correction factor of high-pressure-turbine corrected speed

* CNHPDS design corrected speed of high-pressure turbine

CNI corrected shaft speed of intermediate compressor

CNIP corrected shaft speed of intermediate-pressure turbine

CNIPCF correction factor of intermediate-pressure-turbine corrected speed

* CNIPDS design corrected speed of intermediate-pressure turbine

CNLF corrected shaft speed of low-pressure turbine

CNLPCF correction factor of low-pressure turbine corrected speed

* CNLPS design corrected speed of low-pressure turbine

CS ambient speed of sound, m/sec (ft/sec)

* CVDNOZ velocity coefficient of duct nozzle thrust

* CVDWNG velocity coefficient of wing nozzle thrust

* CVMNOZ velocity coefficient of core nozzle thrust

* DELFG gross-thrust delta degradation multiplier

* DELPN net-thrust delta degradation multiplier
* **DELSFC** specific-fuel-consumption delta degradation multiplier

* **DELT1** correction to standard-day temperature, K (°F)

**DHPCF** \( \Delta \) enthalpy correction factor of high-pressure turbine

**DHIPCF** \( \Delta \) enthalpy correction factor of intermediate-pressure turbine

**DHLPF** \( \Delta \) enthalpy correction factor of low-pressure turbine

**DHTC** work done by high-pressure turbine, J/kg (Btu/lbm)

**DHTCHP** enthalpy change of high-pressure turbine, temperature corrected, J/kg-K (Btu/lbm-°R)

**DHTCIP** enthalpy change of intermediate-pressure turbine, temperature corrected, J/kg-K (Btu/lbm-°R)

**DHTCLP** enthalpy change of low-pressure turbine, temperature corrected, J/kg-K (Btu/lbm-°R)

**DHTF** work done by low-pressure turbine, J/kg (Btu/lbm)

**DHTI** work done by intermediate-pressure turbine, J/kg (Btu/lbm)

* **DPAFD** design pressure drop (\( \Delta P/P \)) of afterburner

**DPAFT** pressure drop (\( \Delta P/P \)) of afterburner

* **DPCODS** design pressure drop (\( \Delta P/P \)) of combustor

**DPDCOM** pressure drop (\( \Delta P/P \)) of combustor

**DPDUC** pressure drop (\( \Delta P/P \)) of fan duct

* **DPDUDS** design pressure drop (\( \Delta P/P \)) of fan duct

* **DPWGD** design pressure drop (\( \Delta P/P \)) of wing duct

**DPWING** pressure drop (\( \Delta P/P \)) of wing duct

* **DT** solution time step for transient, sec

* **DTPRNT** time step for output listings, sec
DUMD? (15) dummy variables

* DUMSPL  (logical) control for spool that does not change temperature or pressure of air
* ERRER  (logical) test of exceeding ITFS

* ETAA  efficiency of afterburner
* ETAADS  design efficiency of afterburner
ETAA SV  saved efficiency of afterburner at design conditions
ETAB  efficiency of combustor
ETABCF  correction factor of combustor efficiency
* ETABDS  design efficiency of combustor
ETAC  adiabatic efficiency of core compressor
ETACCF  correction factor of core-compressor efficiency
* ETACDS  design adiabatic efficiency of core compressor
* ETAF  efficiency of ductburner
ETAF  adiabatic efficiency of fan compressor
ETAFCF  correction factor of fan-compressor efficiency
* ETAPDS  design adiabatic efficiency of fan compressor
ETAI  adiabatic efficiency of intermediate compressor
ETAI CF  correction factor of intermediate-compressor efficiency
* ETAIDS  design adiabatic efficiency of intermediate compressor
* ETAR  pressure recovery of inlet (ram recovery), \( \frac{p2}{p1} \)
ETATHP  adiabatic efficiency of high-pressure turbine
ETATIP  adiabatic efficiency of intermediate-pressure turbine
ETATLP  adiabatic efficiency of low-pressure turbine

24
ETHPCF: correction factor of high-pressure-turbine efficiency

* ETPDS: design adiabatic efficiency of high-pressure turbine

ETIPCF: correction factor of intermediate-pressure-turbine efficiency

* ETIPDS: design adiabatic efficiency of intermediate-pressure turbine

ETLPF: correction factor of low-pressure-turbine efficiency

* ETLPDS: design adiabatic efficiency of low-pressure turbine

* FAN: (logical) control that indicates fan or turbojet

FAR24: fuel-to-air ratio at station 24

FAR4: fuel-to-air ratio at station 4

FAR5: fuel-to-air ratio at station 5

FAR50: fuel-to-air ratio at station 50

FAR55: fuel-to-air ratio at station 55

FAR7: fuel-to-air ratio at station 7

FAR7SV: saved fuel-to-air ratio at station 7 at design conditions

FCOVFN: ratio of core thrust to net thrust

FFOVFN: ratio of fan thrust to net thrust

FG: gross thrust, N (lbf)

FGM: momentum thrust of all but wing, N (lbf)

FGMD: fan duct momentum thrust of all but wing, N (lbf)

FGMM: core nozzle momentum thrust of all but wing, N (lbf)

FGMWNG: momentum thrust of wing, N (lbf)
FGP  pressure thrust of all but wing, N (lbf)
FGPD fan duct pressure thrust of all but wing, N (lbf)
FGPM core nozzle pressure thrust of all but wing, N (lbf)
FGPWNG pressure thrust of wing, N (lbf)
FMNOFN ratio of fan plus core thrust to net thrust
FN net thrust, N (lbf)
FNMAIN net thrust of all but wing, N (lbf)
FNOVFD ratio of net thrust to design-point net thrust
FNWING net thrust of wing, N (lbf)
PRD ram drag, N (lbf)
FWOVFN ratio of net wing thrust to net thrust
* FXFPN2M (logical) control for boosted fan
* FXM2CP (logical) control for supercharged compressor
H1 enthalpy at station 1, J/kg (Btu/lbm)
H2 enthalpy at station 2, J/kg (Btu/lbm)
H21 enthalpy at station 21, J/kg (Btu/lbm)
H22 enthalpy at station 22, J/kg (Btu/lbm)
H23 enthalpy at station 23, J/kg (Btu/lbm)
H24 enthalpy at station 24, J/kg (Btu/lbm)
H25 enthalpy at station 25, J/kg (Btu/lbm)
H28 enthalpy at station 28, J/kg (Btu/lbm)
H29 enthalpy at station 29, J/kg (Btu/lbm)
H3 enthalpy at station 3, J/kg (Btu/lbm)
H38 enthalpy at station 38, J/kg (Btu/lbm)
H39 enthalpy at station 39, J/kg (Btu/lbm)
H4  enthalpy at station 4, J/kg (Btu/lbm)
H5  enthalpy at station 5, J/kg (Btu/lbm)
H50 enthalpy at station 50, J/kg (Btu/lbm)
H55 enthalpy at station 55, J/kg (Btu/lbm)
H6  enthalpy at station 6, J/kg (Btu/lbm)
H7  enthalpy at station 7, J/kg (Btu/lbm)
H8  enthalpy at station 8, J/kg (Btu/lbm)
H9  enthalpy at station 9, J/kg (Btu/lbm)

* HPEXT  power extracted, W (hp)
* IAPTBN index for afterburning desired; zeroes automatically; values of 0, 1, or 2
* IAMTP index for ram or inlet operation desired; values of 0 to 5
* IAMTRX index that indicates if A,B,C, and D matrices are to be calculated
ICOAFB index of error in subroutine COAFBN
ICODUC index of error in subroutine CODUCT
ICOMIX index of error in subroutine COMIX
* IDBURN index for duct burning desired; zeroes automatically; values of 0, 1, or 2
* IDC D duct nozzle convergent-divergent when value=1; convergent when value=0
* IDES index for design point; zeroes automatically; must be set to 1 to design engine
* IDUMP index for dumping of error matrix; values of 0, 1, or 2
IGASMX index for mixed or no-mixed flow turbofans; values of -1, 0, 1, or 2
* IMCD  main nozzle convergent-divergent when value=1; convergent when value=0
* INIT  index for initializing guesses; zeroes automatically; 0 calls GUESS, 1 does not

* ISPOOL  number of engine rotors

* ITRAN  index for initiating transient

* ITRYS  maximum number of iterations allowed through engine counter

JTRAN  index that indicates a transient is in progress

LOOPER  number of loops through engine counter

* MODE  independent variable designator for engine operation

* NOZPLT  index for floating main or duct nozzle; zeroes automatically; values of 0 to 3

  P1  standard pressure, \( \text{N/m}^2 \) (atm)

  P2  total pressure at station 2, \( \text{N/m}^2 \) (atm)

  P21  total pressure at station 21, \( \text{N/m}^2 \) (atm)

  P22  total pressure at station 22, \( \text{N/m}^2 \) (atm)

  P23  total pressure at station 23, \( \text{N/m}^2 \) (atm)

  P24  total pressure at station 24, \( \text{N/m}^2 \) (atm)

  P25  total pressure at station 25, \( \text{N/m}^2 \) (atm)

  P28  total pressure at station 28, \( \text{N/m}^2 \) (atm)

  P29  total pressure at station 29, \( \text{N/m}^2 \) (atm)

  P3  total pressure at station 3, \( \text{N/m}^2 \) (atm)

  P37  total pressure at station 37, \( \text{N/m}^2 \) (atm)

  P38  total pressure at station 38, \( \text{N/m}^2 \) (atm)

  P39  total pressure at station 39, \( \text{N/m}^2 \) (atm)

  P4  total pressure at station 4, \( \text{N/m}^2 \) (atm)

  P5  total pressure at station 5, \( \text{N/m}^2 \) (atm)
P50 total pressure at station 50, N/m² (atm)
P55 total pressure at station 55, N/m² (atm)
P6 total pressure at station 6, N/m² (atm)
P6DSAV saved total pressure at station 6 at design conditions, N/m² (atm)
P7 total pressure at station 7, N/m² (atm)
P8 total pressure at station 8, N/m² (atm)
P9 total pressure at station 9, N/m² (atm)

* PCBLC ratio of compressor bleed to cool turbines to total compressor airflow

* PCBLDU ratio of compressor bleed leaked into fan duct to total compressor bleed flow

* PCBLF ratio of bleed from fan compressor to fan airflow dumped overboard (i.e., leakage)

* PCBLHP fraction of PCBLC used in high-pressure turbine for cooling

* PCBLI fraction of intermediate-compressor air that goes into third stream

* PCBLID ratio of design value of air into wing to air into core; zero for two-stream engine

* PCBLIP fraction of PCBLC used in intermediate-pressure turbine for cooling

* PCBLLP fraction of PCBLC used in low-pressure turbine for cooling

* PCBLOB fraction of bleed air out of compressor lost overboard (for customer use)

* PCNC shaft speed of core compressor as a percent of design

* PCNCDS design corrected speed of core compressor as a percent of design

PCNCGU guessed shaft speed of core compressor as a percent of design
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCNF</td>
<td>Shaft speed of fan compressor as a percent of design</td>
</tr>
<tr>
<td>PCNFDS</td>
<td>Design corrected speed of fan compressor as a percent of design</td>
</tr>
<tr>
<td>PCNFGU</td>
<td>Guessed shaft speed of fan compressor as a percent of design</td>
</tr>
<tr>
<td>PCNI</td>
<td>Shaft speed of intermediate compressor as a percent of design</td>
</tr>
<tr>
<td>PCNIDS</td>
<td>Design corrected speed of intermediate compressor as a percent of design</td>
</tr>
<tr>
<td>PCVIGU</td>
<td>Guessed shaft speed of intermediate compressor as a percent of design</td>
</tr>
<tr>
<td>PMIHP</td>
<td>Polar moment of inertia of high-pressure rotor, kg-m² (slug-ft²)</td>
</tr>
<tr>
<td>PMIIP</td>
<td>Polar moment of inertia of intermediate-pressure rotor, kg-m² (slug-ft²)</td>
</tr>
<tr>
<td>PMILP</td>
<td>Polar moment of inertia of low-pressure rotor, kg-m² (slug-ft²)</td>
</tr>
<tr>
<td>PRC</td>
<td>Pressure ratio of core compressor</td>
</tr>
<tr>
<td>PRCCF</td>
<td>Correction factor of core-compressor pressure ratio</td>
</tr>
<tr>
<td>PRCDS</td>
<td>Design pressure ratio of core compressor</td>
</tr>
<tr>
<td>PRF</td>
<td>Pressure ratio of fan compressor</td>
</tr>
<tr>
<td>PRCPCF</td>
<td>Correction factor of fan-compressor pressure ratio</td>
</tr>
<tr>
<td>PRFDS</td>
<td>Design pressure ratio of fan compressor</td>
</tr>
<tr>
<td>PRI</td>
<td>Pressure ratio of intermediate compressor</td>
</tr>
<tr>
<td>PRICF</td>
<td>Correction factor of intermediate-compressor pressure ratio</td>
</tr>
<tr>
<td>PRIDS</td>
<td>Design pressure ratio of intermediate compressor</td>
</tr>
<tr>
<td>PS28</td>
<td>Static pressure at station 28, N/m² (atm)</td>
</tr>
<tr>
<td>PS29</td>
<td>Static pressure at station 29, N/m² (atm)</td>
</tr>
</tbody>
</table>
PS38  static pressure at station 38, N/m² (atm)
PS39  static pressure at station 39, N/m² (atm)
* PS55  Input: static pressure at low-pressure-turbine exit, station 55, N/m² (atm)
         Output: static pressure at station 55, N/m² (atm)
PS6   static pressure at station 6, N/m² (atm)
PS7   static pressure at station 7, N/m² (atm)
PS8   static pressure at station 8, N/m² (atm)
PS9   static pressure at station 9, N/m² (atm)
S1    entropy at station 1, J/kg-K (Btu/lbm-°R)
S2    entropy at station 2, J/kg-K (Btu/lbm-°R)
S21   entropy at station 21, J/kg-K (Btu/lbm-°R)
S22   entropy at station 22, J/kg-K (Btu/lbm-°R)
S23   entropy at station 23, J/kg-K (Btu/lbm-°R)
S24   entropy at station 24, J/kg-K (Btu/lbm-°R)
S25   entropy at station 25, J/kg-K (Btu/lbm-°R)
S28   entropy at station 28, J/kg-K (Btu/lbm-°R)
S29   entropy at station 29, J/kg-K (Btu/lbm-°R)
S3    entropy at station 3, J/kg-K (Btu/lbm-°R)
S4    entropy at station 4, J/kg-K (Btu/lbm-°R)
S5    entropy at station 5, J/kg-K (Btu/lbm-°R)
S50   entropy at station 50, J/kg-K (Btu/lbm-°R)
S55   entropy at station 55, J/kg-K (Btu/lbm-°R)
S6    entropy at station 6, J/kg-K (Btu/lbm-°R)
S7    entropy at station 7, J/kg-K (Btu/lbm-°R)
S8    entropy at station 8, J/kg-K (Btu/lbm-°R)
S9    entropy at station 9, J/kg-K (Btu/lbm-°R)
SFC  specific fuel consumption, kg/N-hr (lbm/lbf-hr)

* SI  (logical) control for SI or U.S. customary (English) units

T1  standard temperature, station 1, K (°R)

* T2  Input:  total temperature at fan inlet, station 2, K (°R)
      Output: total temperature at station 2, K (°R)

T21  total temperature at station 21, K (°R)

T21DS total temperature at intermediate-compressor exit at design conditions, station 21, K (°R)

T22  total temperature at station 22, K (°R)

T22DS design total temperature at fan exit, station 22, K (°R)

T23  total temperature at station 23, K (°R)

* T24  Input: total temperature at ductburner exit, station 24, K (°R)
       Output: total temperature at station 24, K (°R)

T25  total temperature at station 25, K (°R)

T28  total temperature at station 28, K (°R)

T29  total temperature at station 29, K (°R)

T2DS design total temperature at fan inlet, station 2, K (°R)

T3  total temperature at station 3, K (°R)

T38  total temperature at station 38, K (°R)

T39  total temperature at station 39, K (°R)

* T4  Input: total temperature at combustor exit, station 4, K (°R)
       Output: total temperature at station 4, K (°R)

* T4DS design total temperature at combustor exit, station 4, K (°R)

T4GU guessed total temperature at station 4, K (°R)
T5  total temperature at station 5, K (°R)
T50 total temperature at station 50, K (°R)
T55 total temperature at station 55, K (°R)
T6  total temperature at station 6, K (°R)

* T7  Input: total temperature at afterburner exit, station 7, K (°R)
      Output: total temperature at station 7, K (°R)

* T7DS design total temperature at afterburner exit, station 7, K (°R)

T8  total temperature at station 8, K (°R)
T9  total temperature at station 9, K (°R)

* TF  final time for transient, sec

TFAR total fuel-to-air ratio

TFFHP flow function of high-pressure turbine,
   kg-√K-m²/N-sec (lbm-√R-in²/lbf-sec)

TFFIP flow function of intermediate-pressure turbine,
   kg-√K-m²/N-sec (lbm-√R-in²/lbf-sec)

TFFLP flow function of low-pressure turbine,
   kg-√K-m²/N-sec (lbm-√R-in²/lbf-sec)

TFHPCF correction factor of high-pressure turbine flow

* TFHPDS design flow function of high-pressure turbine,
   kg-√K-m²/N-sec (lbm-√R-in²/lbf-sec)

TFIPCF correction factor of intermediate-pressure
      turbine flow function

* TFIPDS design flow function of intermediate-pressure
      turbine, kg-√K-m²/N-sec (lbm-√R-in²/lbf-sec)

TFLPCF correction factor of low-pressure turbine flow

* TFLPDS design flow function of low-pressure turbine,
   kg-√K-m²/N-sec (lbm-√R-in²/lbf-sec)
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>time, sec</td>
</tr>
<tr>
<td>* TOLALL</td>
<td>tolerance on convergence</td>
</tr>
<tr>
<td>TS28</td>
<td>static temperature at station 28, K (°R)</td>
</tr>
<tr>
<td>TS29</td>
<td>static temperature at station 29, K (°R)</td>
</tr>
<tr>
<td>TS38</td>
<td>static temperature at station 38, K (°R)</td>
</tr>
<tr>
<td>TS39</td>
<td>static temperature at station 39, K (°R)</td>
</tr>
<tr>
<td>TS7</td>
<td>static temperature at station 7, K (°R)</td>
</tr>
<tr>
<td>TS8</td>
<td>static temperature at station 8, K (°R)</td>
</tr>
<tr>
<td>TS9</td>
<td>static temperature at station 9, K (°R)</td>
</tr>
<tr>
<td>U21</td>
<td>internal energy at station 21, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U22</td>
<td>internal energy at station 22, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U24</td>
<td>internal energy at station 24, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U3</td>
<td>internal energy at station 3, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U37</td>
<td>internal energy at station 37, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U4</td>
<td>internal energy at station 4, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U5</td>
<td>internal energy at station 5, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U50</td>
<td>internal energy at station 50, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U55</td>
<td>internal energy at station 55, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>U7</td>
<td>internal energy at station 7, J/kg (Btu/lbm)</td>
</tr>
<tr>
<td>V25</td>
<td>velocity at station 25, m/sec (ft/sec)</td>
</tr>
<tr>
<td>V28</td>
<td>velocity at station 28, m/sec (ft/sec)</td>
</tr>
<tr>
<td>V29</td>
<td>velocity at station 29, m/sec (ft/sec)</td>
</tr>
<tr>
<td>V38</td>
<td>velocity at station 38, m/sec (ft/sec)</td>
</tr>
<tr>
<td>V39</td>
<td>velocity at station 39, m/sec (ft/sec)</td>
</tr>
<tr>
<td>V55</td>
<td>velocity at station 55, m/sec (ft/sec)</td>
</tr>
<tr>
<td>V6</td>
<td>velocity at station 6, m/sec (ft/sec)</td>
</tr>
</tbody>
</table>
V7 velocity at station 7, m/sec (ft/sec)
V8 velocity at station 8, m/sec (ft/sec)
V9 velocity at station 9, m/sec (ft/sec)
VA velocity of aircraft, m/sec (ft/sec)
* VAPTBN control volume associated with afterburner, m³ (ft³)
* VCOMB control volume associated with combustor, m³ (ft³)
* VCOMP control volume associated with core compressor, m³ (ft³)
* VFAN control volume associated with fan compressor, m³ (ft³)
* VFDUCT control volume associated with fan duct, m³ (ft³)
* VHPTRB control volume associated with high-pressure turbine, m³ (ft³)
* VINTC control volume associated with intermediate compressor, m³ (ft³)
* VIPTRB control volume associated with intermediate-pressure turbine, m³ (ft³)
VJD velocity of fan duct exhaust, m/sec (ft/sec)
VJM velocity of core exhaust, m/sec (ft/sec)
VJW velocity of wing duct exhaust, m/sec (ft/sec)
* VLPTRB control volume associated with low-pressure turbine, m³ (ft³)
* VWDUCT control volume associated with wing duct, m³ (ft³)
WA21 airflow at station 21, kg/sec (lbm/sec)
WA22 airflow at station 22, kg/sec (lbm/sec)
WA23DS airflow at station 23 at design conditions, kg/sec (lbm/sec)
WA3 airflow at station 3, kg/sec (lbm/sec)
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA32</td>
<td>airflow at station 32, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WA32DS</td>
<td>airflow at station 32 at design conditions, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WA3CDS</td>
<td>corrected airflow in combustor at design conditions, station 3, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WAC</td>
<td>airflow of core compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WACC</td>
<td>corrected airflow of core compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>* WACCDS</td>
<td>design corrected airflow of core compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WACCF</td>
<td>correction factor of core-compressor corrected airflow</td>
</tr>
<tr>
<td>WACDS</td>
<td>airflow of core compressor at design conditions, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WACI</td>
<td>corrected airflow of intermediate compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WACP</td>
<td>saved airflow of core compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WAD</td>
<td>airflow of fan duct, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WAF</td>
<td>airflow of fan compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WAFC</td>
<td>corrected airflow of fan compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>* WAFCDS</td>
<td>design corrected airflow of fan compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WAFCF</td>
<td>correction factor of fan compressor corrected airflow</td>
</tr>
<tr>
<td>WAFDS</td>
<td>airflow of fan compressor at design conditions, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WAPP</td>
<td>saved airflow of fan compressor, kg/sec (lbm/sec)</td>
</tr>
<tr>
<td>WAI</td>
<td>airflow of intermediate compressor, kg/sec (lbm/sec)</td>
</tr>
</tbody>
</table>
* WAICDS  design corrected airflow of intermediate compressor, kg/sec (lbm/sec)

WAICF  correction factor of intermediate-compressor corrected airflow

WAIDS  airflow of intermediate compressor at design conditions, kg/sec (lbm/sec)

WAIP  saved airflow of intermediate compressor, kg/sec (lbm/sec)

* WFA  fuel flow rate to afterburner (IAPTB=2 only for input), kg/sec (lbm/sec)

* WFB  fuel flow rate to main combustor (MODE=2 only for input), kg/sec (lbm/sec)

* WFBDS  design fuel flow rate to main combustor (MODE=2 only for input), kg/sec (lbm/sec)

WFD  fuel flow rate to ductburner, kg/sec (lbm/sec)

WFT  total fuel flow rate, kg/sec (lbm/sec)

WG24  gas flow at station 24, kg/sec (lbm/sec)

WG37  gas flow at station 37, kg/sec (lbm/sec)

WG4  gas flow at station 4, kg/sec (lbm/sec)

WG5  gas flow at station 5, kg/sec (lbm/sec)

WG50  gas flow at station 50, kg/sec (lbm/sec)

WG55  gas flow at station 55, kg/sec (lbm/sec)

WG6  gas flow at station 6, kg/sec (lbm/sec)

WG6CDS  corrected gas flow at station 6 at design conditions, kg/sec (lbm/sec)

WG7  gas flow at station 7, kg/sec (lbm/sec)

WGT  total gas flow rate, kg/sec (lbm/sec)

XBLDU  saved BLDU

XBLF  saved BLF
**XFAR24**  saved FAR24  
**XFAR55**  saved FAR55  
**XH21**  saved H21  
**XH25**  saved H25  
**XH3**  saved H3  
**XH55**  saved H55  
**XNHP**  speed of core compressor, rpm  
* **XNHPDS**  design speed of high-pressure rotor, rpm  
**XNIP**  speed of intermediate compressor, rpm  
* **XNIPDS**  design speed of intermediate-pressure rotor, rpm  
**XNLDEM**  commanded speed of fan compressor, rpm  
**XNLP**  speed of fan compressor, rpm  
* **XNLPD**  design speed of low-pressure rotor, rpm  
**XP1**  saved P1  
**XP21**  saved P21  
**XP25**  saved P25  
**XP55**  saved P55  
**XS21**  saved S21  
**XS25**  saved S25  
**XS55**  saved S55  
**XT21**  saved T21  
**XT25**  saved T25  
**XT55**  saved T55  
**XWAC**  saved WAC  
**XWAP**  saved WAP  
**XWFB**  saved WFB
XWFD   saved WFD
XWG24  saved WG24
XWG55  saved WG55
XXP1   saved P1

ZC     ratio of core-compressor pressure ratios

* ZCDS  design ratio of core compressor; equals pressure ratio at design point on design speed line minus value of pressure ratio at lowest point on speed line, divided by high (surge) value minus low value of pressure ratio on design speed line

ZF     ratio of fan-compressor pressure ratios

* ZFDS  design ratio of fan compressor (see ZCDS for explanation)

ZI     ratio of intermediate-compressor pressure ratios

* ZIDS  design ratio of intermediate compressor (see ZCDS for explanation)
APPENDIX B

FORTRAN LISTINGS

Main Program DYGABCD

C THIS IS THE MAIN PROGRAM
IMPLICIT REAL*8 (A-H,O-Z) (DHTI, COM(145)), (ZIDS, COY(136)), (PCNIDS, COM(1b7)), ABC30043
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2), ERR(9), DUMD(15), FO(50,4), SD(10,6), ABCD0034
1 PDATA(5,50), TIMPT(50), XS(23), PYRD00(23) ABCD0035
EQUIVALENCE (WORD(1), COM(1)), (IDDS, COM(3)), (IDES, COM(4)), ABCD0037
1 (KDES, COM(5)), (MODA, COM(6)), (EIT, COM(7)), (TDPMP, COM(8)), ABCD0039
2 (IAMTP, COM(3)), (IGAMX, COM(10)), (OUBARN, COM(11)), ABCD0092
3 (TAPTSN, COM(12)), (ICMD, COM(13)), ABCD0010
4 (IDSDNC, COM(15)), (IMSHDC, COM(16)), (W2PFLT, COM(17)), ABCD0011
5 (IETRS, COM(18)), (LOOPER, COM(19)), (NOMAP, COM(20)), ABCD0012
6 (NUMMAP, COM(21)), (MAPEDS, COM(22)), (TOLALL, COM(23)), ABCD0013
7 (ERR(1), COM(24)), (P1, COM(33)), (H22, COM(34)), ABCD0014
8 (AM23, COM(35)), (W23DS, COM(36)), (T23, COM(37)), ABCD0015
9 (P23, COM(38)), (H23, COM(39)), (S23, COM(40)), (A24, COM(41)), ABCD0016
1 (T24, COM(42)), (H24, COM(43)), (S24, COM(44)), (A25, COM(45)), ABCD0017
2 (I25, COM(46)), (P25, COM(47)), (H25, COM(48)), (S25, COM(49)), ABCD0018
3 (AZ2, COM(50)), (A26SAV, COM(51)), (AM29, COM(52)), ABCD0028
4 (V26, COM(53)), (T26, COM(54)), (PS29, COM(55)), (T29, COM(56)), ABCD0029
5 (P29, COM(57)), (H28, COM(58)), (S26, COM(59)), (A29, COM(60)), ABCD0021
6 (V29, COM(61)), (A29, COM(62)), (V29, COM(63)), ABCD0022
7 (T29, COM(64)), (PS29, COM(65)), (T29, COM(66)), (P29, COM(67)), ABCD0023
8 (H29, COM(68)), (S29, COM(69)), (BYPASS, COM(70)), ABCD0024
9 (WAD, COM(71)), (WFD, COM(72)), (ETAP, COM(73)), ABCD0025
1 (DPDUC, COM(74)), (DPDUDS, COM(75)), (XP21, COM(76)), ABCD0025
EQUIVALENCE (XWG24, COM(77)), (XFAR24, COM(78)), (XT25, COM(79)), ABCD0027
1 (XP25, COM(80)), (XK25, COM(81)), (S25, COM(82)), ABCD0028
2 (XK55, COM(83)), (XFAR55, COM(84)), (XT55, COM(85)), ABCD0029
3 (IP55, COM(86)), (IX55, COM(87)), (S55, COM(88)), ABCD0030
4 (WKP6, COM(89)), (XWFD, COM(90)), (T2D5, COM(91)), (T2, COM(92)), ABCD0301
5 (P2, COM(93)), (F2, COM(94)), (S2, COM(95)), (S22, COM(96)), ABCD0302
6 (W22DS, COM(97)), (A25, COM(98)), (V25, COM(99)), ABCD0333
7 (W25, COM(100)), (T2DS, COM(101)), (T5, COM(102)), ABCD0334
8 (H5, COM(103)), (55, COM(104)), (4GC, COM(105)), ABCDC3835
9 (FAR5, COM(106)), (AM55, COM(177)), (V55, COM(109)), ABCD0336
1 (AS55, COM(109)), (P55, COM(110)), (S56, COM(111)), ABCD0337
2 (P56, COM(112)), (V6, COM(113)), (TPHPCF, COM(114)), ABCD0338
3 (CNHPCF, COM(115)), (ETHPCF, COM(116)), (DHHPCF, COM(117)), ABCD0339
4 (THPDS, COM(118)), (CATHPCF, COM(119)), (ETHPDS, COM(120)), ABCD0004
5 (BFPCF, COM(121)), (ETAFPC, COM(122)), (WAFCF, COM(123)), ABCD0041
6 (FCPCDS, COM(124)), (FPCDS, COM(125)), (ETAPDS, COM(126)), ABCD0042
7 (WAPDS, COM(127)), (PCNCSU, COM(128)), (HPEXT, COM(129)), ABCD043
8 (WACCDS, COM(130)), (FPP, COM(131)), (ETAP, COM(132)), ABCD0044
9 (ZCCDS, COM(133)), (CNF, COM(134)), (KAPC, COM(135)), ABCD0045
1 (ZP, COM(136)), (CPNCP, COM(137)), (PCBF, COM(138)), ABCD0005
EQUIVALENCE (ZI, COM(139)), (PCNI, COM(140)), (TFFIP, COM(141)), ABCD0047
1 (CHIP, COM(142)), (ETATIP, COM(143)), (DHTCIP, COM(144)), ABCD0048
2 (DHTI, COM(145)), (ZIDS, COM(146)), (PCNIDS, COM(147)), ABCD0049
3 (PCNIGU, COM(148)), (T1, COM(149)), (H1, COM(150)), ABCD0050
4 (S1, COM(151)), (T3, COM(152)), (R3, COM(153)), (W3, COM(154)), ABCD0051
5 (W3CDS, COM(155)), (T4, COM(156)), (H4, COM(157)), ABCD0052
6 (S4, COM(158)), (WGL, COM(159)), (PASU, COM(160)), ABCD0053
7 (T50, COM(161)), (H50, COM(162)), (S50, COM(163)), ABCD0054
2 (P6DSAV, COM (325)), (AM6DSV, COM (326)), (AM6, COM (327)), ABCD0109
3 (A5, COM (328)), (466DCS, COM (329)), (R6, COM (330)), ABCD0110
4 (T6, COM (331)), (P6, COM (332)), (H6, COM (333)), (467, COM (334)), ABCD0111
5 (PAR7, COM (335)), (FABTSV, COM (336)), (TS7, COM (337)), ABCD0112
6 (PS7, COM (338)), (V7, COM (339)), (AM7, COM (340)), (A7, COM (341)), ABCD0113
7 (T7DS, COM (342)), (T7, COM (343)), (H7, COM (344)), (57, COM (345)), ABCD0114
8 (A8, COM (346)), (AB8SVA, COM (347)), (TSA, COM (348)), ABCD0115
9 (PS8, COM (349)), (V8, COM (350)), (AM8, COM (351)), (P8, COM (352)), ABCD0115
1 (PB, COM (353)), (HB, COM (354)), (S8, COM (355)), (A9, COM (356)), ABCD0117
2 (A9SVA, COM (357)), (T9, COM (358)), (PS9, COM (359)), ABCD0118
3 (V9, COM (360)), (AM9, COM (361)), (T9, COM (362)), (P9, COM (363)), ABCD0119
4 (H9, COM (364)), (S9, COM (365)), (ETAAPN, COM (366)), ABCD0120
5 (DPAPDS, COM (367)), (WFA, COM (368)), (ETAA, COM (369)), ABCD0121
6 (ETAASV, COM (370)), (DPAPF, COM (371)), (XHNP, COM (372)), ABCD0122
7 (XNIP, COM (373)), (XNLB, COM (374)), (P22, COM (375)), ABCD0123
8 (U22, COM (376)), (P21, COM (377)), (U21, COM (378)), ABCD0124
9 (P3, COM (379)), (U3, COM (380)), (P4, COM (381)), (U4, COM (382)), ABCD0125
1 (P5, COM (383)), (U50, COM (384)), (P5, COM (385)), (U5, COM (386)), ABCD0126
EQUVALENCE (P55, COM (387)), (U55, COM (388)), (P7, COM (389)), ABCD0127
1 (V7, COM (390)), (P24, COM (391)), (U24, COM (392)), ABCD0128
2 (P37, COM (393)), (U37, COM (394)), (VFAN, COM (395)), ABCD0129
3 (VINTC, COM (396)), (VCOMP, COM (397)), (VCOMB, COM (398)), ABCD0130
4 (VHPTRB, COM (399)), (VLPTRB, COM (400)), (VLPTRB, COM (401)), ABCD0131
5 (VAPFB, COM (402)), (VFBUTC, COM (403)), (VFBUTC, COM (404)), ABCD0132
6 (DUMD1(1), COM (405)), (WAFF, COM (423)), (WAPF, COM (421)), ABCD0133
7 (ZACP, COM (422)), (XNHPD, COM (423)), (XNHPS, COM (424)), ABCD0134
8 (XNLPS, COM (425)), (PMNH, COM (426)), (PMHR, COM (427)), ABCD0135
9 (PMLP, COM (428)), (DELT1, COM (429)), (FO(1, 1), COM (430)), ABCD0136
1 (XNLDEM, COM (630)), (ISO(1, 1), COM (631)), (FDATA(1, 1), COM (691)), ABCD0137
2 (TIMEPT(1), COM (941)), (PPFNEW, COM (991)), (PRCNEW, COM (992)), ABCD0138
3 (TIME, COM (993)), (DT, COM (994)), (TF, COM (995)), ABCD0139
4 (TPRINT, COM (996)), (DTPRNT, COM (997)), (PVDOT(1), COM (998)), ABCD0140
5 (XS(1), COM (1021)), (ISPOOL, COM (1044)), (ICOAPB, COM (1045)), ABCD0141
6 (ICODUC, COM (1046)), (ICOMIX, COM (1047)), (KKGO, COM (1048)), ABCD0142
7 (ITRAN, COM (1049)), (JTRAN, COM (1050)), (NSTEP, COM (1051)), ABCD0143
8 (IVBIDOT, COM (1052)), (IDOT, COM (1053)), (ITMTRX, COM (1054)), ABCD0144
9 (SI, COM (1055)), (ERROR, COM (1056)), (DUMSPL, COM (1057)), ABCD0145
1 (FXF2M, COM (1058)), (FXM2CFP, COM (1059)), (AMTAP, COM (1060)), ABCD0146
EQUVALENCE (FAN, COM (1061)), (WAFCD, COM (1062)), ABCD0147
ERROR = .FALSE.
ITRAN = 0
JTRAN = 0
NSTEP = 0
TIME = 0.000
TPRINT = 0.000
DTPRNT = 0.000
IDOT = 0
TAMTRX = 0
ABCD0155
1 DO 1 J = 3, 4, 04
2 WORD(J) = 0.000
3 DO 2 J = 1057, 1062
4 C SET ARBITRARY VALUES FOR INTERMEDIATE SPOOL DESIGN PARAMETERS TO ABCD0162
5 WORD(J) = 0.000
6 ISPOOL = 0
7 ABCD0161
C AVOID ERROR WHEN RUNNING A DUMMYPOOL ENGINE
      PRIDS = 1.5D0
      ETAIDS = 1.0D0
      PCNIDS = 100.0D0
      ZIDS = 0.75D0
      PCNCDS = 100.0D0
      KKGO = 0
      CALL CONOUT(1)
      PBDSAV = 1.0D0
      AM6DSV = 1.0D0
      ETAASV = 1.0D0
      PAR7SV = 1.0D0
      CALL ENGBAL
      STOP
      END

Subroutine AFQUIR

      SUBROUTINE AFQUIR (X,AIND,DEPEND,ANS,AJ,TOL,DIR,BNEW,ICON
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(9)
      C X(1) = NAME OF ARRAY TO USE
      C AIND = INDEPENDENT VARIABLE
      C DEPEND = DEPENDENT VARIABLE
      C ANS = ANSWER UPON WHICH TO CONVERGE
      C AJ = MAX NUMBER OF TRIES
      C TOL = PERCENT TOLERANCE FOR CONVERGENCE
      C DIR = DIRECTION AND PERCENTAGE FOR FIRST GUESS
      C ANEW = CALCULATED VALUE OF NEXT TRY AT INDEPENDENT VARIABLE
      C ICON = CONTROL = 1 GO THRU LOOP AGAIN
      C = 2 YOU HAVE REACHED THE ANSWER
      C = 3 COUNTER HAS HIT LIMITS
      C X(2) = COUNTER STORAGE
      C X(3) = CHOOSES METHOD OF CONVERGENCE
      C X(4) = THIRD DEPEND VAR
      C X(5) = THIRD IND VAR
      C X(6) = SECOND DEPEND VAR
      C X(7) = SECOND IND VAR
      C X(8) = FIRST DEPEND VAR
      C X(9) = FIRST IND VAR
      C X(3) MUST BE ZERO UPON FIRST ENTRY TO ROUTINE
      Y = 0.0D0
      IF (ANS .EQ. 0.0D0) GO TO 2
      DEF = DEPEND - ANS
      TOLANS = TOL * ANS
      GO TO 3
      2 DEF = DEPEND
      TOLANS = TOL
      IF (ABS(DEF) .LE. TOLANS) GO TO 5
      IF (X(2) - AJ) 8,8,7
      5 ANEW = AIND
      X(2) = 0.0D0
      ICON = 2
      RETURN
6  ANEW = Y
   X(2) = X(2) + 1.0D0
   ICON = 1
   RETURN
7  ANEW = Y
   X(2) = 0.0D0
   ICON = 3
   RETURN
8  IF (X(3) .GT. 0.0D0) GO TO 12
C *** FIRST GUESS USING DIR
9  X(3) = 1.0D0
   X(8) = DEP
   X(9) = AIND
   IF (AIND .EQ. 0.0D0) GO TO 11
   Y = DIR * AIND
   GO TO 6
   Y = DIR
   GO TO 6
10 IF (X(3) .GT. 1.0D0) GO TO 16
C *** LINEAR GUESS
11 X(3) = 2.0D0
   X(6) = DEP
   X(7) = AIND
   IF (X(8) .EQ. X(6) .OR. X(9) .EQ. X(7)) GO TO 9
   A = (X(9) - X(7)) / (X(8) - X(6))
   Y = X(9) - A * X(8)
   IF (DABS(10.0D0 * X(9)) - DABS(Y)) 9,9,6
C *** QUADRATIC GUESS
12 X(3) = 1.0D0
   X(6) = DEP
   X(7) = AIND
   IF (X(9) .NE. X(5)) GO TO 18
   IF (X(6) .EQ. X(4)) GO TO 13
   IF (X(9) .NE. X(5)) GO TO 23
   IF (X(8) .EQ. X(4)) GO TO 22
13 X(9) = X(7)
   X(8) = X(6)
   GO TO 13
14 X(9) = X(7)
   X(8) = X(6)
   GO TO 30
15 X(4) = DEP
   X(5) = AIND
   IF (X(7) .NE. X(5)) GO TO 18
   IF (X(6) - X(4)) 13,9,13
16 IF (X(6) .EQ. X(4)) GO TO 13
17 IF (X(9) .EQ. X(5)) GO TO 23
18 IF (X(8) .EQ. X(4)) GO TO 22
19 X(9) = X(7)
   X(8) = X(6)
   GO TO 13
20 X(9) = X(7)
   X(8) = X(6)
   X(3) = 1.0D0
21 IF (X(9)) 10,11,10
22 IF (X(8) .EQ. X(4)) GO TO 21
23 P = (X(6) - X(4)) / (X(7) - X(5))
24 A = (X(8) - X(4) - P * (X(9) - X(5))) / ((X(9) - X(7)) * (X(9) - X(5)))
25 B = P - A * (X(5) + X(7))
26 C = X(4) + X(5) * (A * X(7) - P)
27 IF (A .NE. 0.0D0) GO TO 27
28 IF (B .EQ. 0.0D0) GO TO 7
29 Y = -C / B
   GO TO 47
30 IF (B .NE. 0.0D0) GO TO 32
31 IF (C .NE. 0.0D0) GO TO 30
32
Y = 0.0D0
GO TO 47

30  G = -C / A
IF (G .LE. 0.0D0) GO TO 7
Y = DSQRT(G)
YY = -DSQRT(G)
GO TO 37

32  IF (C .NE. 0.0D0) GO TO 34
Y = -B / A
YY = 0.0D0
GO TO 37

34  D = 4.0D0 * A * C / B ** 2
IF (1.0D0 - D) .LT. 37,35,36
GO TO 47

36  E = DSQRT(1.0D0 - D)
Y = (-B / (2.0D0 * A)) * (1.0D0 + E)
YY = (-B / (2.0D0 * A)) * (1.0D0 - E)

37  J = 4
DEPMIN = DABS(X(4))
DO 39  I = 6,8,2
IF (DEPMIN .LE. DABS(X(I))) GO TO 39
J = I
DEPMIN = DABS(X(I))
CONTINUE

39  K = J + 1
IF (((X(K) - Y) * (X(K) - YY) .LE. 0.0D0) GO TO 42
IF (((DABS(X(K) - Y) - DABS(X(K) - YY)) .LE. 0.0D0) GO TO 47
Y = YY
GO TO 47

42  IF (J .GE. 6) GO TO 44
JJ = J + 2
KK = K + 2
GO TO 45

44  JJ = J - 2
KK = K - 2
GO TO 45

45  SLOPE = (X(KK) - X(K)) / (X(JJ) - X(J))
IF ((SLOPE * X(J) * (X(K) - Y)) .LT. 0.0D0) GO TO 47
Y = YY

47  X(9) = X(7)
X(8) = X(6)
X(7) = X(5)
X(6) = X(4)
GO TO 6
END

Subroutine ATMOS

SUBROUTINE ATMOS (ZFT, TM, SIGMA, RHO, THETA, DELTA, CA, AMU, K)
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL SI
C THIS IS A SUBROUTINE TO COMPUTE CERTAIN ELEMENTS OF THE 1962
C U.S. STANDARD ATMOSPHERE UP TO 90 KILOMETERS.
C CALLING SEQUENCE...
CALL ATMOS (ZFT, TM, SIGMA, RHO, THETA, DELTA, CA, AMU, K)
ZFT = GEOMETRIC ALTITUDE (FEET)
TM = MOLECULAR SCALE TEMPERATURE (DEGREES KELVIN)
SIGMA = RATIO OF DENSITY TO THAT AT SEA LEVEL
RHO = DENSITY (LB-SEC**2-FT**(-4) OR SLUG-FT**(-3))
THETA = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL
DELTA = RATIO OF PRESSURE TO THAT AT SEA LEVEL
CA = SPEED OF SOUND (FT/SEC)
AMU = VISCOSITY COEFFICIENT (LB-SEC/FT**2)

K = 1 NORMAL
= 2 ALTITUDE LESS THAN -5300 METERS OR GREATER THAN 90 KM
= 3 FLOATING POINT OVERFLOW

ALL DATA AND FUNDAMENTAL CONSTANTS ARE IN THE METRIC SYSTEM AS THESE QUANTITIES ARE DEFINED AS EXACT IN THIS SYSTEM.

THE RADIUS OF THE EARTH (REFT59) IS THE VALUE ASSOCIATED WITH THE 1959 ABEC ATMOSPHERE SO THAT PROGRAMS CURRENTLY USING THE LIBRARY ROUTINE WILL NOT REQUIRE ALTERATION TO USE THIS ROUTINE.

COMMON /COMALL/ COM(1062)
DIMENSION HB(10), TMB(10), DELTAB(10), ALM(10)
EQUIVALENCE (SI, COM(1055))

DATA HB /-5.0DO, 0.0DO, 11.0DO, 20.0DO, 32.0DO, 47.0DO, 52.0DO, ABCD0343
1 61.0DO, 79.0DO, 88.743DO /
DATA TMB /320.65D0, 288.15D0, 2*216.65D0, 228.65D0,
1 2*270.65D0, 252.65D0, 2*190.65D0 /
DATA DELTAB /1.75363D+03, 1.0~000D+00, 2.23361D-01, 5.43328D-02,ABCD0347
1 8.56663D-03, 1.09455D-03, 5.82289D-04, 1.79718D-04, 1.0241D-05, 1.6223D-06 /
DATA ALM /2*6.5DO, 0.0DO, 1.0DO, 2.8DO, 0.0DO, -2.0DO,
1 -4.0DO, 2*0.0DO /
DATA REFT59/2.0855531D+07/, GZ /9.80665D+00/, ABCD0352
1 AMZ /2.89644D+01/, RSTT /8.31432D+00/, A3CD0353
2 FTTOKM/3.048D-04/, S /1.104D+02/, A3CD0354
3 AMUZ /1.2024D+05/, CAZ /1.11645D+03/, ABCD0355
4 RHOZ /7.6474D-02/, GZENG /3.21741D+01/, ABCD0356
C CONVERT GEOMETRIC ALTITUDE TO GEOPOTENTIAL ALTITUDE
C IF IN SI UNITS, CHANGE ZFT TO FEET
IF (SI) ZFT = ZFT * 3.280833D0
HFT = (REFT59 / (REFT59 + ZFT)) * ZFT
C CONVERT HFT AND ZFT TO KILOMETRES
Z = FTTOKM * ZFT
H = FTTOKM * HFT
K = 1
TMZ = TMB(2)
IF (H .LT. -5.0DO OR Z .GT. 90.0DO) GO TO 7
DO 1 M = 1,10
IF (H - HB(M)) 2,3,1
1 CONTINUE
GO TO 7
2 M = M - 1
3 DELH = H - HB(M)
IF (ALM(N) .EQ. 0.0D0) GO TO 4

TMK = TMB(M) * ALM(N) * DELH

4

C GRADIENT IS NON ZERO, PAGE 10, EQUATION I.2.10-(3)

DELTA = DELTAB(M) * ((TMB(M) / TMK) ** (GZ * AMZ / 1 (RSTAR * ALM(M))))

GO TO 5

5

C GRADIENT IS ZERO, PAGE 10, EQUATION I.2.10-(4)

DELTA = DELTAB(M) * DEXP(-GZ * AMZ * DELH / (RSTAR * TMB(M)))

THETA = TMK / TMZ

SIGMA = DELTA / THETA

ALPHA = DSQRT(THETA ** 3) * ((TMZ + S) / (TMK + S))

C CONVERSION TO ENGLISH UNITS

TM = 1.8D0 * TMK

RHO = RHOS * SIGMA / GZENG

CA = CAZ * DSQRT(THETA)

AMU = AMUZ * ALPHA / GZENG

IF (SI) GO TO 100

GO TO 101

100

TM = TM / 1.8D0

RHO = RHO * 515.379D0

CA = CA * .3048D0

AMU = AMU * 47.880258D0

ZFT = ZFT / 3.280833D0

IF IN SI UNITS:

C T = DEGREES KELVIN

C RHO = KG/M**2

C CA = M/SEC

C AMU = (N-SEC)/M**2

C ZFT = M

101

CALL OVERFL (J)

IF (J .EQ. 2) RETURN

K = K + 2

RETURN

7

RETURN

END

Block Data

BLOCK DATA

IMPLICIT REAL*8 (A-H,O-Z)

C THESE ARE THE GENERALIZED MAPS FOR AN UNREALISTIC SUPERSONIC ENGINE. THEY INCLUDE: FAN, INTERMEDIATE FAN, COMPRESSOR, COMBUSTOR, AND HIGH, INTERMEDIATE AND LOW TURBINES.

COMMON /COMDAT/ CNXP(15), PRXP(15,15), WACXP(15,15), ETAXF(15,15),

1 NPTF(15)

DIMENSION CNXP(15), PRXP(15,15), WACXP(15,15), ETAXF(15,15),

1 NPTI(15)

DIMENSION CNXP(15), PRXP(15,15), WACXP(15,15), ETAXF(15,15),

1 NPTP(15)

DIMENSION CNXN(15), PRXIN(15,15), WACXN(15,15), ETAXIN(15,15),

1 NPTI(15)

DIMENSION CNXP(15), PRXP(15,15), WACXP(15,15), ETAXP(15,15),

1 NPTP(15)

DIMENSION CNXN(15), PRXIN(15,15), WACXN(15,15), ETAXIN(15,15),

1 NPTI(15)

DIMENSION CNXP(15), PRXP(15,15), WACXP(15,15), ETAXP(15,15),

1 NPTP(15)

DIMENSION CNXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION CNXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION CNXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)

DIMENSION PSIXB(15), DELXH(15,15), ETAXB(15,15), NPTB(15)

DIMENSION TFFXH(15,15), CNXH(15,15), DHTCXH(15,15), ETAXH(15,15),

1 NPTP(15)
1 NPTTFH(15)
DIMENSION TFFXI(15), CNXI(15,15), DHTCXI(15,15), ETATXI(15,15),
NPTTFI(15)
DIMENSION CNXI(15,15), CNXI2(15,1)
DIMENSION TFFXL(15), CNXL(15,15), DHTCXL(15,15), ETATXL(15,15),
NPTTFI(15)
DIMENSION CNXL1(15,15), CNXL2(15,1)
EQUIVALENCE (CNXP(1), CMD(1)), (PRXP(1), CMD(16)),
(WACXP(1,1), COMD(241)), (ETAXF(1,1), COMD(466)),
(NPTF(1), CMD(5296)), (CNX1(1,1), CMD(5297))
EQUIVALENCE (CNXI(1), CMD(691)), (PBXIN(1,1), CMD(706)),
(WACXI(1,1), COMD(931)), (ETAXIN(1,1), CMD(1156)),
(NCNI, CMD(5298)), (CNXI(1,1), CMD(5312))
EQUIVALENCE (CNXP(1), CMD(5344)), (PRXP(1), CMD(5345))
EQUIVALENCE (PSIXB(1), CMD(2751)), (DELXB(1,1), CMD(2776)),
EQUIVALENCE (TFFXI(1), CMD(3226)), (CNXH(1,1), CMD(3241))
EQUIVALENCE (TFFXL(1), CMD(4381)), (CNXL(1,1), CMD(4382))
EQUIVALENCE (CNXH(1,1), CMD(5392)), (CNXL(1,1), CMD(5403))
EQUIVALENCE (CNXP(1), CMD(5393)), (PRXP(1), CMD(5404))
EQUIVALENCE (PSIXB(1), CMD(2751)), (DELXB(1,1), CMD(2776)),
EQUIVALENCE (TFFXI(1), CMD(3226)), (CNXH(1,1), CMD(3241))
EQUIVALENCE (TFFXL(1), CMD(4381)), (CNXL(1,1), CMD(4382))
EQUIVALENCE (CNXH(1,1), CMD(5392)), (CNXL(1,1), CMD(5403))
EQUIVALENCE (CNXP(1), CMD(5393)), (PRXP(1), CMD(5404))
EQUIVALENCE (PSIXB(1), CMD(2751)), (DELXB(1,1), CMD(2776)),
EQUIVALENCE (TFFXI(1), CMD(3226)), (CNXH(1,1), CMD(3241))
EQUIVALENCE (TFFXL(1), CMD(4381)), (CNXL(1,1), CMD(4382))
EQUIVALENCE (CNXH(1,1), CMD(5392)), (CNXL(1,1), CMD(5403))
EQUIVALENCE (CNXP(1), CMD(5393)), (PRXP(1), CMD(5404))
EQUIVALENCE (PSIXB(1), CMD(2751)), (DELXB(1,1), CMD(2776)),
EQUIVALENCE (TFFXI(1), CMD(3226)), (CNXH(1,1), CMD(3241))
EQUIVALENCE (TFFXL(1), CMD(4381)), (CNXL(1,1), CMD(4382))
EQUIVALENCE (CNXH(1,1), CMD(5392)), (CNXL(1,1), CMD(5403))
EQUIVALENCE (CNXP(1), CMD(5393)), (PRXP(1), CMD(5404))
EQUIVALENCE (PSIXB(1), CMD(2751)), (DELXB(1,1), CMD(2776)),
EQUIVALENCE (TFFXI(1), CMD(3226)), (CNXH(1,1), CMD(3241))
EQUIVALENCE (TFFXL(1), CMD(4381)), (CNXL(1,1), CMD(4382))
EQUIVALENCE (CNXH(1,1), CMD(5392)), (CNXL(1,1), CMD(5403))
EQUIVALENCE (CNXP(1), CMD(5393)), (PRXP(1), CMD(5404))
EQUIVALENCE (PSIXB(1), CMD(2751)), (DELXB(1,1), CMD(2776)),
EQUIVALENCE (TFFXI(1), CMD(3226)), (CNXH(1,1), CMD(3241))
EQUIVALENCE (TFFXL(1), CMD(4381)), (CNXL(1,1), CMD(4382))
EQUIVALENCE (CNXH(1,1), CMD(5392)), (CNXL(1,1), CMD(5403))
EQUIVALENCE (CNXP(1), CMD(5393)), (PRXP(1), CMD(5404))
EQUIVALENCE (PSIXB(1), CMD(2751)), (DELXB(1,1), CMD(2776))
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

**DATA ETAXF**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

**DATA NCNI, NPTP**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

**DATA PRIN**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

**DATA WACIN**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

**DATA ETA1**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

**DATA NCP, NPTP**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

**DATA PKP**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
<tr>
<td>1</td>
<td>10 * 1.0, 5 * 0.0, 1.84, 1.966, 1.84, 2.008, 2.519, 2.855, 3.261</td>
<td>ABCD0532</td>
</tr>
<tr>
<td>2</td>
<td>1.686, 4.353, 3.765, 5 * 0.0, 2.428, 3.093, 2.68, 3.429, 3.982</td>
<td>ABCD0533</td>
</tr>
<tr>
<td>3</td>
<td>4.297, 4.759, 3.849, 7.622, 6.481, 5 * 0.0, 2.869, 3.933, 3.408</td>
<td>ABCD0534</td>
</tr>
<tr>
<td>4</td>
<td>4.605, 5.277, 5.613, 6.117, 5.466, 10.219, 9.176, 5 * 0.0, 3.835</td>
<td>ABCD0535</td>
</tr>
<tr>
<td>5</td>
<td>4.689, 4.521, 5.697, 6.488, 6.936, 7.454, 6.866, 11.059, 10.219</td>
<td>ABCD0535</td>
</tr>
<tr>
<td>6</td>
<td>5 * 0.0, 4.549, 5.529, 5.445, 6.614, 7.202, 7.622, 8.308, 8.371</td>
<td>ABCD0537</td>
</tr>
<tr>
<td>7</td>
<td>11.899, 11.479, 7 * 0.0, 6.313, 7.538, 8.0, 8.546, 9.218, 8.956</td>
<td>ABCD0538</td>
</tr>
<tr>
<td>8</td>
<td>13.159, 12.711, 7 * 0.0, 5.523, 7.958, 8.567, 9.134, 9.638, 9.638</td>
<td>ABCD0539</td>
</tr>
<tr>
<td>9</td>
<td>9.883, 13.656, 14.412, 9 * 0.0, 9.386, 9.925, 10.513, 10.912, 10.912</td>
<td>ABCD0540</td>
</tr>
<tr>
<td>10</td>
<td>11 * 0.0, 9.596, 10.219, 10.995, 11.815, 82 * 0.0 /</td>
<td>ABCD0541</td>
</tr>
<tr>
<td>DATA</td>
<td>WACXP</td>
<td>ABCD0542</td>
</tr>
<tr>
<td>DATA</td>
<td>ETAXB</td>
<td>ABCD0542</td>
</tr>
<tr>
<td>DATA</td>
<td>DELXB</td>
<td>ABCD0543</td>
</tr>
<tr>
<td>DATA</td>
<td>PSIXB</td>
<td>ABCD0544</td>
</tr>
<tr>
<td>DATA</td>
<td>ETAXP</td>
<td>ABCD0545</td>
</tr>
<tr>
<td>DATA</td>
<td>WACXP</td>
<td>ABCD0546</td>
</tr>
</tbody>
</table>
| DATA | TFFXH / 39.67, 42.99, 47.46, 48.61, 49.175, 49.6, 50.0, 50.425, 50.92, 51.575, 5 * 0.0 /  
| 10 * | 1.872, 5 * 0.0, =372, =394, .4362, .255, =3.9568, =4314,  
| 2.8555, .8969, | 9.0312, .0494, =661, .0698, .07, .0693, .0668, 5  
| 3 * | 7, .8368, .e494  
| 6.3515, .8596, 3 | 2.697, 3.0392, 2.8334, 3.0764, 2.495, 2.045, 1.445,  
| 1.8156, 1.4638, 6 * 0.0, 2.945, 3 * 3.774, 3.1618, 2.5554,  
| 1.2196, .7696, 5 * 0.0, 1.9324, 1.7086,  
| 2.2138, 1.9794, 2.3362, 2.1245, 1.8618, 1.595, 1.3138, .0868,  
| 5 * 0.0, 2.15, 1.9696, 2.552, 2.2794, 2.645, 2.2706, 1.9558,  
| 1.6746, 1.3696, .8254, 5 * 0.0, 2.4058, 2.2706, 2.805, 2.5138,  
| 6 * 0.0, 2.7862, 3.096, 3.2648, 3.1422, 3.152, 2.5372, 2.0824,  
| 1.8156, 1.4638, 6 * 0.0, 2.945, 3 * 3.774, 3.1618, 2.5554,  
| 2.101, 1.8196, 1.4676, 6 * 0.0 /  
| DATA | DHTCXH /  
| 1.0032, .0038, .0046, .0052, .0056, .0068, .008, .0088, .0093,  
| 0.0123, 5 * 0.0, 0.0057, .008, 0.01, 0.0068, 0.0088, 0.012, 0.0164,  
| 0.0196, .0159, .018, 5 * 0.0, 0.0084, 0.0113, 0.0144, 0.012, 0.0146,  
| 0.0192, .0236, .0272, 0.0232, 0.0223, 5 * 0.0, 0.0108, 0.0136, 0.0148,  
| 0.0164, .0192, .0252, 0.0308, 0.0316, 0.0284, 0.0268, 5 * 0.0, 0.0133,  
| 0.0156, 0.0164, 0.0204, 0.0236, 0.03, 0.0372, 0.0356, 0.033, 0.034,  
| 7 * 0.0, 0.0152, .0176, .024, 0.0244, 0.0288, 0.034, 0.0416, 0.0392,  
| 0.0136, .0135, 5 * 0.0, .0164, .0192, 0.0268, 0.028, 0.0321, .0398,  
| 0.0448, .0432, 0.04, .038, 5 * 0.0, .0174, .0212, .0292, .0304,  
| 0.0316, 0.0421, 0.0476, 0.046, 0.0442, 0.0412, 5 * 0.0, .0179, .0228,  
| 0.0316, 0.0336, 0.04, 0.0472, 0.0472, 0.0488, 0.048, 0.044, 5 * 0.0,  
| 0.0176, 0.0248, 0.0331, 0.0356, 0.0444, 0.0524, 0.054, 0.0528, 0.0524,  
| 0.0476, 5 * 0.0, 0.0167, 0.026, 0.0344, 0.0388, 0.0496, 0.356, 0.3576,  
| 0.056, 0.0556, 0.0504, 5 * 0.0, 0.0144, 0.0261, 0.0346, 0.0412, 0.054,  
| 0.0612, 0.06, 0.0596, 0.058, 0.053, 5 * 0.0, 0.012, 0.0241, 0.034, 0.041,  
| 0.0596, 0.064, 0.0628, 0.064, 0.0612, 6 * 0.0, 0.0082, 0.0188, 0.0324,  
| 0.0472, 0.064, 0.0668, 0.064, 0.0654, 0.064, 6 * 0.0, 0.0034, 0.0128,  
| 0.0312, 0.0494, 0.0661, 0.0669, 0.07, 0.0693, 0.0668, 6 * 0.0 /  
| DATA | ESTATXH /  
| 1.6219, .0668, 0.05764, .5643, .5562, .5309, .5062, .5051, .4909,  
| 2.4257, 5 * 0.0, 3 * .7078, 5 * .6068, .539, .4747, 5 * 0.0,  
| 3.7868, 2 * .809, 5 * .7079, .5053, .5056, 5 * 0.0, .809, .8292,  
| 4.8494, 4 * .809, .7665, .6573, .5359, 5 * 0.0, .809, .8353,  
| 5.8543, 4 * .809, .7665, .6573, .5359, 5 * 0.0, .809, .8353,  
| 6.3515, .8596, 3 * .8697, .8292, .7463, .5941, 5 * 0.0, .7779,  
| 7.8368, .8494, .8596, .8695, .3319, .7977, .8494, .7776, .7068,  
| 8 * 0.0, 7.422, 6.302, .8409, .8575, .8662, 2 * .8999, .3596,  
| 9.809, 6.178, 5 * 0.0, .7078, .3254, .8626, 2.8535, .8615, .894,  
| 1.8954, .8697, .8191, .624, 5 * 0.0, .7635, 2 * .809, .8434,  
| 2.8555, .8969, 3, .8808, .8302, 5 * 0.0, .6068, .7696,  
| 3.7579, .8363, .852, .8975, .901, .8849, .8347, .8265, 5 * 0.0,  

51
Subroutine COAFBN

SUBROUTINE COAFBN
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2)
DIMENSION Q(9), AWORD(2)

EQUIVALENCE (WORD(1) , COM(1)) , (IDES , COM(3)) , (IGASXY, COM(10))
1 (TAFTBN , COM(12)) , (TOLALL.COM(23)) , (WG24 , COM(321)) , (FAP24, COM(377))
2 COM(322)) , (WG35 , COM(323)) , (FAP35, COM(324)) , (P6D5S4, COM(378))
3 COM(325)) , (AM6DSV, COM(326)) , (AM6, COM(327)) , (AM6, COM(328))
4 (WG6CDS, COM(329)) , (WG6, COM(330)) , (T6, COM(331))
5 (P6, COM(332)) , (H6, COM(333)) , (WG7, COM(334))
6 (P6T7, COM(335)) , (P6T7S, COM(336)) , (T57, COM(337))
7 (PS7, COM(338)) , (T7, COM(339)) , (AM7, COM(340)) , (A7, COM(341))
8 (TD75S, COM(342)) , (T77, COM(343)) , (T7, COM(344)) , (T7, COM(345))
9 (ETAAADS, COM(365)) , (DPAPDS, COM(366)) , (WFA, COM(367))
10 (TEAA, COM(369)) , (ETAASV, COM(370)) , (DPAP, COM(371))
11 (P7, COM(379)) , (P7, COM(390)) , (WATEN, COM(402))
12 (ICOAFB, COM(1045)) , (SI, COM(1055))

DATA AWORD /HICOAF, 4HMAF, 4HMAF/ WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
Q(2) = 0.0DQ
Q(3) = 0.0DQ
IF (SI) GO TO 100
AJ = 778.26D0
AJX = 2.719D0
CAPSF = 2116.217D0
G = 32.17140.49D0

END
PRATM = 14.69600
TDEL = 2000.000
T7MAX = 4000.000
RA = .025200
GO TO 101

100
AJ = 1.000
AJX = 1.000
CAPSF = 101325.000
G = 1.000
PRATM = 14.69600 / 101324.600
TDEL = 1111.000
T7MAX = 2222.000
RA = 286.900

101
GAJ2 = 2.000 G * AJ
ICOAPB = 0

C** P6DS AND AM6DS ARE SET FOR GENERALIZATION OF AFTERBURNER

C*** EFFICIENCY * AM GENERALIZATION
IF (IDES .NE. 1) GO TO 102
P6DS = P6 * PRATM
AM6DS = AM6

102
WF6 = FAR55 + WG55 / (FAR55 + 1.000)
IF (IGASMX .GT. 0) W6 = WF6 + FAR28 * WG28 / (FAR28 + 1.000)
WA6 = WG6 - WF6

C*** DRY LOSS
WGbC = WG6 * DSQRT(T6) / P6
IF (IDES .EQ. 1) WG6CDS = WG6C
DPAFT = DPAAPDS * (WG6C / WG6CDS)
IF (DPAFT .GT. 1.000) DPAFT = 1.000
P7 = P6 * (1.000 - DPAFT)
A7 = A6
FAR6 = WF6 / WA6
CALL PPROC (FAR6, T6, XX1, XX2, XX3, XX4, PH16, XX6)
WQA = WG6 / A7
C1 = P7 * DSQRT(G / (T6 * AJ)) * CAPSF
AM7 = AM6
TS7 = 0.87500 * T6

1
DO 2 I = 1, 50
CALL PPROC (FAR6, TS7, CS7, AK7, CP7, XP7, PHIS7, HS7)
V7 = AM7 * CS7
HSCAL = H6 - V7 ** 2 / GAJ2
DELHS = HSCAL - HS7
IF (DABS (DELHS) .LE. -5110 * TOLERI. * HSCAL) GO TO 3

2
TS7 = TS7 + DELHS / CP7
ICOAPB = 1
GO TO 14

3
WQAT = C1 * DSQRT(AK7 / REEX7) * AM7 / (1.000 + (AK7 - 1.000))
1 AM7 ** 2 / 2.000 ** ((AK7 + 1.000) / (2.000 * (AK7 - 1.000)))
DIR = WQA / WQAT
EW = (WQA - WQAT) / WQA
CALL AFQFR (Q(1), AM7, EW, 0.000, 0.000, 1.000 * TOLALL, DIR, AM7T, IGO)
ICOAPB = 2
GO TO (4, 5, 14), IGO

4
AM7 = AM7T
IF (AM7 .LE. 0.000) AM7 = 1.000 - 0
IF (AM7 .GE. 1.0) AM7 = 0.90D0
GO TO 1

PS7 = P7 / DEXP((PHI6 - PHI67) / PEX7)
IF (IAFTBN .GT. 0) GO TO 7

C *** NON-AFTERBURNING

6 T7 = T6
WFA = 0.0D0
FAR7 = FAR6
WG7 = WG6
IF (IDES .LT. 1 .OR. T7DS .EQ. 3.9D0) GO TO 26

C *** AFTERBURNING

7 IF (IAFTBN .EQ. 2) T7 = T6 + DEL
IF (IDES .EQ. 1) T7 = T7DS
IF (T7 .LE. T6) GO TO 6

RH065 = CAPSF * PS7 / (AJ * PEX7 * TS7)
PS65 = PS7
V65 = V7
Q(2) = 0.0D0
Q(3) = 0.0D0

8 IF (T7 .GT. T7MAX) T7 = T7MAX
IF (T7 .LT. T6) T7 = T6 * 1.031D0
IF (SI) T7 = T7 * 9.0D0 / 5.0D0
HV = ((((( - .4594317 - 19 - T7) - .2034116D-15) * T7 +
1 .2783643D-11) * T7 + .2051501D-77) * T7 - .2453116D-03) * T7 -
2 .9433296D-01) * T7 + .1845537D+05
IF (.NOT, SI) GO TO 103
T7 = T7 * 5.0D0 / 9.0D0
HV = HV * 2325.4295D0

103 CALL THERMO (P7, HA, T7, XX1, XX2, 1, FAR6, 0)

C *** TO ALTER DESIGN ABETA MAP FROM GENERAL TO SPECIFIC MAP

9 P6GS = P6 * PRATM
PAR7GS = (HA - H6) / (HV * ETAADS)
DO 10 11 = 1,50
IF (IDES .NE. 1) GO TO 9
PAR7DS = (HA - H6) / (HV * ETAADS)
PAR7 = (HA - H6) / (HV * ETAADS)
DELFA7 = DABS (PAR7 - PAR7GS)
IF (DELFA7 .LE. 1.0D0 * TOLALL * PAR7) GO TO 11

10 PAR7GS = PAR7
IF (PAR7 .GT. 0.0D0) GO TO 12
ICOAPB = 3
CALL ERROR

12 WFA = FAR7 * WG6
IF (IAFTBN .EQ. 1) GO TO 15
ERRW = (WFA - WFX) / WFA
DIR = DSQRT(WFA / WFX)
CALL AFQUIR (Q(1), T7, ERRW, 0.0D0, 30.0D0, 5.0D0*TOLALL, DIR, T7, IGO)
ICOAPB = 4

56
GO TO (13, 16, 14), IGO

13 T7 = T7T
GO TO 8

14 CALL ERROR

15 WFA = WFAx

16 FAR7 = (WP6 + WFA) / WA6
WG7 = WG6 + WFA

C *** MOMENTUM LOSS
CALL PROCOM (FAR7, T7, XX1, XX2, XX3, REX7, PHI7, H7)
RH07 = CAPSF * E7 / (AJ * REX7 * T7)
V7 = WG7 / (RH07 * A7)
Q(2) = 0.0D0
Q(3) = 0.0D0
PS7 = PS65 - 0.01D0

RH07 = WG7 / (V7 * A7)
HS7 = H7 - V7 ** 2 / GAJ2
CALL THERMO (1.0D0, HS7, TS7, PHI7, XX2, 1, FAR7, 1)
IF (TS7 .GE. 301.0D0) GO TO 13
CALL THERMO (1.0D0, HS7, 400.0D0, PHI7, XX2, 1, FAR7, 0)
V7 = DSQRT(GAJ2 * (H7 - HS7))
GO TO 17

17 PS7 = RH07 * AJ * REX7 * PS7 / CAPSF
PS7A = PS65 + (RH065 * V65 ** 2 - RH07 * V7 ** 2) / (S * CAPSF)
DIR = DSQRT(DABS(Ps7 / PS7A))
EP = (PS7 - PS7A) / PS7
CALL AQUIR (Q(1), V7, EP, 0.0D0, 50.0D0, 1.0D0 * TOLALL, DIR, V7T, IGO)
V7 = V7T
IF (V7 .LT. 100.0D0) V7 = 100.0D0
ICOAFB = 5
GO TO (17, 19, 14), IGO

19 P7 = PS7 * DEXP((PHI7 - PHI7S) / REX7)
CALL PROCOM (FAR7, TS7, CS7, XX2, XX3, XX4, XX5, XX6)
AM7 = V7 / CS7
GO TO 18

20 CALL THERMO (P7, H7, T7, S7, XX2, 1, FAR7, 0)
IF (VAFTBN .EQ. 0.0D0) GO TO 31
Q(2) = 0.0D0
Q(3) = 0.0D0
WG7P = WG7
H7P = H7
P7DOT = DERIV(18, P7)
GO TO 28

28 CALL THERMO (P7, H7, T7, S7, XX2, 1, FAR7, 0)
WG7 = WG7P - P7DOT * VAFTBN / P7 / (1.4D0 * RA)
U7 = H7 - AJX * RA * T7
U7DOT = DERIV(19, U7)
H7X = (WG7P * H7P - (WG7P - WG7) * U7 - U7DOT * P7 * VAFTBN / 1 T7 / RA) / WG7
ERRW = (H7 - H7X) / H7
DIR = DSQRT(DABS(H7 / H7X))
CALL AQUJR (Q(1), T7, ERRW, 0.0D0, 20.0D0, 1.0D0 * TOLALL, DIR, T7T, IGO)
ICOAPB = 6
GO TO (29, 31, 30), IGO

29 T7 = T7T
GO TO 28

30 CALL ERROR
Subroutine COCOMB

```
SUBROUTINE COCOMB
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL SI, PXM2CP
COMMON /COMALL/ COM(1062)
COMMON /COMDAT/ COMD(5423)
DIMENSION WORD(2), PSIXB(15), DELXB(15, 15), ETAXB(15, 15), NPTB(15)
1 NPTB(15)
DIMENSION Q(9), DUMBO(15, 15), AWORD(2)
EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (MODE, COM(6)),
1 (MAPEDG, COM(22)), (TOLALL, COM(23)), (T3, COM(152)), (H3,
ABCDO973
2 COM(153)), (WA3, COM(154)), (WA3CDS, COM(155)), (T4, COM(156)),
3 (H4, COM(157)), (S4, COM(158)), (WG4, COM(159)),
ABCDO979
4 (FAR4, COM(160)), (T50, COM(161)), (H50, COM(162)),
ABCDO973
5 (S50, COM(163)), (WG50, COM(164)), (TAR50, COM(165)),
ABCDO980
6 (PCBLHP, COM(166)), (PCBLIP, COM(167)), (PCBLLP, COM(168)),
ABCDO981
7 (PCBLDU, COM(169)), (PCBLDG, COM(170)), (CMHP, COM(171)),
ABCDO982
8 (ETATHP, COM(172)), (DHTC, COM(173)), (DHTC, COM(174)),
ABCDO983
9 (TFPHP, COM(175)), (ETABCF, COM(179)), (ETABDS, COM(184)),
ABCDO984
1 (WPB, COM(192)), (WAC, COM(191)),
ABCDO985
3 (DPCOM, COM(193)), (P3, COM(379)), (P4, COM(381)),
ABCDO987
4 (U4, COM(382)), (P50, COM(363)), (VCOMB, COM(398)),
ABCDO983
5 (ISPOOL, COM(1044)), (ITRAN, COM(1049)), (SI, COM(1055)),
ABCDO983
6 (PXMC2P, COM(1059))
EQUIVALENCE (PSIXB(1), COMD(2761)), (DELXB(1, 1), COMD(2776)),
1 (ETAXB(1, 1), COMD(3001)), (NPSB, COMD(5360)),
ABCDO992
2 (NPTB(1), COMD(5361))
DATA AWORD /4HCOCO, 4HMB /
WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
IF (SI) GO TO 100
RA = .0252D0
AJ = 2.719D0
TMAX = 4000.0D0
TMIN = 100.0D0
GO TO 101
100 RA = 286.9D0
AJ = 1.0D0
TMAX = 2222.0D0
TMIN = 555.5D0
101 Q(2) = 0.0D0
Q(3) = 0.0D0
P3PSI = 14.696D0 * P3
IF (SI) P3PSI = .145040-3 * P3
WA3C = WA3 * DSQRT(T3) / P3PSI
```

IF (SI) WA3C = WA3 * DSQRT(T3) / P3
IF (IDES .EQ. 1) WA3CDS = WA3C
DPCOM = DPCODS * (WA3C / WA3CDS)
IF (DPCOM .GT. 1.0D0) DPCOM = 1.0D0
P4 = P3 * (1.0D0 - DPCOM)
IF (IDES .EQ. 1 .AND. MODE .EQ. 2) T4 = (TMAX + TMIN) / 2.0
IF (ITRAN .EQ. 1 .AND. MODE .EQ. 2) CALL FCTRL
1
IF (T4 .GT. TMAX) T4 = TMAX
IF (T4 .GE. TMIN) GO TO 2
T4 = TMIN
IF (MODE .EQ. 1) MAPEDG = 1
2
DTCO = T4 - T3
IF (SI) DTCO = DTCO * 9.0D0 / 5.0D0
P3PSIN = P3PSI
CALL SEARCH (-1.0D0, P3PSIN, DTCO, ETAB, DUMMY, PSIWB(1), NPSB)
1
DELXB(1,1), ETAWB(1,1), DUMBO(1,1), NPTB(1), 15, 15, IGO)
IF (IGO .EQ. 7) CALL ERROR
IF (IDES .NE. 1) GO TO 4
EFABC = ETABDS / ETAB
ETAB = ETABCF * ETAB
IF (SI) T4 = T4 * 9.0D0 / 5.0D0
HV = (((((( - .4594317 - 19 * rr1) - .2031 - 15) * T4 +
   - 2.2783643D-11) * T4 + .2051501D-07) * T4 - 2453116D-03) * T4 +
   - 2.9433296D-01) * T4 + .1845537D+05
IF (.NOT. SI) GO TO 3
T4 = T4 * 5.0D0 / 9.0D0
HV = HV * 2325.4295D0
3
CALL THERMO (P4, HA, T4, XX1, XX2, 0.0D0, 0)
FAR4 = (HA - H3) / (HV * ETAB)
IF (FAR4 .LT. 0.0D0) FAR4 = 0.0D0
WPBX = FAR4 * WA3
IF (MODE .NE. 2) GO TO 7
ERRW = (WPB - WPBX) / WPB
DIR = DSQRT(WPB / WPBX)
CALL APQIR (Q(1), T4, ERRW, 0.0D0, 20.0D0, 1.0D*TOLALL, DIR, T4, IGO)
GO TO (5, 8, 6, IGO)
5
T4 = T4T
GO TO 1
6
CALL ERROR
7
WFB = WPBX
IF (IDES .EQ. 1) WFBDS = WFB
8
CALL THERMO (P4, H4, T4, S4, XX2, 1, FAR4, 0)
WG4 = WFB + WA3
IF (VCONB .EQ. O.0D0) GO TO 21
Q(2) = 0.0D0
Q(3) = 0.0D0
WG4P = WG4
H4P = H4
P4DOT = DERIV(10, P4)
13
CALL THERMO (P4, H4, T4, S4, XX2, 1, FAR4, 0)
WG4 = WG4P - P4DOT * VCOMB / T4 / 1.4D0 / RA
U4 = H4 - AJ * RA * T4
U4DOT = DERIV(11, U4)
H4X = (WG4P * H4P - (WG4P - WG4) * U4 - U4DOT * P4 * VCOMB / T4 / ABCD1065
SUBROUTINE COCOMP
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI, DUMSPL, FXM2CP, APTFAN, FAN
COMMON /CONALL/ CON (1062)
COMMON /COMDAT/ COMD (5423)
DIMENSION WORD(2), ERR(9), CNXP(15), PRXP(15,15), WACXP(15,15),
1 ETAXP(15,15), NPTP(15)
DIMENSION Q(9), WLH(2,2), AWORD(2)
EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (MODE, COM(6))

END
1 \((\text{MAPEDG}, \text{COM}(22)), (\text{TOLALL}, \text{COM}(23)), (\text{ERR}(1), \text{COM}(24)), (\text{T2}, \text{COM}(25))\)
2 \((\text{COM}(92)), (\text{T3}, \text{COM}(152)), (\text{H3}, \text{COM}(153)), (\text{WA3}, \text{COM}(154))\)
3 \((\text{BLQ}, \text{COM}(193)), (\text{WA32}, \text{COM}(271)), (\text{T21}, \text{COM}(263)), (\text{H21}, \text{COM}(264))\)
4 \((\text{T3}, \text{COM}(152)), (\text{H3}, \text{COM}(153)), (\text{WA3}, \text{COM}(154))\)
5 \((\text{EAC}, \text{COM}(308)), (\text{CI}, \text{COM}(309)), (\text{WACI}, \text{COM}(310))\)
6 \((\text{AE}, \text{COM}(311)), (\text{BLI}, \text{COM}(312)), (\text{BLIHP}, \text{COM}(313))\)
7 \((\text{EAC}, \text{COM}(308)), (\text{CI}, \text{COM}(309)), (\text{WACI}, \text{COM}(310))\)
8 \((\text{AE}, \text{COM}(311)), (\text{BLI}, \text{COM}(312)), (\text{BLIHP}, \text{COM}(313))\)
9 \((\text{EAC}, \text{COM}(308)), (\text{CI}, \text{COM}(309)), (\text{WACI}, \text{COM}(310))\)
10 \((\text{AE}, \text{COM}(311)), (\text{BLI}, \text{COM}(312)), (\text{BLIHP}, \text{COM}(313))\)

11 \((\text{WACCP}, \text{COM}(296)), (\text{PRCDS}, \text{COM}(297)), (\text{ETACDS}, \text{COM}(298))\)
12 \((\text{PCBLC}, \text{COM}(302)), (\text{PCNCDS}, \text{COM}(303))\)
13 \((\text{PCBLID}, \text{COM}(305)), (\text{CNCP}, \text{COM}(306))\)
14 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
15 \((\text{PCBLID}, \text{COM}(305)), (\text{CNCP}, \text{COM}(306))\)
16 \((\text{PCNCDS}, \text{COM}(303))\)
17 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
18 \((\text{PCNCDS}, \text{COM}(303))\)
19 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
20 \((\text{PCNCDS}, \text{COM}(303))\)

21 \((\text{PCBLC}, \text{COM}(302)), (\text{PCNCDS}, \text{COM}(303))\)
22 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
23 \((\text{PCNCDS}, \text{COM}(303))\)
24 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
25 \((\text{PCNCDS}, \text{COM}(303))\)
26 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
27 \((\text{PCNCDS}, \text{COM}(303))\)
28 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
29 \((\text{PCNCDS}, \text{COM}(303))\)
30 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
31 \((\text{PCNCDS}, \text{COM}(303))\)
32 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
33 \((\text{PCNCDS}, \text{COM}(303))\)
34 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
35 \((\text{PCNCDS}, \text{COM}(303))\)
36 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
37 \((\text{PCNCDS}, \text{COM}(303))\)
38 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
39 \((\text{PCNCDS}, \text{COM}(303))\)
40 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
41 \((\text{PCNCDS}, \text{COM}(303))\)
42 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
43 \((\text{PCNCDS}, \text{COM}(303))\)
44 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
45 \((\text{PCNCDS}, \text{COM}(303))\)
46 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
47 \((\text{PCNCDS}, \text{COM}(303))\)
48 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
49 \((\text{PCNCDS}, \text{COM}(303))\)
50 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
51 \((\text{PCNCDS}, \text{COM}(303))\)
52 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
53 \((\text{PCNCDS}, \text{COM}(303))\)
54 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
55 \((\text{PCNCDS}, \text{COM}(303))\)
56 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
57 \((\text{PCNCDS}, \text{COM}(303))\)
58 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
59 \((\text{PCNCDS}, \text{COM}(303))\)
60 \((\text{PCNC}, \text{COM}(301)), (\text{PCBLC}, \text{COM}(302))\)
61 \((\text{PCNCDS}, \text{COM}(303))\)
IF (ZC .GT. 1.0D0) ZC = 1.0D0
CNCS = CNC
IF (ISPOOL .EQ. 1) GO TO 12
CALL SEARCH (ZC, CNC, PRC, WACC, ETAC, CNXP(1), NCNP, PRXP(1,1),
1 WACXP(1,1), ETAXP(1,1), NPFP(1), 15, 15, IGO)
GO TO 13
12 PRC = 1.0D0
ETAC = 1.0D0
WAC = WA21
WACC = WAC * Theta / DELTA
CNCS = 1.0D0
PRCCF = 1.0D0
13 IF (MODE .EQ. 1) GO TO 4
IF ((CNCS - CNCS) .GT. 0.5D0 * T) LALL * CNCS MAPEDG = 1
4 IF (IGO .EQ. 1 .OR. IGO .EQ. 2) WRITE (8,9) CNCS, WLH(1,ISO),
1 WLH(2, IGO)
WAC = WACC * DELTA / THETA
IF (IDES .NE. 1) GO TO 5
T21DS = T21
IF (ISPOOL .GE. 2) PRCDF = (PRCDS - 1.0D0) / (PRC - 1.0D0)
ETACCF = ETACDS / ETAC
IF (ISPOOL .EQ. 1) ETACCF = 1.0D0
WACCF = WACDS / WAC
WRITE (6,10) PRCDF, ETACCF, WACCF, T21DS
5 ETAC = ETACCF * ETAC
WAC = WACCF * WAC
WAC = WAC
IF (.NOT. DUMSPL .OR. PCBLID .NE. C.0DC .OR. .NOT. FAN) GO TO 6
WA22 = WAC
WAI = WA22
WA21 = WACC * WACCF
6 BLI = BLA2
WA32 = WAC
WA21 = WAC
WACC = WACC * WACCF
PCBLI = BLI / WAI
CALL WDUCT1
IF (PCBLID .EQ. 0.0D0) ERR(7) = (WAC - WAI) / WAC
IF (.NOT. FAN) ERR(8) = (WAC - WAC - BLF) / WAC
IF (IDES .EQ. 1 .AND. PCELID .EQ. 0.0D0) ERR(7) = 1.0D-4
CALL THCOMP (PRC, ETAC, T21, U21, S21, P21, T3, H3, S3, P3)
IF (VCOMP .EQ. 0.0D0) GO TO 21
Q(2) = 0.0D0
Q(3) = 0.0D0
H3F = H3
P3DOT = DERIV(B, P3)
13 CALL THEPMD (P3, H3, T3, S3, XX2, W, C, CDC, 0)
WAC = WACP - P3DOT * VCOMP / T3 / 1.4D0 / RA
H3 = H3 - AJ * HA * T3
AA3 = DERIV(9, H3)
H3X = (WACP * H3F - (WACP - WAC) * H3 - H3DOT * P3 * VCOMP / T3 /
1 RA) / WAC
EBSW = (H3 - H3X) / H3
DIR = DSQRT(DABS(H3 / H3X))
CALL AFOURR (Q(1), T3, ERW, 0.0D0, 20.0D0, 1.0D0 * TOLALL, DIR, T3T, IGO)
GO TO (19, 21, 20), IGO
19 T3 = T3T
GO TO 18
20 CALL ERROR
21 IF (PCBLIC .GT. 0.0D0) BLC = PCBLIC * WAC
WA3 = WAC - BLC
BLDU = PCBLDU * BLC
BLJB = PCBLJB * BLC
BLHP = PCBLHP * BLC
BLIP = PCBLIP * BLC
BLLP = PCBLLP * BLC
IF (MODE .NE. 1) GO TO 7
IF (DABS(CNC - CNCS) .LT. 1.0D0 * TOLALL * CNCS) GO TO 8
WRITE (8, 11) CNCS, CNCS
CALL ERROR
PCNC = 100.0D0 * THETA * CNC
CALL COCOMB
RETURN
11 FORMAT (19HO* * CNC WAS= , E15.?, 11H AND NOW= , E15.8, 12H CHECK PCNC, 112H INPUT$$$$$$)
END ABCD125

Subroutine CODUCT

SUBROUTINE CODUCT
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL SI, AFTFAN
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2), ERR (9)
DIMENSION Q(9), AWORD1(2), AWORD2(2)
EQUIVALENCE (WORD(1), COM(1)), (INDES, COM(3)), (IGASMX, COM(19))
1 (IDBDM, COM(11)), (IDCD, COM(13)), (IDSHOC, COM(15)),
2 (NOZFLT, COM(17)), (TOLALL, COM(23)), (ERR(1), COM(24)), (P1,
3 COM(33)), (H22, COM(34)), (AM23, COM(35)), (WA23DS, COM(36)),
4 (T23, COM(37)), (P23, COM(38)), (H23, COM(39)), (S23, COM(40)),
5 (A24, COM(41)), (T24, COM(42)), (H24, COM(43)), (S24, COM(44)),
6 (AM25, COM(45)), (T25, COM(46)), (P25, COM(47)), (H25, COM(48)),
7 (S25, COM(49)), (A28, COM(50)), (A28SAV, COM(51)),
8 (AM28, COM(52)), (V28, COM(53)), (TS28, COM(54)),
9 (PS28, COM(55)), (T28, COM(56)), (P28, COM(57)), (H28, COM(58)),
1 (S28, COM(59)), (A29, COM(60)), (A29SAV, COM(61)),
2 (AM29, COM(62)), (V29, COM(63)), (TS29, COM(64)),
3 (PS29, COM(65)), (T29, COM(66)), (P29, COM(67)), (H29, COM(68)),
4 (S29, COM(69)), (BYPASS, COM(70)), (WAD, COM(71)),
5 (WFD, COM(72)), (ETAD, COM(73)), (DPDNC, COM(74)),
6 (DPDUDS, COM(75)), (H3, COM(153)), (WAC, COM(191)),
7 (PCBLID, COM(305)), (WAI, COM(311)), (BLF, COM(316)), A9CD1275
9 (BLDU, COM(317)), (WAF, COM(313)), (W624, COM(321)), A9CD1277
1 (PAF24, COM(322)), (P22, COM(375)), (P24, COM(394)), A9CD1279
1 (U24, COM(392)), (VFDUCT, COM(403)), (JCODUC, COM(546)), A9CD1279
EQUIVALENCE (ITPAN, COM(1049)), (SI, COM(1055)), A9CD1280
1 (APTAN, COM(1060)) A9CD1281
DAIA AWORD1, AWORD2 /4HCOU, HCLE, HDNO7, HzL / A9CD1282
WORD(1) = AWORD1(1) A9CD1283
WORD(2) = AWORD1(2) A9CD1284
Q(2) = 0.00D A9CD1285
Q(3) = 0.00D A9CD1286
QGO = 0.00D A9CD1287
IF (SI) goto 100 A9CD1288
AJ = 774.25D0 A9CD1289
AJX = 2.719D0 A9CD1290
CSF = 411.217CD0 A9CD1291
G = 32.17449D0 A9CD1292
TSID = 518.67D0 A9CD1293
TDEL = 2000.00D A9CD1294
TMAX = 400.00D A9CD1295
BA = .0252D0 A9CD1296
GO TO 101 A9CD1297
100 AJ = 1.00D A9CD1298
AJX = 1.00D A9CD1299
CSF = 101325.00C A9CD1300
G = 1.00D A9CD1301
TSID = 234.15D0 A9CD1302
TDEL = 1111.00D A9CD1303
TMAX = 4222.00D A9CD1304
BA = 296.90D A9CD1305
GO TO 101 A9CD1306
101 GAJ2 = 2.00D * G * AJ A9CD1307
ICODUC = 0 A9CD1308
WAX = WAF - WAI - BLF A9CD1309
IF (PCBLID .EQ. 0.00D) WAX = WAF - WAC - BLF A9CD1310
IF (APTAN) WAX = WAF - BLF A9CD1311
WAD = WAX * BLDU A9CD1312
P23 = P22 A9CD1313
C*** DRY LOSS A9CD1314
H23 = (BLDU * H3 + WAX * H22) / WAD A9CD1315
CALL THERMO (P23, H23, T23, S23, XX2, 1, 0.00D, 1) A9CD1316
W23 = WAD * DSQT(T23) / P23 A9CD1317
IF (IDES .EQ. 1) W23DS = WA23C A9CD1318
BYPASS = (WAP - WAI) / WAI A9CD1319
IF (APTAN) BYPASS = WAF / WAI A9CD1320
DPDUC = DPDUDS * (WA23C / WA23DS) A9CD1321
IF (DPDUC .GT. 1.00D) DPDUC = 1.00D A9CD1322
P24 = P23 * (1.00D - DPDUC) A9CD1323
CALL PROCOM (0.00D, T23, XX1, XX2, XX3, XX4, PHI23, XX6) A9CD1324
IF (IGASMX .GT. 0) IDBURN = 0 A9CD1325
AM24 = AM23 A9CD1326
TS24 = T23 * 0.875D0 A9CD1327
1 DO 2 I = 1, 50 A9CD1328
CALL PROCOM (0.00D, TS24, CS24, AK24, CP24, REX24, PHIS24, HS24) A9CD1329
V24 = AM24 * CS24 A9CD1329
C***
HSCAL = H23 - V24 ** 2 / GA24
DELHS = HSCAL - HS24
IF (DAEB (DELHS) .LE. 1.0D0 * TOLAIL * HSCAL) GO TO 3
TS24 = TS24 + DELHS / CF24
ICODUC = 1
GO TO 11

C1 = P24 * DSQRT (G / (T2J * T)) * CAPSF
AK24M1 = AK24 - 1.0D0
AK24P1 = AK24 + 1.0D0
AKM1 = AK24M1 / 2.0D0
AKP1 = AK24P1 / 2.0D0
IF (IDES .NE. 1) GO TO 4
IF (GOGO .GT. 0.0D0) GO TO 4
ASTOA = AKF1 ** (AKF1 / AK24M1) * AM24 * (1.0D0 + AM1 ** 1 AM24 ** 2) ** (- AKP1 / AK24M1)
EQWCRT = DSQRT (G * AK24 / REX24 / AMI) / DSQRT (TSTD) / CAPSF) * 1 (2.0D0 / AK24P1) ** (AKP1 / AK24M1)
WA23CC = WP23C / DSQRT (TSTD)
A24 = 1.0D0 / ASTOA * WA23CC / EQWCS
GOGO = 1.0D0
GO TO (5,6,11), IGO

WQA = WAL / A24
WQAT = C1 * DSQRT (AK24 / REX24) * AM24 / (1.0D0 + AM1 ** AM24 ** 1 2) ** (AKP1 / AK24M1)
DIR = WQA / WQAT
FW = (WQA - WQAT) / WQA
CALL AQUI2 (Q1) AM24, LW, 0.0D0, 3.0D0, 1.0D0 * TOLAIL, DIR, AM24, IGO
GO TO (5,6,11), IGO

AM24 = AM24T
IF (AM24 .GT. 1.0D0) AM24 = 0.5D0
GO TO 1

PS24 = P24 / DEXP ((P23 - PHIS24) / PEX24)
IF (IBURST .GT. 0) GO TO 8

C*** NON-DUCT BURNING
T24 = T23
WFD = 0.0D0
PAR24 = 0.0D0
GO TO 17

IF (IDBUFPN .EQ. 2) T24 = T23 + TDEL
IF (T24 .GT. TMAX) T24 = TMAX
IF (T24 .LT. T23) T24 = T23

C*** DUCT BURNING
RH042 = CAPSF * PS24 / (AJ * REX24 * TS24)
PS42 = PS24
V42 = V24
Q(2) = 0.0D0
Q(3) = 0.0D0
IF (T24 .LT. T23) T24 = T23 * 1.001D0

C *** IF DESIRED, ENTER CALCULATIONS FOR ETAD HERE
IF (SI) T24 = T24 * 9.0D0 / 5.0D0
IF (.NOT. SI) GO TO 102

GO TO 11

T24 = T24 * 5.0D0 / 9.0D0
HV = HV * 2325.4295D0

102 CALL THERMO (P24, HA, T24, XX1, XX2, 0, 0.0D0, 0)
PAR24 = (HA - H23) / (HV * ETAD)
IF (PAR24 .LT. 0.0D0) PAR24 = 0.0D0
WFDX = PAR24 * WAD
IF (IDBURN .NE. 2) GO TO 12
ERRW = (WFD - WFDX) / WFD
DIR = DSQRT(WFD / WFDX)
CALL AFQUIR (Q(1), T24, ERRW, 0.0D0, 20.0D0, 1D0*TOALL, DIR, T24, T60)
ICODUC = 3
GO TO (10, 13, 11), IGN

10 T24 = T24T
GO TO 9

11 CALL ERROR

12 WFD = WFDX

C*** MOMENTUM LOSS

13 WGD4 = WFD + WAD
CALL PROCOM (PAR24, T24, XX1, XX2, XX3, REX24, PHI24, H24)
RH024 = CAPSF * P24 / (AJ * REX24 * T24)
V24 = WG24 / (RH024 * A24)
Q(2) = 0.0D0
Q(3) = 0.0D0
PS24 = PS42 - 0.01D0

14 RH024 = WG24 / (V24 * A24)
HS24 = H24 - V24 ** 2 / GAJ2
CALL THERMO (1.0D0, HS24, TS24, PHI24, XX1, XX2, 1, PAR24, 1)
IF (TS24 .GE. 301.000) GO TO 15
CALL THERMO (1.0D0, HS24, 40.0D0, PHI24, XX1, XX2, 1, PAR24, 1)
V24 = DSQRT(GAJ2 * (H24 - HS24))
GO TO 14

15 PS24 = PH024 * AJ * REX24 * TS24 / CAPSF
PS24A = PS42 + (RH024 * V24 ** 2 - RH024 * V24 ** 2) / (G * CAPSF)
DIR = DSQRT(DABS(PS24 / PS24A))
EP = (PS24 - PS24A) / PS24
CALL AFQUIR (Q(1), V24, EP, 1.0D0, 0.0D0, 1.0D0*TOALL, DIR, V24T, IGN)
V24 = V24T
IF (V24 .LT. 25.0D0) V24 = 25.0D0
ICODUC = 4
GO TO (14, 16, 11), IGN

16 P24 = PS24 * DEXP((PHI24 - PHI324) / REX24)
CALL PROCOM (PAR24, TS24, CS34, XX1, XX2, XX3, XX4, XX5, XX6)
AM24 = V24 / CS24

17 CALL THERMO (P24, H24, T24, S24, XX1, 1, PAR24, 0)
WGD4 = WFD + WAD
IF (VPDUDT .EQ. 0.0D0) GO TO 31
Q(2) = 0.0D0
Q(3) = 0.0D0
WG24P = WG24
H24P = H24
P24DOT = DEXP(20, P24)

21 CALL THERMO (P24, H24, T24, S24, XX1, 1, PAR24, 0)
WG24 = WG24P - P24DOT * VPDUDT / T24 / (1.4D0 * RA)
U24 = H24 - AJX * RA * T24

66
U24DOT = DERIV(21, U24)
H24X = (WG24P * H24P - (WG24P - WG24)) * U24 - U24DOT * P24
1 VFDUCT / T24 / RA) / WG24
ERRW = (H24 - H24X) / H24
DIR = DSQRT(DABS(H24 / H24X))
CALL AFQUR (Q(1), T24, ERRW, 0.0D0, 20.0D0, 1D0*TOLALL, DIR, T24T, IGO)
ICODUC = 5
IGO = 29, 31, 30, IGO
T24 = T24T
IGO = 28
CALL ERROR
T25 = T24
P25 = P24
H25 = H24
S25 = S24
AM25 = AM24
IF (IGASNX .GE. 0) GO TO 21
WORD(1) = WORD2(1)
WORD(2) = WORD2(2)
A28SAV = A28
A29SAV = A29
NOZD = 0
IDNOZ = 0
IF (NOZFLT .EQ. 2) GO TO 19
IF (IDES .EQ. 1) GO TO 19
GO TO (15, 19, 19, 11)
CALL CONVRS (T25, H25, P25, S25, AM24, WG24, 1, IDNOZ, A28, A29, A29, AM29)
GO TO (19, 19, 19, 11)
IF (IDCD .EQ. 1) GO TO 19
CALL CONDIV (T25, H25, P25, S25, AM24, WG24, 1, IDNOZ, A28, A29, A29, AM29)
2 AM29, ICON
IDSHOC = ICON
ICODUC = 6
IGO = 20, 26, 26, 11, ICON
T29 = T28
H29 = H28
P29 = P28
S29 = S28
TS29 = TS28
PS29 = PS28
V29 = V28
AM29 = AM28
A29 = A29
IDSHOC = ICON + 3
ERR (5) = (P25R - P25) / P25R
IF (IDNOZ .EQ. 1) WFIT (6, 22) A28, AM28, A29, AM29
ICODUC = 0
CALL FAST9K
RETURN
20 FORMAT (9H0DUCT NOZZLE DESIGN, 5X, 8H.A28 = F15.P.

67
Subroutine COFAN

IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI, PXM2CP
COMMON /COMAIL/ COM(1062)
COMMON /COMDAI/ COM(5423)
DIMENSION WORD(2), CNXF(15), PRXF(15,15), WACXF(15,15)
1 ETA(15,15), NTF(15), DUMD(15)
DIMENSION Q(3), WLH(2,2), AWORD(2)
EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (IDES, COM(4)),
1 (MODE, COM(6)), (INIT, COM(7)), (MAPEDG, COM(22)), (TOLALL, COM(107)),
2 (COM(23)), (H22, COM(34)), (T2DS, COM(91)), (T2, COM(92)), (P2, COM(1504)),
3 COM(93)), (H2, COM(94)), (S2, COM(95)), (S22, COM(96)), (T22DS, COM(1505)),
4 COM(97)), (T4GU, COM(100)), (RUDS, COM(101)), (PRFCF, COM(121))
5 (ETAPCF, COM(122)), (WAPCF, COM(123)), (PRFDS, COM(124))
6 (PRFDS, COM(125)), (ETAPDS, COM(126)), (WAPDS, COM(127))
7 (PCNCGU, COM(128)), (PFP, COM(131)), (ETAF, COM(132))
8 (ZCDS, COM(133)), (CNP, COM(134)), (WAPF, COM(135))
9 (ZP, COM(136)), (PCNF, COM(137)), (PCHELP, COM(138))
1 (ZI, COM(139)), (PCNI, COM(140)), (ZIDS, COM(146))
2 (PCNIDS, COM(147)), (PCNCGU, COM(149)), (T4, COM(156))
3 (WPBDS, COM(185)), (WPE, COM(192)), (T21, COM(263))
4 (T21DS, COM(267)), (T22, COM(268)), (ZC, COM(300))
5 (PCNC, COM(301)), (PCNCDS, COM(303)), (BLF, COM(316))
6 (WAP, COM(319)), (P22, COM(375)), (U22, COM(376))
7 (VFAN, COM(395)), (DUMD(1), COM(405)), (WAPP, COM(420))
8 (JTRAN, COM(1050)), (SI, COM(1555)), (PMXCP2, COM(1059))
EQUIVALENCE (CNXF(1,1), COM(1)), (PRXF(1,1), COM(16))
1 (WACXF(1,1), COM(241)), (ETA(1,1), COM(466))
2 (NCNF, COM(5296)), (NPT(1), COM(5297))
DATA AWORD, WLH /4H COP, 4HAN , 4H (LO, 4H) , 4H (HI, 4H) /
WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
IF (SI) GO TO 100
TSTD = 518.668D0
PSTD = 1.0D0
RA = .0252D0
AJ = 2.719D0
GO TO 101
100 TSTD = 288.149D0
PSTD = 101325.0D0
RA = 286.9D0
AJ = 1.0D0
GO TO 101
101 THETA = DSQRT(T2 / TSTD)
DELTA = P2 / PSTD
IF (IDES .NE. 1) GO TO 1
WAPDS = WAPF * DELTA / THETA
1 CNP = PCNF / (100.0D0 * THETA)
IF (ZP .LT. 0.000) ZP = 0.00D0
IF (ZP .GT. 1.000) ZP = 1.00D0
CNFS = CNF
CALL SEARCH (ZF, CNF, PRF, WAF, ETA, CNXP(1), CNF, PRFX(1, 1),
1 WACXF(1, 1), ETA, CNF(1, 1), NFDF(1), 15, 15, IGO)
IF ((CNF - CNFS) GT 5D0 * TOLALL * CNF) MAPEDG = 1
IF (IGO .EQ. 1 OR. IGO .EQ. 2) WRITE (8,12) CNFS, WLH(1,IGO).
1 WLH(2,IGO)
WAF = WAPC * DELTA / THETA
IF (IDES .NE. 1) GO TO 2
PRFCF = (PRFDS - 1.0D0) / (PPF - 1.0D0)
ETAFCF = ETAFCF / ETA
WAFCF = WAFCS / WAF
WRITE (6,13) PRFCF, ETAFCF, WAFCF, T2DS
ETAF = ETAFCF * ETADF
WAF = WAPCF * WAF
WAPF = WAPF
WAPC = WAPC
PCNP = 100.0D0 * THETA * CNF
DUM1(1) = PCNF
CALL THCOMP (PRF, ETA, T2, H2, S2, P2, T2, H2, S2, P2)
IF (VPAN .EQ. 0.0D0) GO TO 21
Q(2) = 0.0D0
Q(3) = 0.0D0
H22P = H22
P22DOT = DERIV(4, P22)
18 CALL THERMO (P22, H22, T2, S22, XX2, 0.0D0, 0)
WAF = WAFP - P22DOT * VFAN / T2 / 1.4D0 / RA
U22 = H22 - AJ * RA * T22
U22DOT = DERIV(5, U22)
H22X = (WAFP * H22P - (WAFP * WAF) * H22 - U22DOT * P22 * VFAN /
1 T22 / RA) / WAF
ERRW = (H22 - H22X) / H22
DIR = DSOQT(DABS(H22 / H22X))
CALL AFQUIR (Q(1), T22, ERRW, 0.0D0, 29.0D0, 1D0 * TOLALL, DIR, T22, IGO)
GO TO (19, 21, 20), IGO
19 T22 = T22T
GO TO 18
20 CALL ERROR
21 IF (PCBLF .GT. 0.0D0) BLP = PCBLF * WAF
IF (JDES .EQ. 1) GO TO 9
JDES = 1
IF (INIT .EQ. 1) GO TO 8
IF (IDES .EQ. 1) GO TO 6
IF (JTBAN .EQ. 1) GO TO 9
IF (MODE .NE. 2) GO TO 3
T4 = GUESS(3, Y1, Y2, PCNF, PCNFDS, WBF, WBFDS, Y7, Y8, T4DS)
PCNF = GUESS(8, T4, T4DS, Y3, Y4, Y5, Y6, T22, T22DS, PCNFDS)
PCNC = GUESS(4, Y1, Y2, PCNF, PCNFDS, WBF, WBFDS, Y7, Y8, PCNCDS)
GO TO 7
3 IF (MODE .EQ. 1) GO TO 5
IF (MODE .EQ. 0) GO TO 4
T4 = GUESS(7, Y1, Y2, PCNF, PCNFDS, Y5, Y6, T2, T2DS, T4DS)
4 PCNC = GUESS(5, T4, T4DS, Y3, Y4, Y5, Y6, T22, T22DS, PCNCDS)
IF (FXHZCP) PCNC = PCNCDS * 3D0 * THETA
PCNC1 = PCNC
PCNC2 = PCNCDs
PCNI = GUESS(9, Y1, Y2, PCNC1, PCNC2, Y5, Y6, T22, T22DS, PCNIDS)
GO TO 7

5 T4 = GUESS(6, Y1, Y2, PCNC, PCNCD5, Y5, Y6, T22, T22DS, T4DS)
PCNI = GUESS(8, T4, T4DS, Y3, Y4, Y5, Y6, T22, T22DS, PCNIDS)
GO TO 7

6 T4 = T4DS
WFE = WPDS
T21DS = T21

7 ZC = ZCDS
ZI = ZIDS
PCNGU = PCNI
PCNGU = PCNC
T4GU = T4

8 INIT = 0
IF (MODE .NE. 3) GO TO 10
IF (DABS(CNF - CNFS) .LE. 1.0D0 * TOLALL * CNFS) GO TO 11
WRITE (8, 14) CNPS, CNF
CALL ERROR
PCNF = 100.0D0 * TSETA * CNF
CALL CONTINUE
RETURN

12 FORMAT (1EH0* * * CNF OFF MAP, F10.4, 2X, A4, A2, 11H* * *$$*$$)
13 FORMAT (11H) FAN DESIGN, 13X, 8H PRFCF = .E15.8, PH ETAFCF = .E15.8,
18 WAFCF = .E15.8, 9H T2DS = .E15.8)
14 FORMAT (1040CNF WAS = , E15.3, 11H AND NOW = , E15.8, 12H CHECK PCNF,
112H INPUT$$$$$$)
END

Subroutine COHPTB

SUBROUTINE COHPTB
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL SI, DUMSPL, FXFN2Y
COMMON /CONALL/ COM(1062)
COMMON /COMDAT/ COMD(5423)
DIMENSION WORD(2), SFP(9), TFFXH(15), CNXH(15, 15), DHTCHH(15, 15),
1 ETATXH(15, 15), NPTTFH(15)
DIMENSION Q(9), AWORD(2), WLO(2), WHI(2)
EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (NOMAP, COM(29)),
1 (TOLALL, COM(23)), (EFT(1), COM(24)), (TS, COM(102)), (HS, COM(103)),
2 COM(103), (SS, COM(104)), (WS, COM(15)), (FARS, COM(106)),
3 (TPHCF, COM(114)), (CNHPCF, COM(115)), (ETHPCF, COM(116)),
4 (DHPCF, COM(117)), (TFHPS, COM(118)), (CNNPS, COM(119)),
5 (ETHPS, COM(120)), (HEXT, COM(129)), (TFFTP, COM(141)),
6 (CNIP, COM(142)), (ETATIP, COM(143)), (DHTCIP, COM(144)),
7 (BTH, COM(145)), (H3, COM(153)), (T4, COM(156)), (HS, COM(157)),
8 (S4, COM(159)), (WH4, COM(159)), (FAR4, COM(160)),
9 (T50, COM(161)), (H50, COM(162)), (SS0, COM(163)),
1 (WS50, COM(164)), (FARS0, COM(165)), (CNHF, COM(171)),
2 (ETATHP, COM(172)), (DHTCHF, COM(173)), (DHTC, COM(174))
3 (TFFHP, COM(175)), (H21, COM(264)), (PCNC, COM(301)), A B C D 1643
4 (BLHP, COM(313)), (XNHP, COM(372)), (P4, COM(381)), A B C D 1649
5 (P50, COM(383)), (U50, COM(384)), (P5, COM(385)), A B C D 1650
6 (VHPTPB, COM(399)), (WACP, COM(422)), (XNHPDS, COM(423)), A B C D 1651
7 (PMHHP, COM(426)), (ISPOOL, COM(1044)), (SI, COM(1055)), A B C D 1652
8 (DUMSPL, COM(1057)), (FXPN2M, COM(1058)), A B C D 1653
EQUIVALENCE (TFFXH(1), COMD(3226)), (CNXH(1,1), COMD(3241)), A B C D 1654
1 (DHTCXH(1,1), COMD(3466)), (ETATXH(1,1), COMD(3691)), A B C D 1655
2 (NTFFSH, COMD(3736)), (NPTFFH(1), COMD(3777)), A B C D 1656
DATA AWORD, WLO, WHI /4HC3HP, 4HTB, *H (LO, 411), 4H (HI, 1 4H) /
WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
IF (SI) GO TO 100
RA = .0252DO
AJ = 2.719DO
CONFAC = 1.409D-5
GO TO 101
100 RA = 286.9D0
AJ = 1.000D0
CONFAC = 1.0966D-2
IF (ISPOOL .EQ. 1) GO TO 8
THDE = DSQRT(T4) / PCNC
IF (IDES .EQ. 0) GO TO 1
CNHPCF = CNHPCNS * THDE
1 CNHP = CNHPCF / THDE
CNHPS = CNHP
TFFHPS = TFFHP
CALL SEARCH (-1.0D0, TFFHP, CNHP, DHTCP, ETATHP, TFFXH(1), NTFFSH,
1 CNXH(1,1), DHTCXH(1,1), ETATXH(1,1), NPTFFH(1), 15, 15, 15, 15, 15
1 IF (IGO .EQ. 1) GO TO 11
IF (IGO .EQ. 11) IF (ISPOOL .EQ. 1 AND IGO .EQ. 15)
1 WRITE (8,9) TFFHPS, WLO
1 IF (IGO .EQ. 2) IF (ISPOOL .EQ. 1 AND IGO .EQ. 12)
1 WRITE (8,9) TFFHPS, WHI
1 IF (IGO .EQ. 10) IF (ISPOOL .EQ. 1 AND IGO .EQ. 15, 15, 15, 15, 15
1 WRITE (8,10) CNHPS, WLO
1 IF (IGO .EQ. 20) IF (ISPOOL .EQ. 1 AND IGO .EQ. 15, 15, 15, 15, 15
1 WRITE (8,10) CNHPS, WHI
1 IF (IGO .NE. 7) GO TO 2
CALL ERROR
RETURN
2 NOMAP = 0
TFCALC = WG4 * DSQRT(T4) / (14.596D0 * P4)
BTUEXT = 0.706705D0 * HPEXT
IF (.NOT. SI) GO TO 102
TFCALC = WG4 * DSQRT(T4) / P4
BTUEXT = HPEXT
102 XNHP = XNHPDS / PCNC / 100.0D0
XNHDOT = DERIV(XNHP)
DHTCC = (BTUEXT * WACP * (H3 - R21) + CONFAC * PMHHP * XNHP * XNHDOT) / (WG4 * T4)
IF (IDES .EQ. 0) GO TO 5
TFCALC = TFCALC / TFCALC
DHTCPF = DHTCC / DHTCP
RETURN
ETHPCF = ETHPDS / ETHATP
WRITE (6,11) CNHPCF,TFHPCF,ETHPCF,DHHPCF
5
TFHCAL = TFHPCF * TFHCAL
DHTCHP = DHHPCF * DHTCHP
ETHATP = ETHPCF * ETHATP
DHTC = DHTCC * T4
ERR(1) = (TFHCAL - TFHPH) / TFHCAL
ERR(2) = (DHTCC - DHTCHP) / DHTCC
CALL THTURB (DHTC,ETHATP,FAR4,H4,S4,P4,T50,H50,S50,P50)
IF (BLAP .LE. 0.000) GO TO 6
FAR50 = FAR4 * WG4 / (WG4 + BLHP * (FAR4 + 1.000))
WG50 = WG4 + BLHP
H50 = (BLHP * H3 + WG4 * H50) / WG50
CALL THERMO (P50,H50,T50,S50,XX2,1,FAR50,1)
GO TO 7
6
FAR50 = FAR4
WG50 = WG4
7
IF (VHPTRB .EQ. 0.000) GO TO 21
Q(2) = 0.000
Q(3) = 0.000
WG50P = WG50
H50P = H50
P50DOT = DERIV(12,P50)
18
CALL THERMO (P50,H50,T50,S50,XX2,1,FAR50,0)
WG50 = WG50P - P50DOT * VHPTRB / T50 / 1.400 / RA
U50 = H50 - RA * AJ * T50
U50DOT = DERIV(13,U50)
H50X = (WG50P * H50P - (WG5CP - WG50) * U50 - U50DOT * P50 * 1
VHPTRB / T50 / RA) / WG50
ERR5 = (H50 - H50X) / H50
DIR = DSQRT(DABS(H50 / H50X))
CALL AFQDIR (Q(1),T50,ERPw,0.000,20.000,1.000*TOLALL,DIR,T50,T50,IGO)
GO TO (19,21,20), IGO
19
T50 = T50T
GO TO 18
20
CALL ERROR
21
IF (FXFN2M .OR. DUMSPL) GO TO 8
CALL COIPTB
RETURN
8
P5 = P50
H5 = H50
T5 = T50
S5 = S50
FAR5 = FAR50
WG5 = WG50
C
SET MIDDLE TURBINE PARAMETERS TO ZERO, NOT USED
TFFIP = 0.000
CNIP = 0.000
DHTIP = 0.000
ETATIP = 0.000
CALL COLPTB
RETURN
C
Subroutine COINLT

IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2)
DIMENSION AWORD(2)
EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (MODE, COM(6)),
1 (INIT, COM(7)), (IAMTP, COM(9)), (TOLALL, COM(23)), (P1, ABCD1759
2 COM(33)), (T2DS, COM(91)), (T2, COM(92)), (P2, COM(93)), (H2, ABCD1771
3 COM(94)), (S2, COM(95)), (TOLDS, COM(101)), (PCNFDs, COM(124)),
4 (ZP, COM(135)), (PCNP, COM(137)), (T1, COM(149)), (H1, COM(150)), ABCD1773
5 (SI, COM(151)), (T4, COM(156)), (WP3DS, COM(185)), (ZFDs, ABCD1776
6 COM(186)), (ETAR, COM(187)), (WP5, COM(192)), (CS, COM(194)), ABCD1775
7 (AM, COM(195)), (ALT, COM(196)), (PCNPGU, COM(199)), ABCD1777
8 (PCNC, COM(301)), (PCNCDS, COM(303)), (DELT1, COM(429)), ABCD1779
9 (SI, COM(1055))
DATA AWORD /4HCOIN, 4HLT /
WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
IF (SI) GO TO 10
AJ = 778.26D0
G = 32.17404900
REP59 = 2.0855531D07
R = 1.986375D0
TSTD = 518.668D0
GO TO 11
10 AJ = 1.0D0
G = 1.0D0
REP59 = 6.3567658D06
R = 8314.34D0
TSTD = 288.149D0
11 ALT = ALTP * REP59 / (REP59 - ALTP)
GAJ2 = 2.0D0 * G * AJ
CALL ATMOS (ALT, T1STD, XX1, XX2, XX3, DELTA, CS, XX4, IIER)
P1 = DELTA
IF (SI) P1 = 101325.0D0 * DELTA
T1 = T1STD
IF (IAMTP .EQ. 2) T1 = T1STD + DELT1
IF (IAMTP .EQ. 5) CALL RAM2 (AM, ETAR)
IF (IAMTP .NE. 1 .AND. IAMTP .NE. 5) CALL RAM (AM, ETAR)
PAR = 0.0D0
CALL PROCOM (PAR, T1, CS, XX2, XX3, R1, PHI1, H1)
S1 = PHI1 - R1 * DLOG(DELTA)
H2 = H1 + (AM * CS) ** 2 / GAJ2
p2r = 1.0d0
if (s1) p2r = 101325.d0
do : i = 1,50
  call thermo (p2t,h2,t2,t2,s2t,aw,0,0.0d0,1)
  if (dabs(s2t - s1) .le. .1d0 * tolall * s1) go to 2
  p2t = p1 * dexp((aw / r) * ((s2t - s1) + (r / aw) *
  1 dlog(p2t / p1)))
call error
return
endif
if (iamp .eq. 3 .or. iamp .eq. 4) etaf = p2 / p2t
etar = p2 / p2t
if (iamp .eq. 4) etar = 1
  call thermo (p2,h2,t2,s2t,aw,0.0d0,1)
endif
if (ides .eq. 1) go to 3
if (mode .eq. 3) go to 4
  pcf = guess (mode,t4,'t&i')s, pcf, pcfds, w,uf90s,t2, t2ds, pcfds)
  pcf = pcf
  go to 4
  pcf = pcf * dsqrt(t2 / tstd)
  pcf = pcf
  t2ds = t2
  zf = zfds
  return
end

Subroutine COINTC

SUBROUTINE COINTC
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI, DUMSPL, PXFN2M, APTFAV, FAN
COMMON /COMALL/ Com(1062)
COMMON /COMDAT/ Com(5423)
DIMENSION Word(2), CnXi(15), Prxi(15,15), WacXi(15,15),
  1 Etaxi(15,15), CnnXi(15), Prpxi(15,15),
  2 Etaxi(15,15), Nptxi(15), Nptxi(15)
DIMENSION Q(9), WLI(2,2), AWord(2)
EQUIVALENCE (Word(1), Com(1)), (ides, Com(3)), (mapdf, Com(22)),
  1 (tolall, Com(23)), (h2, Com(34)), (t2, Com(92)), (p2, Com(93)),
  2 (s2, Com(94)), (s2, Com(35)), (s22, Com(96)), (t2ds, Com(37)),
  3 (cf, Com(134)), (z1, Com(139)), (pcnf, Com(140)), (pcnids),
  4 (com(147)), (price, Com(175)), (pricea, Com(177)), (waicp),
  5 (com(178)), (pfids, Com(180)), (aXids, Com(181)), (wacids),
  6 (COM(182)), (wacds, Com(153)), (fri, Com(198)), (pri, Com(199)),
  7 (wac, Com(191)), (t21, Com(263)), (h21, Com(264)),
  8 (s21, Com(265)), (wa21, Com(255)), (h22, Com(266)),
  9 (wa22, Com(259)), (wa32, Com(271)), (pcbli, Com(304)),
  1 (pcmblid, Com(305)), (cwi, Com(309)), (waci, Com(310)),
  2 (pai, Com(311)), (blt, Com(312)), (blf, Com(316)),
  3 (waf, Com(319)), (p22, Com(375)), (p21, Com(377)),
  4 (u21, Com(378)), (rintc, Com(396)), (waip, Com(211)),
  5 (ispool, Com(1044)), (s1, Com(1055)), (dumsp, Com(1057)),
  6 (pxf2m, Com(1058)), (aptfan, Com(1060)), (fan, Com(1061))
74
EQUIVALENCE (CNXIN(1), COMD(691)), (PRXIN(1,1), COMD(706)),
1 (WACXIN(1,1), COMD(931)), (ETAXIN(1,1), COMD(1156)),
2 (CNXXI(1), COMD(1381)), (PRXXI(1,1), COMD(1396)),
3 (WACXXI(1,1), COMD(1621)), (ETAXXI(1,1), COMD(1846)),
4 (NCNI, COMD(5312)), (NPFI(1), COMD(5313)), (NCNXI, COMD(5328)),
5 (NPXI(1), COMD(5329))

DATA AWORD, WLH /HCOIN, HTC, H (LO, 4H), H (HI, 4H) /

IF (SI) GO TO 100

TSTD = 518.668D0
PSTD = 1.0D0
RA = .0252D0
AJ = 2.719D0
GO TO 101

100 TSTD = 288.149D0
PSTD = 101325.0D0
RA = 286.9D0
AJ = 1.0D0
GO TO 101

101 IF (.NOT. AFTPAN) GO TO 1

T22S = T22
H22S = H22
S22S = S22
P22S = P22
T22 = T2
H22 = H2
S22 = S2
P22 = P2

1 THETA = DSQRT(T22 / TSTD)
DELTA = P22 / PSTD
IF (.NOT. FAN) WAICD ==WAP- PLF
IF (IDES .NE. 1) GO TO 2
PCNI = PCNIDS * THETA
PHI = PRIDS
PCBLI = PCBLID
IF (FAN) GO TO 102
WAICDS = WAICD * THETA / DELTA
DUMSPL = .TRUE.

102 WACI = WAICDS
WAIDS = WACI * DELTA / THETA
ETAI = ETAIDS

2 IF (.NOT. FXFN29) GO TO 3
FAN AND MIDDLE SPOOL ROTATE AT SAME SPEED
SPDFAN = CNF * DSQRT(T2 / TSTD)
CNF = SPDFAN / Theta
PCNI = 100.0D0 * THETA * CNF
IF (IDES .EQ. 1) PCNIDS = PCNI / THETA

3 CNI = PCNI / (100.0D0 * THETA)
ZI = DMIN1(ZI, 0.0D0)
ZI = DMIN1(ZI, 1.0D0)
CNTS = CNI
IF (.NOT. DUMSPL) GO TO 4
CALL INDUMY (CNI, ZI, WAICDS, IDES)
CALL SEARCH (ZI, CNI, PHI, WACI, ETAI, CNXXI(1), NCNI, PRXI(1,1),
CALL SEARCH (ZI, CNI, PRI, WAI, ETAI, CNXIN(1), NCNI, PRXIN(1), 1)
1 WACXI(1,1), ETAXXI(1,1), NPTXI(1), 15, 15, IGO
GO TO 5
4 CALL SEARCH (ZI, CNI, PRI, WAI, ETAI, CNXIN(1), NCNI, PRXIN(1), 1)
1 WACXI(1,1), ETAXXI(1,1), NPTXI(1), 15, 15, IGO
5 IF ((CNI - CNIS) .GT. 5.0D0 * TOLALL * CNI) MAPEDG = 1
IF (IGO .EQ. 1 .OR. IGO .EQ. 2) WRITE (8,12) CNIS, WLH(1, IGO),
1 WLH(2, IGO)
6 IF (.NOT. FAN) WACI = WAI * THETA / DELTA
WA22 = WAI
IF (IDES .NE. 1) GO TO 7
T22DS = T22
IF (AFTFAN) T22DS = T22S
ETAICF = ETAICF / ETAI
WAICF = WAICF / WAI
PRICF = (PRIDS - 1.0D0) / (FFT - 1.0D0)
IF (.NOT. DUMSPL) GO TO 5
PRICF = 1.0D0
ETAICF = 1.0D0
WAICF = 1.0D0
WRITE (6,13) PRICF, ETAICF, WAICF, T22DS
7 PRI = PRICF * (PRI - 1.0D0) + 1.0D0
ETAI = ETAICF * ETAI
WAI = WAICF * WAI
WAIP = WAI
WACI = WACI * WAICF
WA22 = WAI
CALL THCOMP (PRI, ETAI, T22, H22, S22, P22, T21, H21, S21, P21)
IF (VINTC .EQ. 0.0D0) GO TO 21
Q(2) = 0.0D0
Q(3) = 0.0D0
H21P = H21
P21DOT = DERIV(6, P21)
19 CALL THERMO (P21, H21, T21, S21, XX2, 0.0D0, 0)
WAI = WAIP - P21DOT * VINTC / T21 / 1.4D0 / RA
U21 = H21 - AJ * RA * T21
U21DOT = DERIV(7, U21)
H21X = (WAIP * H21P - (WAIP - WAI) * H21 - U21DOT * P21 * VINTC / T21 / RA) / WAI
ERFW = (H21 - H21X) / H21
DIR = DSQRT(DABS(H21 / H21X))
CALL APQUR (Q(1), T21, ERFW, C.0D0, 20.0D0, 1.0D0*TOLALL, DIR, T21, IGO)
GO TO (19, 21, 20), IGO
19 T21 = T21T
GO TO 18
20 CALL ERROR
21 IF (.NOT. DUMSPL) GO TO 9
PRI = 1.0D0
ETAI = 1.0D0
T21 = T22
H21 = H22
S21 = S22
P21 = P22
IF (ISPOOL .EQ. 1) WA21 = WAI
IF (IDES .NE. 1) GO TO 9
BLI = PCCT * WAT
WA21 = WA22 - BLI
WA32 = BLI
IF (FAN .OR. IDES .EQ. 1) GO TO 10
WHITE (8,14) CNIS,CNI
CALL ERROR
PCKI = 100.000 * THETA * CNI
10 IF (.NOT. AFTFAN) GO TO 11
T22 = T22S
H22 = H22S
S22 = S22S
P22 = P22S
11 CALL COCOMP
RETURN
C
C
C 13 FORHAT (20HOMIDDLE SPOOL DESIGN,bX,RH  PRICF=,E15.8,8H  ETATCF=,11 FORMAT (10IiOCNI WAS=,E15.9,11H  ANI) WOW=,E15.9,1111  CHFCK PCNI,
14 FORMAT (19HO* * CNI OFF MAP,F10.4,2X,A4,A2,11H* *$$$$$$)
15 FORMAT (20HOMIDDLE SPOOL DESIGN,4X,8H  PRICF=,E15.8,8H ETATCF=,11 FORMAT (10HOCNI WAS= ,E15.9,11H AND NOW= ,E15.9,12H CHECK PCNI, 1 12H INPUT$$$$$$)
END

Subroutine COIPTB

SUBROUTINE COIPTB
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI, FXHZCP, AFTFAV
COMMON /COMALL/ COM(1062)
COMMON /COMDAT/ COMD(5423)
DIMENSION WORD(2), ERR(9), TPFXH(15), CNXH(15,15), DHTCHXH(15,15)
1 ETATXH(15,15), TPFXI(15), CNXI(15,15), DHTCXI(15,15)
2 ETATXI(15,15), NPTTFH(15), NPTTFI(15)
DIMENSION Q(9), AWORD(2), WLO(2), WHI(2)
EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (NOMAP, COM(27)),
1 (TOLALL, COM(23)), (ERR(1), COM(24)), (H22, COM(34)), (H2),
2 (COM(94)), (T5, COM(102)), (H5, COM(103)), (S5, COM(104)), (WG5,
3 COM(105)), (FARS, COM(106)), (TPHPDS, COM(118)), (CNPDS,
4 COM(119)), (ETHPD, COM(120)), (HPEXT, COM(129)), (PCNI,
5 COM(140)), (TPPIP, COM(141)), (CNIP, COM(142)), (ETATIP,
6 COM(143)), (DHTCP, COM(144)), (DHTI, COM(145)), (H3, COM(153)),
7 (T50, COM(161)), (H50, COM(162)), (S50, COM(163)),
8 (WG50, COM(164)), (FARS, COM(165)), (H21, COM(264)),
9 (TP1PS, COM(278)), (CNP3DS, COM(279)), (ETP3DS, COM(280)),
1 (TP1PCF, COM(285)), (CNP3PCF, COM(286)), (ETP3PCF, COM(287)),
2 (DHPICF, COM(288)), (BLIP, COM(314)), (XNP3, COM(373)),
3 (P50, COM(383)), (P5, COM(385)), (U5, COM(386)),
4 (VP1TRB, COM(400)), (WAIP, COM(421)), (WACP, COM(422)),
5 (XN1PS, COM(424)), (PMIIP, COM(427)), (SI , COM(1055)),
6 (FMS22CP, COM(1059)), (AFTFAN, COM(1060))
EQUIVALENCE (TPFXH(1), COMD(3226)), (CNXH(1,1), COMD(3241)),

77
1 (DHTCXH (1,1), COMD(3466)), (EFATXH (1,1), COMD(3691)), ABCD2017
2 (TFFXI(1), COMD(3916)), (CNXI(1,1), COMD(3931)), ABCD2013
3 (DHTCIX(1,1), COHD(4156)), (ETATXI(1,1), COHD(4381)), ABCD2019
4 (NTPFSH, COMD(5376)), (NPTTFH(1), COMD(5377)), ABCD2020
5 (NTFFSI, COMD(5392)), (NPTTFI(1), COMD(5393)), ABCD2021

DATA AWORD, WLO, WHI /4HCOIP, YHTB /4H (LO, 4H) /4H

IF (SI) GO TO 100
RA = .0252DO
AJ = 2.719D0
CONPAC = 1.4091D-5
GO TO 101

100 RA = 286.9D0
AJ = 1.0D0
CONPAC = 1.0966D-2

101 H2SAY = H2
IF (APFTAN) H22 = H2
WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
THDE = DSQRT(T50) / PCNI
IF (IDES .EQ. 0) GO TO 1
CNPCF = CNIPDS * THDE
IF (FXM2CP) CNPCF = CNHPDS * THDE

1 CNIP = CNIPCF / THDE
CNIPS = CNIP
TFIPS = TFIP
IF (FXM2CP) GO TO 2
CALL SEARCH (-1.0D0, TFFIP, CNIP, DHTCIP, ETATIP, TFFXI(1), NTPFSI,
1 CNXI(1,1), DHTCXI(1,1), ETAIXI(1,1), NPTTFI(1), 15, 15, IGO)
GO TO 104

2 CALL SEARCH (-1.0D0, TFFIP, CNIP, DHTCIP, ETATIP, TFFXI(1), NTPFSH,
1 CNXH(1,1), DHTCWXH(1,1), ETAIXH(1,1), NPTTFH(1), 15, 15, IGO)

104 IF (IGO .EQ. 1 .OR. IGO .EQ. 11 .OR. IGO .EQ. 21)
1 WRITE (8,9) TFIPS, WLO
IF (IGO = 2 .OR. IGO .EQ. 12 .OR. IGO .EQ. 22)
1 WRITE (8,9) TFIPS, WHI
IF (IGO .EQ. 3 .OR. IGO .EQ. 11 .OR. IGO .EQ. 12)
1 WRITE (8,10) CNIPS, WLO
IF (IGO .EQ. 5 .OR. IGO .EQ. 21 .OR. IGO .EQ. 22)
1 WRITE (8,10) CNIPS, WHI
IF (IGO .NE. 7) GO TO 3
CALL ERROR
RETURN

NOMAP = 0

1 TFICAL = WG50 * DSQRT(T50) / (10.69600 * P50)
IF (SI) TFICAL = WG50 * DSQFT(T50) / P50
1 WRITE (8,9) TFICAL, PCNI /100.0DO
XNIP = XNIPDS * PCNI / 100.0DO
XNIDOT = DERIV(2,XNIP)
BTUEXT = .7667050D * HPFEXT
1 WRITE (8,9) BTUEXI = HPFEXT
IF (SI) BTUEXI = HPFEXT
DHACEL = CCNFC * PMIIP * XNIP * XNIDOT
DHTIC = (WAIP * (H21 - H22) + DHACEL) / (WG50 * T50)
IF (FXM2CP) DHTIC = (BTUEXT * WACP * (H3 - H21) + WAIP *
1 (H21 - H22) + DHACEL) / (WG50 * T50)
IF (IDES .EQ. 0) GO TO 6
TFIPCF = TFIPDS / TFICAL
DHIPCIF = DHTIC / DHTCIP
ETIPCF = ETIPDS / ETATIP
IF (.NOT. FXM2CF) GO TO 102
TFIPCF = TFHPDS / TFICAL
ETIPCF = ETHPDS / ETATIP

102 WRITE (6,11) CNIPCF,TFIPCF,ETIPCF,DHIPCF

6 TFICAL = TFICPCF * TFICAL
DHTCIP = DHIPCF * DHTCIP
ETATIP = ETIPCF * ETATIP
DHTI = DHTIC * T50
N1 = 8
N2 = 9
IF (.NOT. FXM2CF) GO TO 103
N1 = 1
N2 = 2

103 ERR(N1) = (TFICAL - TFFIP) / TFICAL
ERR(N2) = (DHTIC - DHTCIP) / DHTIC
CALL THTURB (DHTI,ETATIP,FAR50,H50,S50,P50,T5,H5,S5,P5)
IF (BLIP .LE. 0.0) GO TO 103
FAR50 = FAR50 * W500 / (W500 + BLIP * (FAR50 + 1.0D0))
W50 = W500 + BLIP
H5 = (BLIP * H3 + W50 * H5) / W50
CALL THERMO (P5,H5,T5,S5,XX2,1,FAR5,1)
GO TO 8

7 FAR5 = FAR50
W50 = W500
IF (VIPTRB .EQ. 0.0D0) GO TO 21
Q(2) = 0.0D0
Q(3) = 0.0D0
W50P = W50
H5P = H5
P5DOT = DERIV(14,P5)

19 CALL THERMO (P5,H5,T5,S5,XX2,1,FAR5,0)
W50 = W50P - P5DOT * VIPTRB / T5 / 1.4D0 / RA
U5 = H5 - RA * AJ * T5
U5DOT = DERIV(15,U5)
H5X = (W50P * H5P - (W50P - W50) * U5 - U5DOT * P5 * VIPTRB / 1 T5 / RA) / W50
ERRW = (H5 - H5X) / H5
DIR = DSORT(DABS(H5 / H5X))
CALL AFQUIR (Q(1),T5,ERRW,3.0D0,20.0D0,1D0*TOLALL,DIR,TST,IGO)
GO TO (19,21,20), IGO

15 T5 = T5T
GO TO 18

20 CALL ERROR

21 H22 = H22SAV
CALL COLPTB
RETURN

C

9 FORMAT (19H0****TFIP OFF MAP,P10.4,2X,A4,A2,11H****$S$$S$)
Subroutine COLPTB

IMPLICIT REAL*8 (A-H, O-Z)

LOGICAL ST, FXNP2M, APFTP

COMMON /COMALL/ COM (1062)

COMMON /COMDAT/ COND (5023)

DIMENSION WORD (2), ERR (9), TFFXL (15), CNXL (15, 15), DHPCXL (15, 15)

1 ETAI XL (15, 15), NPTFL (15)

DIMENSION Q (9), AWORD (2), WLO (2), WHI (2)

EQUIVALENCE (WORD (1), COM (1)), (IDES, COM (3)), (NOMAP, COM (20)),
1 (TOLALL, COM (23)), (ERF (1), COM (24)), (H22, COM (34)), (H2, COM (94)),
2 (T5, COM (102)), (T5, COM (103)), (S5, COM (104)), (W5, COM (105)),
3 (PAS5, COM (165)), (HPEXT, COM (129)), (PCNP, COM (137)),
4 (H2, COM (153)), (H2, COM (264)), (E2, COM (272)), (E2, COM (273)),
5 (H5, COM (273)), (H5, COM (274)), (TFFLP, COM (275)),
6 (CNLPCF, COM (276)), (CNLPCF, COM (277)), (TFFLCF, COM (281)),
7 (CNLPCF, COM (282)), (CNLPCF, COM (283)), (DHPCXL, COM (284)),
8 (TFFLP, COM (289)), (CNLPDS, COM (290)), (CNLPDS, COM (291)),
9 (DHTCLP, COM (232)), (DHTCLP, COM (233)), (BLLP, COM (315)),
1 (W655, COM (323)), (PAS5, COM (324)), (XNLPCF, COM (274)),
2 (E5, COM (385)), (P55, COM (387)), (U55, COM (388)),
3 (WLPTRB, COM (401)), (WLPTRB, COM (402)), (WAIP, COM (421)),
4 (XNLPCF, COM (425)), (EMILP, COM (428)), (ISPCL, COM (1044)),
5 (SI, COM (1055)), (FXNP2M, COM (1058)), (APFTP, COM (1060)),
6 (IDS, COM (4506)), (CNXL (1, 1), COM (4621)),
7 (DHTCLP (1, 1), COM (4846)), (EIATXL (1, 1), COM (5071)),
8 (TFFFL, COM (5408)), (NPTFL (1), COM (5409)),
9 (NPTFL (1), COM (5409))

DATA AWORD, WLO, WHI /&HCOLP, 4HTS, /&HLO, 4H, 4H (HI, 1 4H) /

WORD (1) = AWORD (1)
WORD (2) = AWORD (2)

IF (SI) GO TO 100
RA = .0252D0
AJ = 2.719D0
CONFAC = 1.4091D-5
GO TO 101

100 RA = 286.9D0
AJ = 1.000
CONFAC = 1.0966D-2

THDE = DSQRT (225) / PCNP

IF (IDES .EQ. 0) GO TO 1

CNLPCF = CNLPCF * THDE

CALL SEARCH (-1.0D0, TFFLP, CNLP, DHTCLP, ETAI XL, TFFXL (1), TFFFL, 1
CNXL (1, 1), DHTCLP (1, 1), ETAI XL (1, 1), NPTFL (1), 15, 15, 1G3)

END
1 WRITE (6,8) TFFLPS, WLO
IF (IGO .EQ. 2 .OR. IGO .EQ. 12 .OR. IGO .EQ. 22)
1 WRITE (6,8) TFFLPS, WLO
IF (IGO .EQ. 10 .OR. IGO .EQ. 11 .OR. IGO .EQ. 12)
1 WRITE (6,9) CNLPS, WLO
IF (IGO .EQ. 20 .OR. IGO .EQ. 21 .OR. IGO .EQ. 22)
1 WRITE (6,9) CNLPS, WLO
IF (IGO .NE. 7) GO TO 2
CALL ENDOE
RETURN
TFLCAL = WG5 * DSQRT (T5) / (14.696DO * P5)
IF (P55) TFLCAL = WG5 * DSQRT (T5) / P5
XNLPS = XNLPS * PCNF / 100.0D0
XNLPS = DERIV (XNLPL, XNLPL)
DHTCF = (WAPP * (H22 - H2) + DAHEL) / (WG5 * T5)
DHTCF = DEXT + WAIP * (H21 - H22)
1/(WG5 * T5)
IF (PXPNM2 .AND. .NOT. AFTAN) DHTCF = DEXT + WAIP * (H21 - H22)
1/(WG5 * T5)
TFLCF = TFLCF / TFLCAL
DHTCLF = DHTCLF / DHTCLP
ETLPCF = ETLPCF / ETATLP
WRITE (6, 10) CNLPCF, TFLPCF, ETLPCF, DHTCLF
IF (I) = 3
I2 = 4
IF (ISPOOL .GE. 2) GO TO 11
BTUEXT = 0.706705DO * HPXT
IF (SI) BTUEXT = HPXT
DHTCF = DEXT + BTUEXT / (WG5 * T5)
IF (IDES .EQ. 0) GO TO 5
TFLPCF = TFLPCF / TFLCAL
DHTCLP = DHTCLP / DHTCLP
ETATLP = ETLPCF / ETATLP
DHTF = DHTCF * T5
I1 = 3
I2 = 4
IF (ISPOOL .NE. 1) GO TO 102
I1 = 1
I2 = 2
102 ERR (11) = (TFLCAL - TFFLPS) / TFLCAL
ERR (I2) = (DHTCF - DHTCLP) / DHTCF
CALL THTUB (DHTF, ETATLP, PAR5, H5, S5, P5, T5, H5, S5, P5, P55)
IF (BLLP .LE. 0.0D0) GO TO 6
PAR5 = PAR5 * WG5 / (WG5 + BLLP * (1.0D0 + PAR5))
WG55 = WG5 + BLLP
H55 = (BLLP * H5 + WG5 * H55) / WG55
CALL THERMO (P55, H55, T55, S55, XX2, 1, PAR55, 1)
GO TO 7
6 PAR5 = PAR5
WG55 = WG5
7 IF (VLPTRB .EQ. 0.0D0) GO TO 21
Q (2) = 0.0D0
Q (3) = 0.0D0
81
WG55P = WG55
H55P = H55
P55DOT = DERIV(16, P55)
19 CALL THERMO (P55, H55, T55, S55, XX2, 1, FAR55, 0)
WG55 = WG55P - P55DOT * VLPTRB / T55 / 1.4D0 / RA
U55 = H55 - RA * AJ * T55
U55DOT = DERIV(17, U55)
H55X = (WG55P * H55P - (WG55P - WG55) * U55 - U55DOT * P55 -
1 VLPTRB / T55 / RA) / WG55
ERRW = (H55 - H55X) / H55
DIR = DSQRT(DABS(H55 / H55X))
CALL APQUR (Q(1), T55, ERRW, 0.0D0, 20.0D0, 1D0*TOLALL, DIR, T55, IGO)
GO TO (19, 21, 20), IGO
19 T55 = T55T
GO TO 18
20 CALL ERROR
21 CALL FIROS
RETURN

FORMAT (19H0*****TPFD OFF MAP, F10.4, 2X, A4, A2, 11H0*****$7)$2
9 FORMAT (19H0***** CNLP OFF MAP, F10.4, 2X, A4, A2, 11H0*****$7)$2
10 FORMAT (20HOL TP, TURBINE DESIGN, 5X, 7HCNLPCF==, E15.8, 8H TPLPCF==,
1 E15.8, 8H ETLPCF==, E15.8, 8H DHLPCF==, E15.8)
END

Subroutine COMIX

SUBROUTINE COMIX
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL SI
COMMON /COMALL/ T55M (1062)
DIMENSION WORD (2), ERR (9), DMD1 (16)
DIMENSION QQ (9), AWORD (2)
EQUIVALENCE (WORD (1), COM (1)), (IDES, COM (3)), (MODE, COM (6)),
1 (IGASMX, COM (10)), (NMAP, COM (20)), (TOLALL, COM (23)), (PRF (1)),
2 (COM (24)), (A*K25, COM (45)), (T25, COM (85)), (P25, COM (47)), (H25,
3 COM (48)), (S25, COM (49)), (A25, COM (95)), (V25, COM (99)), (AYE,
4 COM (107)), (V55, COM (108)), (AY5, COM (109)), (PS55, COM (117)),
5 (S6, COM (111)), (P56, COM (112)), (V6, COM (113)),
6 (PRFDS, COM (125)), (2P, COM (136)), (PCMF, COM (137)),
7 (T55, COM (272)), (H55, COM (273)), (S55, COM (274)),
8 (PRCDS, COM (297)), (WG55, COM (321)), (PARL, COM (322)),
9 (WG55, COM (323)), (PFR55, COM (324)), (AY6, COM (327)),
1 (A6, COM (328)), (WG6, COM (330)), (AY, COM (331)), (P6, COM (332)),
2 (H6, COM (333)), (P55, COM (347)), (DMD1(1), COM (405)),
3 (FPRNEW, COM (991)), (PRCNEW, COM (992)), (ICOMIX, COM (1047)),
4 (KKGO, COM (1048)), (SI, COM (1055))
DATA AWORD /4H COM, 4HIX /
WORD (1) = AWORD (1)
WORD (2) = AWORD (2)
IF (SI) GO TO 100
AJ = 778.26D0
CAPSF = 2116.2170D0

82
G = 32.174049D0
RDEM = 1.986375D0
GO TO 101

100 AJ = 1.0D0
CAPSF = 1.0D0
G = 1.0D0
RDEM = 8.31641D0

101 GAJ2 = 2.0 * G * AJ
ICOMIX = 0
CALL PROC1M (PAR55, T55, XX1, XX2, XX3, XX4, PHI55, XX5)
CALL PROC1M (PAR24, T25, XX1, XX2, XX3, XX4, PHI25, XX5)
IF (IDES .EQ. 0) GO TO 12

C ***
CALCULATE A55 AND A25 WITH PS25 = PS55
IF (PS55 .GT. 0.0D0) GO TO 3
POP = PS55 / PS55
ALPOPI = DLOG (1.0D0 / POP)
TS55 = T55 * POP ** .286D0
DO 1 I = 1, 50
CALL PROC1M (PAR55, T55, CS55, AK55, CP55, RX55, PHI55, HS55)
PHIS = PHI55 - RX55 * ALPOPI
DELPHI = PHIS - PHI55
IF (DABS (DELPHI) .LE. .1D0 * TOLALL * PHIS) GO TO 6
TS55 = TS55 * DEXP (4.0D0 * DELPHI)
ICOMIX = 1
1
CALL ERROR
RETURN

3 TS55 = 0.875D0 * T55
DO 4 I = 1, 50
CALL PROC1M (PAR55, T55, CS55, AK55, CP55, RX55, PHI55, HS55)
V55 = AM55 * CS55
HSCAL = HS55 - V55 ** 2 / GAJ2
DELHS = HSCAL - HS55
IF (DABS (DELHS) .LE. .5D0 * TOLALL * HSCAL) GO TO 5
TS55 = TS55 + DELHS / CP55
ICOMIX = 2
4
GO TO 2

5 PS55 = PS55 / DEXP ((PHI55 - PHI55) / RX55)
IF (PS55 .GT. P25 .AND. IDES .EQ. 1 .AND. IGASMX .GT. 0) GO TO 4
IF (H55 .GT. HS55) GO TO 7
WRITE (8,46) PS55, PS55, T55, TS55, HS55, HS55
ICOMIX = 2

7 VS55 = DSQRT (GAJ2 * (HS55 - HS55))
RHO = CAPSF * PS55 / (AJ * RX55 * TS55)
A55 = WG55 / (RHO * V55)
AM55 = V55 / CS55
IF (IGASMX .GT. 0) GO TO 9
WRITE (6,47) A55, AM55
IF (IGASMX) 35, 41, 8

8 PS25 = PS55
POP = PS25 / P25
ALPOPI = DLOG (1.0D0 / POP)
TS25 = T25 * POP ** .286D0
DO 9 I = 1, 50
CALL PFOC3M (FA24,TS25,CS25,a25,~~25,~~X25,PHIS25,HS25)

PHIS = PHI25 - REX25 * ALPOPI

DELPHI = PHIS - PHIS25

IF (DABS(DELPHI) .LE. .1D0 * TOLALL * PHIS) GO TO 10

TS25 = TS25 * DEXP(4.0D0 * DELPHI)

ICOMIX = 4

GO TO 2

10

IF (H25 .GT. HS25) GO TO 11

WRITE (9,48) P25,~~25,T25,rsS5,H25,~~~5

ICOMIX = 5

GO TO 2

9

CALL ERROR

RHO = CAPSF * PS25 / (AJ * REX * TS25)

A25 = WG24 / (RHO * V25)

AM25 = V25 / CS25

WRITE (6,49) 855,AM55,A25,AM25

GO TO 27

11

V25 = DSQRT(GAJ2 * (H25 - HS25))

RHO = CAPSF * PS25 / (AJ * REX25 * TS25)

A25 = WG24 / (RHO * V25)

AM25 = V25 / CS25

WRITE (6,49) A55,AM55,A25,AM25

GO TO 27

12

C ***

CALCULATE PS55 AND PS25

WQAT = WG55 / A55

C1 = P55 * DSQRT(G / (T55 * AJ)) * CAPSF

MCON = 0

QQ(2) = 0.0D0

QQ(3) = 0.0D0

AM55 = 0.50D0

TS55 = 0.875D0 * T55

DO 14 I = 1,50

CALL PROCOM (FA55,TS55,CS55,AK55,CP55,REX55,PHIS55,HS55)

V55 = AM55 * CS55

HSCAL = H55 - V55 ** 2 / GAJ2

DELHS = HSCAL - HS55

IF (DABS(DELHS) .LE. .5D0 * TOLALL * HSCAL) GO TO 15

TS55 = TS55 + DELHS / CP55

ICOMIX = 6

GO TO 2

15

WQAT = C1 * DSQRT(AK55 / REX55) * AM55 / (1.0D0 + (AK55 - 1.0D0) * AM55 ** 2 / 2.0D0) ** ((AK55 + 1.0D0) / (2.0D0 * (AK55 - 1.0D0)))

AMX = AM55

IGOGO = 0

16

DIR = WQA / WQAT

EW = (WQA - WQAT) / WQA

CALL APQUIR (QQ(1),AMX,EW,0.0D0,30.0D0,.5D0*TOLALL,DIR,AMXT,ICON)

ICOMIX = 7

GO TO (17,22,2), ICON

17

IF (AMXT .LE. 1.0D0) GO TO 20

AMXT = 0.7D0

MCON = MCON + 1

IF (MCON .LE. 1) GO TO 20

IF (MODE .EQ. 3) GO TO 19

PCNF = DUMD1(1)

WRITE (9,50) PCNF,AMX,P55,PS55,P25,PS25

PCNF = 1.01D0 * PCNF

DUMD1(1) = PCNF

10 MAP = 7

ICOMIX = 0

ABCD2335

ABCD2336

ABCD2337

ABCD2339

ABCD2339

ABCD2340

ABCD2341

ABCD2342

ABCD2343

ABCD2344

ABCD2345

ABCD2346

ABCD2347

ABCD2348

ABCD2349

ABCD2350

ABCD2351

ABCD2352

ABCD2353

ABCD2354

ABCD2355

ABCD2356

ABCD2357

ABCD2358

ABCD2359

ABCD2360

ABCD2361

ABCD2362

ABCD2363

ABCD2364

ABCD2365

ABCD2366

ABCD2367

ABCD2368

ABCD2369

ABCD2370

ABCD2371

ABCD2372

ABCD2373

ABCD2374

ABCD2375

ABCD2376

ABCD2377

ABCD2378

ABCD2379

ABCD2380

ABCD2381

ABCD2382

ABCD2383

ABCD2384

ABCD2385

ABCD2386

ABCD2387

ABCD2388

ABCD2389

84
RETURN
19 WRITE (8,51) ZF,AMX,P55,P55,P25,PS25
ZF = 0.990D0 * ZF
GO TO 18
20 IF (IGOGO .EQ. 1) GO TO 21
AM55 = AMXT
GO TO 13
21 AM25 = AMXT
GO TO 23
22 IF (IGOGO .EQ. 1) GO TO 26
PS55 = P55 / DEXP((PHI55 - PHIS55) / REX55)
IF (IGASMX) 35,41,103
103 WQA = WG24 / A25
C1 = P25 * DSQRT(G / (T25 * AJ)) * CAPSF
MCON = 0
QO(2) = 0.0D0
QO(3) = 0.0D0
AM25 = 0.25D0
TS25 = 0.875D0 * T25
23 DO 24 I = 1,50
CALL PROCOM (FAR24,TS25,CS25,AK25,CP25,REX25,PHIS25,HS25)
V25 = AM25 * CS25
HSCAL = H25 - V25 ** 2 / GAJ2
DELHS = HSCAL - HS25
IF (DABS(DELHS) .LE. 1.5D0) GO TO 25
IF (IGOGO .EQ. 1) GO TO 26
TS25 = TS25 + DELHS / CP25
ICOMIX = 8
GO TO 2
24 TS25 = TS25 + DELHS / CP25
ICOMIX = 8
GO TO 2
25 WQAT = C1 * DSQRT(AK25 / REX25) * AM25 / ((AK25 - 1.0D0) * AM25)
AMX = AM25
IGOOGO = 1
GO TO 16
WG6 = WG24 + W55
ERR(5) = (PS25 - PS55) / PS25
WP55 = FAR55 * W55 / (FAR55 + 1.0D0)
WA55 = W55 / (FAR55 + 1.0D0)
WP24 = FAR24 * W24 / (FAR24 + 1.0D0)
WA24 = W24 / (FAR24 + 1.0D0)
W26 = (WP55 + WP24) / (WA55 + WA24)
H6 = (WG24 * H25 + W55 * HS25) / WG6
CALL THERMO (1.0D0,H6,T6,PAI6,AUX,1,FAR6,1)
C1 = PS55 * A55 * (1.0D0 + AK55 + AM55 ** 2) + PS25 * A25 *
1 (1.0D0 + AK25 + AM25 ** 2)
TS6 = 0.83D0 * T6
DO 32 I = 1,50
CALL PROCOM (FAR6,TS6,CS6,AK6,CP6,REX6,PHIS6,HS6)
C2 = WG6 * DSQRT(AJ * REX6 / T6 / (AK6 * G))
C3 = C2 / (CAPSF * C1)
C4 = (AK6 - 1.0D0) / 2.0D0 - (C3 * AK6) ** 2
C5 = 1.0D0 - 2.0D0 * AK6 * C3 ** 2
C6 = C5 ** 2 + 4.0D0 * C4 * C3 ** 2
ICOMIX = 9
28 IF (C6) 28,29,30 CALL ERROR RETURN
29 AM62G = - C5 / (2.0D0 * C4) GO TO 31
30 AM62G = (DSQRT(C6) - C5) / (2.0D0 * C4)
31 IF (AM62G .LE. 0.0D0) GO TO 28
AM6G = DSQRT(AM62G)
V6 = AM6G * C5
HSCAL = H6 - V6 ** 2 / GAJ2
DELHS = HSCAL - HS6
IF (DABS(DELHS) .LE. .5D0 * TOLALL * HSCAL) GO TO 33
32 TS6 = TS6 + DELHS / CP6
ICOMIX = 10
CALL ERROR
33 A6G = A25 + A55
C7 = DSQRT (1.0D0 + (AK6 - 1.0D0) * AM62G / 2.0D0)
PS6 = C2 / (CAPSF * A6G * AM6G * C7)
P6 = PS6 * DEXP((PHI6 - PHIS6) / REX6)
CALL THERMO (P6,H6,T6,S6,XXI,I,FAF6P0)
S6AVE = (WG24 * S55 + WG55 * S55) / WG6
IF (S6 .GE. S6AVE) GO TO 35
34 TS5 = T6 / (1.0D0 + (((AK6 - 1.0D0) / 2.0D0) * AM6 ** 2))
DO 34 JJ = 1,50
AK6P = AK6
CALL PROCOM (FAR6,TS6,CS6,AK6,CP6,REX6,PHIS6,HS6)
V6 = AM6 * CS6
DELAK6 = AK6P - AK6
IF (DABS(DELAK6) .LE. .5D0 * TOLALL * AK6) GO TO 54
TS6 = T6 / (1.0D0 + (((AK6 - 1.0D0) / 2.0D0) * AM6 ** 2)) ** 1
AK6 = (AK6 / (AK6 - 1.0D0)))
AM6ABD = AM6
RHO = CAPSF * PS6 / (AJ * REX6 * TS6)
A6 = WG6 / (RHO * V6)
WRITE (6,52) A6
GO TO 44
C CALLS A6 AS A FUNCTION OF INPUT AM6
TS6 = T6 / (1.0D0 + (((AK6 - 1.0D0) / 2.0D0) * AM6 ** 2)) ** 1
A6 = AM6ABD = AM6
RHO = CAPSF * PS6 / (AJ * REX6 * TS6)
A6 = WG6 / (RHO * V6)
WRITE (6,52) A6
GO TO 44
C CALLS A6 = F(A6 DESIGN)

86
TS6P = T6 / (1.0D0 + (((AK6 - 1.0D0) / 2.0D0) * AM6ABD ** 2))
DO 39 I = 1,50
CALL PROCOM (PAR6, TS6P, CS6, AK6, CP6, REX6, PHIS6, HS6)
PS6P = PS6 * (TS6P / TS6) ** (AK6 / (AK6 - 1.0D0))
RH06 = CA6SF * PS6P / (AJ * REX6 * TS6P)
V6 = DSQRTGAJ2 * (H6 - HS6)
IP ((H6 - HS6) .LT. 0.0D0) GO TO 42
A6P = WG6 / (RH06 * V6)
DEL6 = A6P - A6
V6 = WG6 / (RH06 * A6)
AM6 = V6 / CS6
AM62 = AM6 ** 2
IP (DABS(DEL6) .LE. .002D0 * A6) GO TO 40
TS6P = T6 / (1.0D0 + (((AK6 - 1.0D0) / 2.0D0) * AM62))
ICOMIX = 12
CALL ERROR
TS6 = TS6P
PS6 = PS6P
GO TO 44
T6 = T55
P6 = P55
H6 = H55
S6 = S55
WG6 = WG55
PS6 = PS55
V6 = V55
AM6 = AM55
IP (IGASMX .EQ. 0) A6 = A55
GO TO 44
WRITE (6,53) H6, HS6
ICOMIX = 13
CALL ERROR
AM62 = AM62G
AM6 = AM6G
A6 = A25 + A55
ICOMIX = 0
CALL COAPBN
RETURN
KKGO = 1
OPRDS = PRFDS * PRCDS
PRFNEW = PRFDS * PS55 / P25 * 1.02D0
PRCNEW = OPRDS / PRFNEW
ICOMIX = 0
CALL ENGVAL
RETURN
FORMAT (22HOSQRT OF H55-HS55 NEG ,6E15.6,6H$$$$$$)
Subroutine COMNOZ

IMPLICIT REAL*8 (A-H, O-Z)
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2), ERR(9)
DIMENSION AWDRD(2)

EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (IAFBN, COM(12))
1 (IMCD, COM(14)), (IMSHOC, COM(16)), (NOZFLT, COM(17)),
2 (ERR(1), COM(24)), (P1, COM(33)), (WG7, COM(334)),
3 (FAR7, COM(335)), (T7, COM(343)), (H7, COM(344)), (S7, COM(345)),
4 (A9, COM(346)), (A8SAV, COM(347)), (TS8, COM(348)),
5 (PS8, COM(349)), (V9, COM(350)), (AM8, COM(351)), (T8, COM(352)),
6 (F8, COM(353)), (H8, COM(354)), (S8, COM(355)), (A9, COM(356)),
7 (A8SAV, COM(357)), (TS9, COM(358)), (PS9, COM(359)),
8 (V9, COM(360)), (AM9, COM(361)), (T9, COM(362)), (P9, COM(363)),
9 (H9, COM(364)), (S9, COM(365)), (P7, COM(369)),
1 (ISPOOL, COM(1044)), (ITRAN, COM(1049))
DATA AWDRD, WORD(4H2L) /

ABCD2557

ABCD2558

ABCD2559

ABCD2560

ABCD2561

ABCD2562

ABCD2563

ABCD2564

ABCD2565

ABCD2566

ABCD2567

ABCD2568

ABCD2569

ABCD2570

ABCD2571

ABCD2572

ABCD2573

ABCD2574

ABCD2575

ABCD2576

ABCD2577

ABCD2578

ABCD2579

ABCD2580

ABCD2581

ABCD2582

ABCD2583

ABCD2584

ABCD2585

ABCD2586

ABCD2587

ABCD2588

ABCD2589

ABCD2590

ABCD2591

ABCD2592

ABCD2593

ABCD2594

ABCD2595

ABCD2596

ABCD2597

ABCD2598

ABCD2599

ABCD2600

ABCD2601
IMSHOC = ICON + 3
ERR(6) = (P7R - P7) / P7R
IF (ISPOOL .EQ. 1) ERR(3) = ERR(6)
IF (IHNOZ .EQ. 1) WRITE (6,5) A8,AM8,A9,AM9
RETURN

C

FORMAT ('14H0NOZZLE DESIGN,10X,3H A8=,E15.8,8H AM8=,E15.8, A9=,E15.8, AM9=,E15.8) EN

Subroutine CONDIV

SUBROUTINE CONDIV (T1,H1,PI,SI,FAR,WG,PA,IDES,AT,AO,PIa,TT8HT,PT,ST,TO,HO,PO,SO,TST,TSO,PSR,PS~, VR, VO, ANT, AMO, ICON)
C ICON=1 SUBSONIC, COMPARE PIR WITH PI
C ICON=2 SONIC, SHOCK INSIDE NOZZLE, COMPARE PIR WITH PI
C ICON=3 SONIC, SHOCK OUTSIDE NOZZLE, COMPARE PIR WITH PI
C ICON=4 ERROR
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL TI
COMMON /COMALL/ COM(1062)
DIMENSION Q(9)
EQUIVALENCE (TOLALL, COM(23)), (ZT, COM(1055))
Q(2) = 0.0D0
Q(3) = 0.0D0
IF (ZI) GO TO 100
AJ = 778.26D0
CAPSF = 2116.2170D0
G = 32.174049D0
GO TO 101
100 AJ = 1.0D0
CAPSF = 101325.0D0
G = 1.0D0
GO TO 101
101 GAJ2 = 2.0 * G * AJ
CALL PROCOR (FAR,T1,XX1,XX2,XX3,XX4,PHII,XX6)
C *** SONIC CALCULATIONS
TSS = 0.833D0 * TI
DO 2 J = 1,50
CALL PROCOR (FAR,TSS,CSS,AK,CP,REXS,PHISS,HSS)
HSCAL = HI - CSS ** 2 / GAJ2
DELHS = HSCAL - HSS
IF (DABS (DELHS) .LE. 5D0 * TOLALL * HSCAL) GO TO 4
2 TSS = TSS + DELHS / CP
3 ICON = 4
RETURN
4 IF (IDES .LE. 0) GO TO 11
C *** SONIC DESIGN, CALCULATE AT
VT = CSS
TST = TSS
PST = PI * (TST / TI) ** (AK / (AK - 1.0D0))
RHO = CAPSF * PST / (AJ * REXS * TST)
AT = WG / (RHO * VT)
AMT = 1.0D0
C *** IDEAL EXPANSION DESIGN, CALCULATE A0
PSO = PA
TSO = TI * (PSO / PI) ** .286D0
DO 7 J = 1,50
    CALL PROCOM (PAR, TSO, CSO, AK, CP, REX, PHI0, HSO)
    PHICAL = PHI0 - REX * DLOG(PI / PSO)
    DELPHI = PHICAL - PHI0
    IF (DABS(DELPHI) .LE. .1D0 * TOLALL * PHICAL) GO TO 8
7 TSO = TSO * DEXP(4.0D0 * DELPHI)
GO TO 3
8 VO = DSQRT(GA12 * (HI - HS0))
    AMO = VO / CSO
    AKP1 = AK + 1.0D0
    AKM1 = AK - 1.0D0
    AO = (AT / AMO) * (2.0D0 * (1.0D0 + AKM1 * AMO ** 2 / 2.0D0) / 1 AKP1) ** (AKP1 / (2.0D0 * AKM1))
    PIR = PI
    ICON = 3
    TO = TI
    HO = HI
    PO = PI
    SO = SI
10 TT = TI
    HT = HI
    PT = PI
    ST = SI
RETURN
C *** ASSUME SONIC THROAT AND ISENTROPIC EXPANSION TO A0
11 VT = CSS
    AMT = 1.0D0
    TST = TSS
    RHO = WG / (AT * VT)
    PST = RHO * AT * REXS * TST / CATSF
    PIR = PST * (TI / TST) ** (AK / (AK - 1.0D0))
    IF (PIR .GE. PA) GO TO 27
    TSO = 0.95D0 * TI
    MAM = 0
13 CALL PROCOM (PAR, TSO, CSO, AK, CP, REX, PHI0, HSO)
    AKP1 = AK + 1.0D0
    AKM1 = AK - 1.0D0
    AMO = DSQRT(2.0D0 * ((TI / TSO) - 1.0D0) / AKM1)
    AOCAL = (AT / AMO) * (2.0D0) * (1.0D0 + AKM1 * AMO ** 2 / 2.0D0) / 1 AKP1) ** (AKP1 / (2.0D0 * AKM1))
    EA = (AO - AOCAL) / AO
    DIR = DSQRT(AO / AOCAL)
    CALL APQUIR (Q1), TSO, EA, 0.0D0, 1.0D0, TOLALL, DIF, TSS0, JCON
    GO TO (14,16,3), JCON
14 TSO = TS0
    IF (TSO - TI) 15,13,16
15 TSC = 2.0D0 * TI / (AK + 1.0D0)
    IF (TSO .GT. TSC) GO TO 17
16 TSO = 0.98D0 * TI
    GO TO 13
17 IF (Q2) .LT. 30.0D0 .OR. AMO .LT. 0.95D0 .OR. MAM .EQ. 1)
GO TO 13
TSO = 2.0D0 * TI / (2.0D0 + 0.98D0 * (AK - 1.0D0))
IAM = 1
GO TO 13

PSO = PIP * (TSO / TI) ** (AK / (AK - 1.0D0))
IF (PSO - PA) = 20, 19, 27

GO TO 13

C *** CRITICAL FLOW, ISENTROPIC EXPANSION TO PA

VO = AMO * CSO
ICON = 1
GO TO 9

C *** SUBSONIC FLOW

PSO = PA
Q(2) = 0.0D0
Q(3) = 0.0D0
TSO = 0.833D0 * TI
DO 22 J = 1, 50
CALL PROCOM (PAR, TSO, CSO, AK, CP, REX, PHI10, HSO)
RHO = CAPSF * PSO / (AJ * REX * TSO)
V0 = WG / (RHO * AO)
HSCAL = HI - VO ** 2 / GAJ2
DLH3 = HSCAL - HSO
IF (DABS(DLH3) .LE. .5D0 * TOLALL * HSCAL) GO TO 23

TSO = TSO + DLH3 / CP
GO TO 3

AMO = VO / CSO
PIR = PSO * (TI / TSO) ** (AK / (AK - 1.0D0))
TST = TSO

CALL PROCOM (PAR, TST, CST, AK, CP, REX, PHIST, HST)
FST = PIR * (TST / TI) ** (AK / (AK - 1.0D0))
RHO = FST * CAPSF / (AJ * REX * TST)
VT = WG / (RHO * AT)
HSCAL = HI - VT ** 2 / GAJ2
EH = (HSCAL - HST) / HSCAL
DIR = 1.0D0 + (HSCAL - HST) / (CP * TST)
CALL AFQUIR (Q(1), TST, EH, 0.0D0, 20.0D0, .5D0*TOLALL, DIR, TSTT, JCON)
GO TO (25, 26, 3), JCON

TST = TSTT
GO TO 24

AMT = VT / CST
ICON = 1
GO TO 9

C *** SUPERCRITICAL FLOW, ISENTROPIC EXPANSION TO PA

PSO = PA
TSO = TI * (PSO / PIP) ** .286D0
DO 29 J = 1, 50
CALL PROCOM (PAR, TSO, CSO, AK, CP, REX, PHI10, HSO)
PHICAL = PHI10 - REX * DLOG(PIR / PSO)
DELPHI = PHICAL - PHI10
IF (DABS(DELPHI) .LE. .1D0 * TOLALL * PHICAL) GO TO 30

TSO = TSO * DEXP(4.0D0 * DELPHI)

GO TO 3

VO = DSQRT(GAJ2 * (HI - HS3))
AMO = VO / CSO
AKP1 = AK + 1.0D0
\[
\text{IF } (A0 - AOID) \leq 31,9,32 \text{, GO TO 35}
\]
\[C *** \text{ SUPERCRITICAL FLOW, ISENTROPIC EXPANSION TO AO}
\]
\[N = 1
\]
\[T50 = 0.633DO \ast TI
\]
\[DO 34 \; J = 1,100
\]
\[\text{CALL PROCOM } (\text{FAR, TSO, CSO, AK, CP, REX, PHISO, HSO})
\]
\[\text{AKP1} = AK + 1.0DO
\]
\[\text{AKM1} = AK - 1.0DO
\]
\[\text{AM0} = \text{DSQRT}(2.0DO \ast ((TI/TSO) - 1.0DO)/AKM1)
\]
\[\text{AOCAL} = (AT/AMO) \ast (2.5DO \ast (1.0DO + AM0 \ast AMO) 2/2.5DO)/2.5DO
\]
\[\text{DELA} = A0 - AOCAL
\]
\[\text{IF (DABS(DELA) \leq 1DO \ast T3L1LL \ast AO) GO TO 35}
\]
\[\text{GO TO 3}
\]
\[C *** \text{ UNDEREXPANDED, SHOCK OUTSIDE NOZZLE}
\]
\[PS0 = PIR \ast (T5O/TI) \ast (AK/(AK - 1.0DO))
\]
\[\text{VO} = AM0 \ast CSO
\]
\[\text{GO TO 9}
\]
\[C *** \text{ OVEREXPANDED, FIND SHOCK POSITION}
\]
\[PSY = PSX \ast (2.0DO \ast AK \ast AM0 \ast 2/2.0DO - (AK + 1.0DO) - (AK - 1.0DO))/AKM1
\]
\[\text{IF (PA \geq PSY) GO TO 39}
\]
\[C *** \text{ OVEREXPANDED, SHOCK OUTSIDE NOZZLE}
\]
\[PS = PSX
\]
\[\text{VO = AM0 \ast CSO}
\]
\[\text{GO TO 9}
\]
\[C *** \text{ OVEREXPANDED, SHOCK INSIDE NOZZLE}
\]
\[PS0 = PA
\]
\[T50 = 0.833DO \ast TI
\]
\[DO 41 \; J = 1,50
\]
\[\text{CALL PROCOM } (\text{FAR, TSO, CSO, AK, CP, REX, PHISO, HSO})
\]
\[\text{RHO = CAPSF \ast PS0/(AJ \ast REX \ast TSO)}
\]
\[\text{VO = WG/(RHO \ast AO)}
\]
\[\text{HSCAL = HI - VO \ast 2/GAJ2}
\]
\[\text{DELHS = HSCAL - HSO}
\]
\[\text{IF (DABS(DELHS) \leq 5DO \ast TOLALL \ast HSCAL) GO TO 42}
\]
\[\text{GO TO 3}
\]
\[\text{GO TO 10}
\]
\[\text{END}
\]
### Subroutine CONOUT

**SUBROUTINE CONOUT (ICON)**

**IMPLICIT REAL*8 (A-H, O-Z)**

**LOGICAL SI**

**COMMON /COMALL/ (COM(1062)**

**DIMENSION WOR2(368,2), IOUT(150), AOUT(6), WOWT(6,2)**

| 1 | WOR1(152), WOR2(152), WOR3(64), WOR12(152), WOR22(152). |
|---|---|---|---|---|---|---|
| 2 | WOR32(64), THEEND(2), BLANK(2), AIN(2), CHANGE(2). |

**EQUIVALENCE (TIME, COM(993)), (ITRAN, COM(1049)), (SI, COM(1055))**

1. (WOR31(1), WOR3(1), WOR1(1), WORDY(1, 2)),
2. (WOR22(1), WOR2(1), WOR3(1), WOR32(1), WOR31(1)),

**DATA WOR1**/ WOR2 |

| 1 | 4HP1, 4HH82, 4HAM23, 4HWHA23, 4HT3, 4H3P23, 4HH23, 4HS23. |
|---|---|---|---|---|---|---|
| 2 | 4HP2, 4HH24, 4HS24, 4HAN25, 4HT25, 4H3P25, 4HH25. |
| 3 | 4HS25, 4HT29, 4H29, 4HA29, 4HAM29, 4HY29, 4HS29. |
| 4 | 4HP28, 4H28, 4HS28, 4H3P28, 4HH28, 4HS28. |
| 5 | 4H529, 4HT29, 4H29, 4HA29, 4HAM29, 4HY29, 4HS29. |
| 6 | 4H3ETD, 4HDPDU, 4HDPDU, 4H3XPT, 4HWPW2, 4H3XPT, 4H5P25. |
| 7 | 4H7XH25, 4H5S25, 4H3XNG5, 4H3XCFAR, 4H3X55, 4H5X55, 4H5S55. |
| 8 | 4H5XF6, 4H3XPF, 4H3X25, 4H25, 4H3X25. |
| 9 | 4H3TX22D, 4H3A25, 4H3V25, 4H3T45U, 4H3T45DS, 4HT5, 4HH5, 4HS5. |
| 10 | 4H3W5, 4H5PAR5, 4H3AX55, 4H3V55, 4H5AS55, 4H3S56, 4H3P56. |
| 11 | 4H3V6, 4H3TP25, 4H3C35, 4H3TH5P, 4H3TH5P, 4H3H5P. |
| 12 | 4H3P3CF, 4H3PFC, 4H3AWAF, 4H3PCFP, 4H3A356, 4H3A356. |
| 13 | 4H3P3EX, 4H3DMD, 4H3P3EF, 4H3C356, 4H3C356. |
| 14 | 4H3PCNF, 4H3PCL, 4H3ZI, 4H3PCNI, 4H3PFPI, 4H3PFPI, 4H3PFPI. |
| 15 | 4H3DHITI, 4H3ZDS, 4H3DCTI, 4H3DLP2, 4H3DLP2, 4H3DLP2. |
| 16 | 4H3HT4, 4H3W3, 4H3HT4, 4H3W3, 4H3HT4. |
| 17 | 4H3HT50, 4H3S50, 4H3HT50, 4H3S50, 4H3HT50. |
| 18 | 4H3P3CBL, 4H3P3CBL, 4H3HTAT, 4H3HTAT, 4H3HTAT. |
| 19 | 4H3ET1, 4H3WAI, 4H3ETAB, 4H3ETAB, 4H3ETAB. |

**DATA WOR2** |

| 1 | 4H3WFD, 4H3FD, 4H3ETAR, 4H3ETAR, 4H3ETAR, 4H3ETAR. |
|---|---|---|---|---|---|
| 2 | 4H3LBC, 4H3LBC, 4H3LBC, 4H3LBC, 4H3LBC. |
| 3 | 4H3XIT, 4H3XIT, 4H3XIT, 4H3XIT, 4H3XIT. |
| 4 | 4H3XBLD, 4H3XBLD, 4H3XBLD, 4H3XBLD, 4H3XBLD. |
| 5 | 4H3T85, 4H3T85, 4H3T85, 4H3T85, 4H3T85. |
| 6 | 4H3T39, 4H3T39, 4H3T39, 4H3T39, 4H3T39. |
| 7 | 4H3P39, 4H3P39, 4H3P39, 4H3P39, 4H3P39. |
| 8 | 4H3MP4, 4H3MP4, 4H3MP4, 4H3MP4, 4H3MP4. |
| 9 | 4H3V4N, 4H3V4N, 4H3V4N, 4H3V4N, 4H3V4N. |
| 10 | 4H3V4F, 4H3V4F, 4H3V4F, 4H3V4F, 4H3V4F. |
| 11 | 4H3P4D, 4H3P4D, 4H3P4D, 4H3P4D, 4H3P4D. |
| 12 | 4H3F4, 4H3F4, 4H3F4, 4H3F4, 4H3F4. |
| 13 | 4H3S21, 4H3S21, 4H3S21, 4H3S21, 4H3S21. |
| 14 | 4H3S55, 4H3S55, 4H3S55, 4H3S55, 4H3S55. |
| 15 | 4H3HT50, 4H3HT50, 4H3HT50, 4H3HT50, 4H3HT50. |
| 16 | 4H3P3C6, 4H3P3C6, 4H3P3C6, 4H3P3C6, 4H3P3C6. |
| 17 | 4H3P3C7, 4H3P3C7, 4H3P3C7, 4H3P3C7, 4H3P3C7. |
| 18 | 4H3B3L, 4H3B3L, 4H3B3L, 4H3B3L, 4H3B3L. |

**ABCD2815**

**ABCD2816**

**ABCD2817**

**ABCD2818**

**ABCD2819**

**ABCD2820**

**ABCD2821**

**ABCD2822**

**ABCD2823**

**ABCD2824**

**ABCD2825**

**ABCD2826**

**ABCD2827**

**ABCD2828**

**ABCD2829**

**ABCD2830**

**ABCD2831**

**ABCD2832**

**ABCD2833**

**ABCD2834**

**ABCD2835**

**ABCD2836**

**ABCD2837**

**ABCD2838**

**ABCD2839**

**ABCD2840**

**ABCD2841**

**ABCD2842**

**ABCD2843**

**ABCD2844**

**ABCD2845**

**ABCD2846**

**93**
DATA WOR31 /
1 4HTS7, 4HPS7, 4HV7, 4HAM7, 4HA7, 4HT8DS, 4HT7, 4HR7, 4HCRD29866
2 4H87, 4H4B8, 4H8AS4, 4HT8, 4H8PS8, 4HV8, 4H8AM8, 4HT8, 4HCRD29853
3 4HPCH, 4H8H8, 4H8S8, 4HA9, 4H8A9S4, 4HT8, 4H89S9, 4HPS9, 4HVR, 4HCRD29869
4 4H4AM9, 4H8T9, 4H8P9, 4H89, 4H89, 4H8AET4, 4H8DPA, 4H8WFA, 4HCRD29870
5 4H8ET, 4H8TAA, 4H8DPA, 4H8XHP, 4H8XNIP, 4H8XNLP, 4H8P22, 4H8U22, 4HCRD29871
6 4H8P21, 4H8U21, 4H8P3, 4H8U3, 4H8P4, 4H8U4, 4H8P50, 4H8U50, 4HCRD29872
7 4H85, 4H8U5, 4H8P55, 4H8U55, 4H8P7, 4H8U7, 4H8P24, 4H8U24, 4HCRD29873
8 4H8P7, 4H8U7, 4H8DUMS, 4H8FP, 4H8XM2, 4H8FTP, 4H8FP, 4H8ISPO / 4HCRD29874

DATA WOR12 /
1 3 * 4H, 4HDS, 14 * 4H, 4HAV, 9 * 4H, 4HAV, 4HCRD29875
2 8 * 4H, 4HDS, 3 * 4H, 4HC, 4HD5, 4H, 4H4, 4HCRD29877
3 4HD24, 4 * 4H, 4H5, 4H55, 12 * 4H, 4HS, 4HCRD29878
4 16 * 4H, 4 * 4HCF, 3 * 4HDS, 4HP, 4HC, 4HF, 4HCRD29879
5 4HD5, 4H5, 4HD58, 4H5S, 4H, 4G5, 4HT, 4H1, 4HCRD29880
6 7 * 4H, 4HF, 2 * 4H, 4HP, 4H, 2 * 4HIP, 4HCRD29881
7 2 * 4H, 4HD5, 4H5G, 6 * 4H, 4HD5, 4H, 9 * 4H, 4HCRD29882
8 4H0, 4HHP, 4HIP, 4HLP, 4HD5, 4HOB, 4H, 4HCRD29883
9 2 * 4HHP, 4H, 4HP, 4HF, 4HCF, 4HF, 4HCF, 4HCRD29884
1 4H, 4HD5, 4H5, 2 * 4HD5, 4HCRD29885

DATA WOR22 /
1 4H5, 11 * 4H, 4HDS, 4H5, 4H, 4GU, 9 * 4H, 4HO, 4HCRD29887
2 17 * 4H, 4HD5, 4H55, 4H, 4H5, 4H5, 4H4, 4H5, 4HCRD29893
3 4HD5, 4H5, 4H5, 4H5, 4H5, 4H55, 4H55, 4H5, 4H5, 4HCRD29898
4 3 * 4HFP, 4HFP, 4 * 4H5, 4H5, 7 * 4H, 6 * 4HD5, 4HCRD29898
5 8 * 4HCF, 4HP, 4H, 2 * 4HLP, 4H, 4HF, 4HCF, 4HCRD29891
6 4H6, 4H5, 4HD5, 4H5, 2 * 4H, 4HC, 4HD5, 4HCRD29882
7 4H5, 4HD5, 16 * 4H, 4H5, 4H, 4H5, 4H5, 4H5, 4HCRD29893
8 4H55, 2 * 4H, 4HD5, 6 * 4H, 4H55, / 4HCRD29898

DATA WOR32 /
1 4H5, 4H5, 4H5, 4H5, 4H5, 4H5, 4H5, 4H5, 4H5, 4HCRD29888

DATA THEEND, BLANK, LIMIT /H THEE, 4HND, 4H, 4H, 368 / 4HCRD29833

IF (ICON .EQ. 1) GO TO 24
IF (SI) GO TO 22
WRITE (6,21)
GO TO 24
WRITE (6,23)

GO TO (1,6), ICON

C *** INPUT SECTION
1 DO 4 N = 1,150
NUM = N
READ (5,11) AIN,CHANGE
IF (AIN(1) .EQ. THEEND(1) .AND. AIN(2) .EQ. THEEND(2)) GO TO 5
DO 2 J = 1, LIMIT
JJ = J
IF (AIN(1) .EQ. WORDY(J,1) .AND. AIN(2) .EQ. WORDY(J,2)) GO TO 3
CONTINUE
WRITE (6,12) AIN
GO TO 4
3 IOUT(NUM) = JJ
IF (CHANGE(1) .EQ. BLANK(1) .AND. CHANGE(2) .EQ. BLANK(2)) 1 GO TO 4
1 GO TO 4
WORDY(JJ,1) = CHANGE(1)
WORDY(JJ,2) = CHANGE(2)

C CONTINUE
WRITE (6,13)
5 NUM = NUM - 1
RETURN
C *** OUTPUT SECTION
6 IF (NUM .EQ. 1) RETURN

C THE FOLLOWING THREE STATEMENTS ARE USED AT LEWIS ONLY
IF (TIME .EQ. 0.00) AND. ITRAN .NE. 1) GO TO 16
WRITE (50,19) TIME, COM(248), COM(257), COM(258)
WRITE (50,20) (COM(I), I = 372,394)

C 15 N = NUM
J = 6
DO 9 I = 1,NUM,6
IF (N.GT.6) GO TO 7
J = N
7 N = N - 6
DO 8 K = 1,J
L = I + K - 1
M = IOUT(L)
WOUT(K,1) = WORDY(M,1)
WOUT(K,2) = WORDY(M,2)
IF (M.LE.362) AOUT(K) = COM(M+32)
IF (M.GT.362 .AND. M.LE.367) AOUT(K) = COM(M+694)
IF (M.GT.367) AOUT(K) = COM(1044)
8 CONTINUE
WRITE (6,14) (WOUT(K,1), WOUT(K,2), K = 1,J)
WRITE (6,15) (AOUT(K),K=1,J)
IF (N.LE.0) RETURN
9 CONTINUE
RETURN

C 11 FORMAT (A4, A2, 6X, A4, A2)
12 FORMAT (1HO, 9H THE WORD , A4, A2, 6H NOT FOUND IN COMMON ARRAY)
13 FORMAT (22H ERROR IN CONOUT INPUT)
14 FORMAT (1HO, 25X, A4, A2, 5(9X, A4, A2))
15 FORMAT (21X, 6E15.6)

C THE FOLLOWING TWO FORMATS ARE USED AT LEWIS ONLY
16 FORMAT (1X, 1PE20.6)
17 FORMAT (1X, 1PE20.6)
18 C FORMAT (1X, 30H THE OUTPUT IS IN ENGLISH UNITS)
19 FORMAT (1X, 25H THE OUTPUT IS IN SI UNITS)
END
Subroutine CONVRG

SUBROUTINE CONVRG (TI, HI, PI, SI, FAR, WG, PA, IDES, AO, PR, TO, HO, PO, SO)

1 TSO, PSO, VO, AMO, ICON)
C ICON=1  SUBSONIC, COMPARE PI WITH PR
C ICON=2  SONIC, COMPARE PI WITH PR
C ICON=4  ERROR
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL ZI
COMMON /COPIALL/ CON (1062)
EQUIVALENCE (TOLALL, CON (23)), (ZI, CON (1055))
IF (ZI) GO TO 100
AJ = 778.2600
CAPSF = 2116.217DO
G = 32.174049DO
GO TO 101
AJ = 1.0D0
CAPSF = 1.0D0
G = 1.0D0
CPG = 1048.0D0
GO TO 101
100 AJ = 1.0D0
CAPSF = 1.0D0
G = 1.0D0
CPG = 1048.0D0
101 GAI2 = 2.0D0 * G * AJ
CALL PROCOM (PAR, TI, XX1, XX2, XX3, XX4, PHII, XX6)
C *** SONIC CALCULATIONS
TSS = 0.033D0 * TI
DO 2 J = 1, 50
CALL PROCOM (PAR, TSS, CSS, AKS, CP, REXS, PHISS, HSS)
HSCAL = HI - CSS ** 2 / GAI2
DELHS = HSCAL - HSS
IF (DABS (DELHS) .LE. .5D0 * TOLALL * HSCAL) GO TO 4
2 TSS = TSS + DELHS / CP
3 ICON = 4
RETURN
4 IF (IDES .LE. 0) GO TO 12
C *** ISENTROPIC EXPANSION CALCULATIONS
TSI = TI * (PA / PI) ** 0.286D0
DO 7 J = 1, 70
CALL THERMO (PA, HSI, TSI, SSI, XX1, 1, PAR, 0)
IF (DABS (SSI - SI) .LE. .1D0 * TOLALL * SI) GO TO 8
7 TSI = TSI / DEXP ((SSI - SI) / CPG)
GO TO 3
8 VIS = DSQRT (GAI2 * (HI - HSI))
IF (VIS .GE. CSS) GO TO 11
C *** SUBSONIC DESIGN, CALCULATE AO
VO = VIS
TSO = TSI
PSO = PA
CALL PROCOM (PAR, TSO, CSO, XX2, XX3, REX, PHISO, HSO)
RHO = CAPSF * PSO / (AJ * PEX * TSO)
AO = WG / (RHO * VO)
AMO = VO / CSO
PR = PI
ICON = 1
RETURN
10 TO = TI
HO = HI
PO = PI
SO = SI
RETURN
END
RETURN

C *** SONIC DESIGN, CALCULATE AO
11 VO = CSS
TSO = TSS
PSO = PI * (TSO / TI) ** (AKS / (AKS - 1.0D0))
RHO = CAPSF * PSO / (AJ * REXS * TSO)
AC = WG / (RHO * VO)
AMO = 1.0D0
PR = PI
ICON = 2
GO TO 10

C *** NON-DESIGN, CALCULATE CRITICAL CONDITIONS
12 VO = CSS
TSO = TSS
PSO = PA
RHO = CAPSF * PSO / (AJ * REXS * TSO)
AOCRIT = WG / (RHO * VO)
AMO = 1.0D0
PR = PSO * (TI / TSO) ** (AKS / (AKS - 1.0D0))
IF (AO .GT. AOCRIT) GO TO 14

C *** NON-DESIGN, CRITICAL AND SUPERCritical CONDITIONS
PSO = PSO * AOCRIT / AO
PR = PR * AOCRIT / AO
ICON = 2
GO TO 10

C *** NON-DESIGN, SUBSONIC CALCULATIONS
14 PSO = PA
TSO = 0.833D0 * TSO
DO 16 J = 1, 50
CALL PROCOC (FAR, TSO, CSO, AKO, CP, REX, PHISO, HSO)
RHO = CAPSF * PSO / (AJ * REXS * TSO)
VO = WG / (RHO * AO)
HSCAL = HI - VO ** 2 / GAJ2
DELHS = HSCAL - HSO
IF (DABS (DELHS) .LE. .5D0 * TOLALL * HSCAL) GO TO 17
15 TSO = TSO + DELHS / CP
GO TO 3
17 AMO = VO / CSO
PR = PSO * (TI / TSO) ** (AKO / (AKO - 1.0D0))
ICON = 1
GO TO 10
END

FUNCTION DERIV (I, X)
IMPLICIT REAL*8 (A-H, O-Z)
COMMON / COMALL/ COM (1062)
DIMENSION PO (50, 4), PVREDIT (23), XS (23)
EQUIVALENCE (PO(1, 1), COM(430)), (DT, COM(994)),
1 (PVREDIT (1), COM(998)), (XS (1), COM (1021)), (JTRAN, COM (1050)),
2 (IVAVDOT, COM (1052)), (IARIX, COM(1054))
IF (IARIX .EQ. 1) GO TO 2
IF (JTRAN .EQ. 1) GO TO 1

97


```fortran
DERIV = 0.0DO
FO(I,1) = X
FO(I,2) = X
FO(I,3) = DERIV
FO(I,4) = DERIV
RETURN

1 X0 = FO(I,2)
DERIV = (X - X0) / DT
IF (DABS(DERIV) .LT. 1.0D-10) DERIV = 0.0DO
FO(I,1) = X
FO(I,3) = DERIV
RETURN

2 IF (I .LT. IVRDOT) GO TO 3

3 DERIV = XS(I) * PV RDOT(I)
RETURN
END

Subroutine DISTRB

SUBROUTINE DISTRB
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CORBLL/ COM (1062)
DOUBLE PRECISION LWV, MWV
DIMENSION XVAF (23), PVRD3T (23), XS (23)
DIMENSION XSAVE (1062), AINV (529), ARXNV (52a), AX (23,23)
DIMENSION CINT(23,23), X(23,23), A12INV(23,23), A22INV (23,23)
DIMENSION CR(50,23), UVAR(50,23), AIBM(23,5), UVAR(1). COM(372)
DIMENSION AIBM(23,5), B(23,5), BR(23,5), Y(50,5)
DIMENSION D(50,5), DR(50,5), XSS(23), YI(23), USS(50), DU(5), BS(5)
DIMENSION PVRDOT (1), CON(998), X5 (I), CON (1021)
DIMENSION INVTHE TOTAL NUMBER OF STATES POSSIBLE FOR THE PARTICULAR ENGINE BEING MODELED, LESS THAN OR EQUAL TO 23
DATA INV / 16 / DATA INVRED / 9 /
DIVARBE THE ORDER IN WHICH THE STATES ARE TO BE DONE. IF THIS IS A REDUCED MODEL, THE STATES TO BE INCLUDED ARE LISTED BEFORE THOSE NOT INCLUDED.
```
DATA IVARB / 1, 3, 21, 19, 19, 9, 8, 11, 20, 5, 13, 17, 12, 10, 14, 16, 7 * 0 /
INB THE NUMBER OF CONTROL INPUTS, LESS THAN OR EQUAL TO 5
DATA INB / 2 /
BPER A PERCENT. THE STEADY STATE VALUE OF EACH INPUT WILL BE MULTIPLIED BY THIS PERCENT TO FORM THE DELTA U OF EACH INPUT IN ORDER TO CALCULATE THE B AND D MATRICES
DATA BPER / .01D0 /
INC THE NUMBER OF OUTPUTS, LESS THAN OR EQUAL TO 50
DATA INC / 3 /
NUCOM A LIST OF THE EQUIVALENCED "COM" NUMBERS OF THE OUTPUTS DESIRED. ALL OF THE EQUIVALENCES WILL BE FOUND IN THE MAIN ROUTINE, DYGABCD
DATA NUCOM / 249, 257, 258, 47 * 0 /

IF THE A AND C MATRIX CALCS ARE FINISHED, GO ON TO THE B AND D MATRIX CALCS
IF (IB .EQ. 977) GO TO 4
IF THIS IS NOT THE FIRST TIME THROUGH, GO TO 7
IF (IDOT .GT. 0) GO TO 7
IGIN = 0
IBCK = 0
ABCK = 1.D-5
IF THE PROGRAM IS TO CALCULATE THE PVRDOT'S, ICHOIC=1
IF THE PROGRAM IS TO SET THE PVRDOT'S, ICHOIC=0
ICHOIC = 1
PVRDOT A VECTOR OF (INITIAL CONDITION) PERCENTS. THE STEADY STATE VALUE OF EACH STATE WILL BE MULTIPLIED BY THE PERCENT ASSOCIATED WITH THAT STATE TO FORM THE DELTA XDOT FOR THAT PARTICULAR STATE IN ORDER TO CALCULATE THE A MATRIX
CORE ROTOR SPEED
PVRDOT(1) = .02D0
INTERMEDIATE ROTOR SPEED
PVRDOT(2) = 0.0D0
FAN ROTOR SPEED
PVRDOT(3) = .02D0
FAN EXIT PRESSURE
PVRDOT(4) = .02D0
FAN EXIT INTERNAL ENERGY
PVRDOT(5) = .02D0
INTERMEDIATE COMPRESSOR EXIT PRESSURE
PVRDOT(6) = 0.0D0
INTERMEDIATE COMPRESSOR EXIT INTERNAL ENERGY
PVRDOT(7) = 0.0D0
COMPRESSOR EXIT PRESSURE
PVRDOT(8) = .02D0
COMPRESSOR EXIT INTERNAL ENERGY
PVRDOT(9) = .02D0
COMBUSTOR EXIT PRESSURE
PVRDOT(10) = .02D0
COMBUSTOR EXIT INTERNAL ENERGY
PVRDOT(11) = .02D0
HIGH PRESSURE TURBINE EXIT PRESSURE
PVRDOT(12) = .02D0
HIGH PRESSURE TURBINE EXIT INTERNAL ENERGY
PVRDOT(13) = .02D0
INTERMEDIATE PRESSURE TURBINE EXIT PRESSURE
PVRDOT(14) = 0.0D0
INTERMEDIATE PRESSURE TURBINE EXIT INTERNAL ENERGY
PVRDOT(15) = 0.0D0
LOW PRESSURE TURbine EXIT PRESSURE
PVRDOT(16) = .02DO

LOW PRESSURE TURBINE EXIT INTERNAL ENERGY
PVRDOT(17) = .02DO

AFTERBURNER EXIT PRESSURE
PVRDOT(18) = .02DO

AFTERBURNER EXIT INTERNAL ENERGY
PVRDOT(19) = .02DO

DUCT BURNER EXIT PRESSURE
PVRDOT(20) = .02DO

DUCT BURNER EXIT INTERNAL ENERGY
PVRDOT(21) = .02DO

THIRD STREAM EXIT PRESSURE
PVRDOT(22) = 0.0DO

THIRD STREAM EXIT INTERNAL ENERGY
PVRDOT(23) = 0.0DO

CONTINUE
IDOT = IDOT + 1
IF (IDOT .GT. INV) GO TO 10
IVBDOT = IVARB(IDOT)
IF (IDOT .NE. 1) GO TO 10

SAVE STEADY STATE VALUES AND INITIALIZE FOR A AND C MATRIX CALCS

DO 5 I = 1, INV
J = IVARB(I)
XS(J) = XVAR(J)
XSS(I) = XVAR(J)
IF (XSS(I) .EQ. 0.0DO) WRITE (6,50)
CONTINUE
5
DO 96 I = 1, INC
IU = NUCOM(I)
USS(I) = COM(IU)
DO 1 I = 1, 1062
1 XSAVE(I) = COM(IU)
WRITE (6,65)
WRITE (6,44) (XSS(I), I = 1, INV)
10 RETURN
11 IDOT1 = IDOT - 1
12 C
13 GENERATE MATRIX FOR A MATRIX CALCS
14 C
15 DO 15 I = 1, INV
16 J = IVARB(I)
17 X(I, IDOT1) = XVAR(J) - XSS(I)
18 J = IVARB(IDOT1)
19 IF (ICHOIC .EQ. 0) GO TO 23
20 ERRX = DABS(X(IDOT1, IDOT1) / XSS(IDOT1)) - .002
21 IF (DABS(ERRX) .LE. .000010D) GO TO 17
22 IDOT = IDOT1
23 IVRDOT = IVARB(IDOT)
24 IGIN = IGIN + 1
25 IF (DABS(X(1, IDOT1)) .NE. 0.0D0) GO TO 18
26 PVRP = PVRDOT(J)
27 PVREDOT(J) = 2.0D0 * PVREDOT(J)
28 IF (ERRX .EQ. ERRXP .OR. IBCK .NE. 0) GO TO 9
29 PVREDOT(J) = 1.5D0 * PVRP * ABCK
30 ABCK = 2.0D0 * ABCK
31 ERRXP = ERRX
32 IBCK = 1
33 GO TO 23
34 9 IF (IGIN .NE. 1) GO TO 16
35 IBCK = 0
36 PVRP = PVREDOT(J)
37 ERRXP = ERRX
38 PVREDOT(J) = 1.05D0 * PVREDOT(J)
39 GO TO 23
40 16 PVRED = (PVREDOT(J) * ERRXP - PVRP * ERRX) / (ERRXP - ERRX)
41 IBCK = 0
42 IF (PVRED .LE. 0.0D0) PVRED = .95D0 * PVREDOT(J)
43 IF (IGIN .EQ. 50) PVRED = ((PVREDOT(J) + PVRP) / 2.0D0) + .1D-3
44 IF (IGIN .EQ. 100) PVRED = ((PVREDOT(J) + PVRP) / 2.0D0) + .1D-2
45 PVRED = PVREDOT(J)
46 ERRXP = ERRX
47 PVREDOT(J) = PVRED
48 GO TO 23
49 C
50 GENERATE MATRIX FOR C MATRIX CALCS
51 C
52 IGIN = 0
53 IBCK = 0
54 ABCK = 1.0D-5
55 IF (PVREDOT(J) .LE. 0.0D0) PVREDOT(J) = ABCK
56 GO TO 97
57 23 DO 97 I = 1, INC
58 IU = NUCOM(I)
59 97 UVAR(I, IDOT1) = COM(IU) - USS(I)
60 WRITE (6, 45) IDOT1, IVARB(IDOT1)
61 WRITE (6, 60)
62 WRITE (6, 44) (X(I, IDOT1), I = 1, INV)
63 WRITE (6, 282) IGIN
64 WRITE (6, 296) PVREDOT(J)
C IAMTSV = IAMTRX
C IVRSV = IVRDOT
C IDOTSV = IDOT
C DO 19 I = 1,23
19
C PYRDOS(I) = PYRDOT(I)
C C RESET STEADY STATE VALUES
C DO 21 I = 1,1062
C COM(I) = XSAVE(I)
C IAMTRX = IAMTSV
C IVRDOT = IVRSV
C IDOT = IDOTSV
C DO 22 I = 1,23
C IF (IGIN .NE. 0) RETURN
C IF (IDOT .LE. INV) RETURN
C C FINAL CALCS FOR A AND C MATRICES
C DO 24 I = 1,INV
C J = IVARB(I)
24
C YI(I) = 1.0D0 / XSS(I) / PYRDOT(J)
C WRITE (6,55)
C WRITE (6,44) (YI(I), I = 1,INV)
C WRITE (6,75)
C DO 31 J = 1,INV
C DO 30 I = 1,INV
C K = INV * (J-1) + I
C LL = K - INV + 1
C WRITE (6,44) (AINV(I), I = LL,K)
C CONTINUE
C DO 46 J = 1,INVRED
C DO 46 I = 1,INVRED
C K = INVRED * (J - 1) + I
C DO 47 J = 1,INVRED
C DO 48 I = 1,INVRED
C A12INV(J,I) = YI(J) * X(J,I)
C CONTINUE
C DO 98 J = 1,INV
C DO 98 I = 1,INC
C A21INV(I,J) = YI(J) * UVAR(I,J)
CALL DMINV (AINV, INV, DET, LWV, MWV)
IF (DET .EQ. 0.0D0) WRITE (6,70)
DO 32 J = 1,INV
DO 32 I = 1,INV
K = INV * (J-1) + I
A(I,J) = AINV(K)
WRITE (6,80)
C
THE FOLLOWING STATEMENT IS USED AT LEWIS ONLY
WRITE (10) A
DO 100 K = 1,INV
DO 100 I = 1,INC
SUM = 0.0D0
DO 99 J = 1,INV
SUM = SUM + CA(I,J) * A(J,K)
99
C(I,K) = SUM
C
THE FOLLOWING STATEMENT IS USED AT LEWIS ONLY
WRITE (20) C
C
DO 33 J = 1,INV
WRITE (6,44) (A(J,I), I = 1,INV)
33
CONTINUE
WRITE (6,81)
DO 101 J = 1,INV
WRITE (6,44) (C(J,I), I = 1,INV)
101
CONTINUE
C
RESET STEADY STATE VALUES AND INITIALIZE
FOR B AND D MATRIX CALCS
C
IB = 977
JDOT = 0
DO 26 I = 1,1062
COM(I) = XSAVE(I)
DO 76 I = 1,INC
IU = NUCOM(I)
76
USS(I) = COM(IU)
DO 34 I = 1,INV
34
PVRDOT(I) = 0.0D0
BS(1) = WFB
BS(2) = A8
BS(3) = X3
BS(4) = X4
BS(5) = X5
DO 27 I = 1,5
27
DU(I) = BPER * BS(I)
4
IF (JDOT .EQ. 1) GO TO 35
DO 2 I = 1,INV
J = IVARB(I)
2
XSS(I) = XVAR(J)
DO 6 I = 1,1062
6
XSAVE(I) = COM(I)
104
IAMTRX = 1
ITRAN = 0
IDOT = 0
IDES = 0
MODE = 2
JDOT = 1
WFB = WFB + DU(1)
RETURN
IDOT = IDOT + 1

C GENERATE MATRIX FOR B MATRIX CALCS
C
DO 36 I = 1, INV
    J = IVARB(I)
    AIBM(I,IDOT) = (XVAR(J) - XSS(I)) / DU(IDOT)
C
C GENERATE MATRIX FOR D MATRIX CALCS
C
DO 77 I = 1, INC
    IU = NUCOM(I)
    Y(I,IDOT) = (COV(IU) - USS(I)) / DU(IDOT)
C
C RESET STEADY STATE VALUES
C
IAMTSV = IAMTRX
MODESV = MODE
IDESV =IDES
ITRASV = ITRAN
IDOTS V = IDOT
DO 29 I = 1, 1062
    COM(I) = XSAVE(I)
    IAMTRX = IAMTSV
    MODE = MODESV
    IDESV = IDESV
    ITRAN = ITRASV
    IDOT = IDOTS V
    IF (IDOT .GE. INB) GO TO 28
    GO TO (11, 12, 13, 14), IDOT

11 A8 = A8 + DU(2)
RETURN
12 X3 = X3 + DU(3)
RETURN
13 X4 = X4 + DU(4)
RETURN
14 X5 = X5 + DU(5)
RETURN

C
C FINAL CALCS FOR B MATRIX
C
DO 38 K = 1, INB
DO 38 I = 1, INV
SUM = 0.00D0
DO 37 J = 1, INV
SUM = SUM - (A(I,J) * AIBM(J,K))
E(I,K) = SUM
THE FOLLOWING STATEMENT IS USED AT LEWIS ONLY
WRITE (15) B

FINAL CALCS FOR D MATRIX
DO 87 K = 1, N
DO 97 I = 1, INV
SUM = 0.0
DO 86 J = 1, INV
SUM = SUM + C(I, J) * AIEM(J, K)
87 D(I, K) = Y(I, K) - SUM
THE FOLLOWING STATEMENT IS USED AT LEWIS ONLY
WRITE (25) D

WRITE (6, 85)
GO TO (39, 40, 41, 42, 43), INB
39 WRITE (6, 91) ((B(I, J), J = 1, INB), I = 1, INV)
GO TO 110
40 WRITE (6, 92) ((B(I, J), J = 1, INB), I = 1, INV)
GO TO 110
41 WRITE (6, 93) ((B(I, J), J = 1, INP), I = 1, INV)
GO TO 110
42 WRITE (6, 94) ((B(I, J), J = 1, IN3), I = 1, INV)
GO TO 110
43 WRITE (6, 95) ((B(I, J), J = 1, INW), I = 1, INV)
110 WRITE (6, 82)
GO TO (111, 112, 113, 114, 115), INB
111 WRITE (6, 97) ((D(I, J), J = 1, INB), I = 1, INC)
GO TO 200
112 WRITE (6, 98) ((D(I, J), J = 1, INB), I = 1, INC)
GO TO 200
113 WRITE (6, 99) ((D(I, J), J = 1, INB), I = 1, INC)
GO TO 200
114 WRITE (6, 100) ((D(I, J), J = 1, INR), I = 1, INC)
GO TO 200
115 WRITE (6, 101) ((D(I, J), J = 1, INB), I = 1, INC)
200 IF (L .GT. INV) STOP

CALCULATIONS FOR ALL MATRICES OF REDUCED ORDER MODEL

FINAL CALCS FOR REDUCED A MATRIX
CALL DMINV (ARINV, INVRED, DET, LW, MW)
DO 201 J = 1, INVRED
DO 201 I = 1, INVRED
K = INVRED * (J - 1) + I
201 AR(I, J) = ARINV(K)

FINAL CALCS FOR REDUCED B MATRIX
DO 203 K = 1, INB
DO 203 I = 1, INVRED
SUM = 0.0D0
DO 202 J = 1, INVRED
202 SUM = SUM - AR(I,J) * AIBM(J,K)
203 BR(I,K) = SUM
C
C FINAL CALCS FOR REDUCED C MATRIX
C
DO 205 K = 1, INVRED
DO 205 I = L, INV
SUM = 0.0D0
DO 204 J = 1, INVRED
204 SUM = SUM + A21INV(I,J) * AR(J,K)
205 CINT(I,K) = SUM
DO 207 K = 1, INVRED
DO 207 I = 1, INC
SUM = 0.0D0
DO 206 J = L, INV
206 SUM = SUM + C(I,J) * CINT(J,K)
207 CR(I,K) = C(I,K) + SUM
C
C FINAL CALCS FOR REDUCED D MATRIX
C
DO 209 K = L, INV
DO 209 I = 1, INVRED
SUM = 0.0D0
DO 208 J = 1, INVRED
208 SUM = SUM + AR(I,J) * A12INV(J,K)
209 CINT(I,K) = SUM
DO 211 K = L, INV
DO 211 I = L, INV
SUM = 0.0D0
DO 210 J = 1, INVRED
210 SUM = SUM + A21INV(I,J) * CINT(J,K)
211 DINT(I,K) = SUM - A22INV(I,K)
DO 213 K = L, INV
DO 213 I = 1, INC
SUM = 0.0D0
DO 212 J = L, INV
212 SUM = SUM + C(I,J) * DINT(J,K)
213 BINT(I,K) = SUM
DO 103 K = 1, INB
DO 103 I = 1, INC
SUM = 0.0D0
DO 102 J = L, INV
102 SUM = SUM + BINT(I,J) * B(J,K)
103 DR(I,K) = SUM + D(I,K)
WRITE (6,214)
DO 218 I = 1, INVRED
WRITE (6,44) (BR(I,J), J = 1, INVRED)
218 CONTINUE
GO TO (219, 220, 221, 222, 223), INB
WRITE (6, 91)  ((BR(I,J), J = 1, INB), I = 1, INVRED)
GO TO 224
WRITE (6, 92)  ((BR(I,J), J = 1, INB), I = 1, INVFED)
GO TO 224
WRITE (6, 93)  ((BR(I,J), J = 1, INB), I = 1, INVRED)
GO TO 224
WRITE (6, 94)  ((BR(I,J), J = 1, INB), I = 1, INVFED)
WRITE (6, 95)  ((BR(I,J), J = 1, INB), I = 1, INVRED)
WRITE (6, 216)
DO 225 J = 1, INC
WRITE (6, 44)  (CF(J,I), I = 1, INVRED)
CONTINUE
WRITE (6, 217)
GO TO (226, 227, 228, 229, 230), INB
WRITE (6, 91)  ((DR(I,J), J = 1, INB), I = 1, INC)
STOP
WRITE (6, 92)  ((DR(I,J), J = 1, INB), I = 1, INC)
STOP
WRITE (6, 93)  ((DR(I,J), J = 1, INB), I = 1, INC)
STOP
WRITE (6, 94)  ((DR(I,J), J = 1, INB), I = 1, INC)
STOP
WRITE (6, 95)  ((DR(I,J), J = 1, INB), I = 1, INC)
STOP
FORMAT (1P10E12.4)
FORMAT (1X, I5, 2X, 'THE VARIABLE IS NO. ', I3)
FORMAT (1X, ' THE S.S. VARIABLE IS 0.0 ')
FORMAT (1X, ' Y INVERSE = ')
FORMAT (1X, ' THIS COLUMN OF X = ')
FORMAT (1X, ' THIS IS THE S.S. SOLUTION ')
FORMAT (1X, ' AINV IS SINGULAR ')
FORMAT (1X, 7HAINV = )
FORMAT (1X, 4HA = )
FORMAT (1X, 4HC = )
FORMAT (1X, 4HD = )
FORMAT (1X, 4HB = )
FORMAT (1X, 1PE12.4)
FORMAT (1X, 1P2E12.4)
FORMAT (1X, 1P3E12.4)
FORMAT (1X, 1P4E12.4)
FORMAT (1X, 1P5E12.4)
FORMAT (1X, 5HAR = )
FORMAT (1X, 5HBR = )
FORMAT (1X, 5HCR = )
FORMAT (1X, 5HDR = )
FORMAT (1H1)
FORMAT (1X, 'IGN = ', I5)
FORMAT (1X, 'THE NEXT TRY FOR PVRDOT = ', 1PE12.4)
END
Subroutine ENGBAL

IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL EERR, DUMSPL, FXPN2M, FXM2CP, PAN
COMMON /COMALL/ COM (1062)
DIMENSION WORD(2), ERF (9)
DIMENSION VAR (9), DEL(9), ERRB (9), DELVAR(9), EMAT (9, 9), VMAT (9, 9)
1 AMAT (9), DELSAV (9), AWORD (2)
EQUIVALENCE (WORD(1), COM(1)), (MODE, COM(6)), (INIT, COM(7)),
1 (ITRYS, COM(18)), (LOOPER, COM(19)), (NOMAP, COM(20)),
2 (NUMAP, COM(21)), (MAPEDG, COM(22)), (TOLALL, COM(23)),
3 (ERR(1), COM(24)), (TFHPDS, COM(118)), (ZF, COM(136)),
4 (PCNF, COM(137)), (ZI, COM(139)), (PCNI, COM(140)),
5 (TFFIP, COM(141)), (T4, COM(155)), (TFPHP, COM(175)),
6 (TFLPDS, COM(275)), (TFPDS, COM(278)), (TFPLP, COM(299)),
7 (ZC, COM(300)), (PCNC, COM(301)), (TIME, COM(993)),
8 (DT, COM(994)), (TF, COM(995)), (ISPOOL, COM(1044)),
9 (ITRAN, COM(1049)), (JTRAN, COM(1050)), (NSTEP, COM(1051)),
1 (IAMTRX, COM(1054)), (ERRER, COM(1056)), (DUMSPL, COM(1057)),
2 (FXPN2M, COM(1058)), (FXM2CP, COM(1059)), (PAN, COM(1061))
DATA AWORD /IIHENGB, @HAL/
DATA VDELTA, VLIM, VCHNGE, NORISX /1.0D-4, 0.1D0, 0.85D0, 4/
DATA DEL /g*0.OD0/
DATA DELSAV /9*1.OD-4/

IF (ITRAN .NE. 1) GO TO 100
CALL SYG(1)
JTRAN = 1
INIT = 1
NSTEP = NSTEP + 1
IF (IAMTRX .EQ. 1) NSTFP = NSTEP - 1
TIME = DT * DFLOAT(NSTEP)
IF (TIME .GT. TF) GO TO 100
CALL DISTR
CALL CINLT
GO TO 101
IF (IAMTRX .EQ. 1) CALL DISTR
IF (IAMTRX .EQ. 1) CALL CINLT
IF (IAMTRX .EQ. 1) CALL PUTIN
GO TO 101
IF (INIT .EQ. 1) GO TO 1
TFHP = TFHPDS
TFIP = TFPDS
IF (FXM2CP) TFP = TFHPDS
TFLP = TFPDP
LOOPER = 0
NUMAP = 0
NOMISS = 0
LOOP = 0
NISMAT = 0
NOMAP = 0
IGO = 2
DO 3 I = 1, 9
VMAT(I) = 0.0D0
AMAT(I) = 0.0D0
DELVAR(I) = 0.0D0
DO 3 L = 1, 9
EMAT(I,L) = 0.0D0
LOOPER = LOOPER + 1
CALL COFAN
WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
IF (.NOT. FAN) DUMSPL = .TRUE.
IF (LOOPER .LE. ITRY) GO TO 45
ERROR = .TRUE.
GO TO 26
IF (NUMAP .GT. 0) GO TO 2
NUMAP = 0
VAR(1) = ZF * 100.0D0
VAR(2) = PCNF
IF (MODE .EQ. 3) VAR(2) = T4 / 10.0D0
VAR(3) = ZC * 100.0D0
VAR(4) = PCNC
IF (MODE .EQ. 1) VAR(4) = T4 / 10.0D0
VAR(5) = TFFHP
VAR(6) = TFFLP
VAR(7) = ZI * 100.0D0
VAR(8) = PCNI
VAR(9) = TFFIP
NMAX = 9
IF (FAN) GO TO 39
NMAX = 6
IF (ISPOOL .EQ. 2) GO TO 7
NMAX = 3
VAR(3) = TFFLP
GO TO 7
IF (.NOT. FXFN2M .AND. (.NOT. DUMSPL)) GO TO 6
NMAX = 7
IF (DUMSPL) NMAX = 6
IF (.NOT. FXM2CP) GO TO 7
NMAX = 7
VAR(4) = PCNI
VAR(5) = TFFIP
DO 8 I = 1,NMAX
IF (DABS (EFEI(I)) .GT. TOLALL) GO TO 9
CONTINUE
IF (ITRAN .EQ. 1) CALL FOLL
CALL PERF
CALL ERROR
IF (LOOP .GT. 0) GO TO 11
MAPEDG = 0
MAPSET = 0
DO 10 I = 1,NMAX
ERRB(I) = ERR(I)
10 DEL(I) = VDELT(A * VAR(I))
GO TO 14
IF (MISMAT .GT. 0) GO TO 29
IF (MAPEDG .EQ. 0) GO TO 12
MAPEDG = 0
MAPSET = 1
VAR(LOOP) = VAR(LOOP) + 2.0D0 * DEL(LOOP)
GO TO 15

12 IF (MAPSET .EQ. 0) VAR(LOOP) = VAR(LOOP) + DEL(LOOP)
IF (MAPSET .EQ. 1) VAR(LOOP) = VAR(LOOP) - DEL(LOOP)
MAPSET = 0
DO 13 I = 1,NMAX
IF (DEL(LOOP) .NE. 0.0D0) DELSAV(LOOP) = DEL(LOOP)
IF (DEL(LOOP) .EQ. 0.0D0) DEL(LOOP) = DELSAV(LOOP)
13 EMAT(I,LOOP) = (ERRB(I) - ERR(I)) / DEL(LOOP)
14 LOOP = LOOP + 1
IF (LOOP .GT. NMAX) GO TO 17
VAR(LOOP) = VAR(LOOP) - DEL(LOOP)
15 ZF = VAR(1) / 100.0D0
IF (MODE .NE. 3) PCNF = VAR(2)
IF (MODE .EQ. 3) T4 = VAR(2) * 10.0D0
ZC = VAR(3) / 100.0D0
IF (MODE .NE. 1) PCNC = VAR(4)
IF (MODE .EQ. 1) T4 = VAR(4) * 10.0D0
TFFHP = VAR(5)
TFFLP = VAR(6)
ZI = VAR(7) / 100.0D0
PCNI = VAR(8)
TFPIP = VAR(9)
IF (.NOT. PXM2CP) GO TO 16
16 IF (ISPOOL .EQ. 1) TFFLP = VAR(3)
IF (ZI .LT. 0.0D0) ZI = 0.05D0
IF (ZF .LT. 0.0D0) ZF = 0.05D0
IF (ZC .LT. 0.0D0) ZC = 0.05D0
GO TO (2,4), IGO
17 DO 18 I = 1,NMAX
18 AMAT(I) = - ERBB(I)
DO 20 I = 1,NMAX
IZERO = 0
DO 19 LOOP = 1,NMAX
19 IF (EMAT(I,LOOP) .EQ. 0.0D0) IZERO = IZERO + 1
CONTINUE
IF (IZERO .LT. NMAX) GO TO 20
WRITE (6,32) I
LOOPER = ITRYS + 100
GO TO 26
20 CONTINUE
DO 22 LOOP = 1,NMAX
IZERO = 0
DO 21 I = 1,NMAX
21 IF (EMAT(I,LOOP) .EQ. 0.0D0) IZERO = IZERO + 1
IF (IZERO .LT. NMAX) GO TO 22
WRITE (6,33) LOOP
LOOPER = ITRYS + 100
GO TO 26
22 CONTINUE
CALL MATRIX (EMAT,VMAT,AMAT,NMAX)
LBIG = 0
VARBIG = 0.0D0
DO 24 L = 1,NMAX
ABSVAR = DABS(VMAT(L))
IF (ABSVAR .LE. VLIM * VAR(L)) GO TO 24
IF (ABSVAR .LE. VARBIG) GO TO 24
LBIG = L
VARBIG = ABSVAR
24 CONTINUE
VRATIO = 1.0D0
IF (LBIG .GT. 0) VRATIO = VLIM * VAR(LBIG) / VARBIG
ERRAVE = 0.0D0
VMTAVE = 0.0D0
DELAVE = 0.0D0
FNMAX = NMAX
DO 25 L = 1,NMAX
DELVAR(L) = VRATIO * VMAT(L)
ERRAVE = ERRAVE + DABS(AMAT(L)) / FNMAX
VAR(L) = VAR(L) + DELVAR(L)
VMTAVE = VMTAVE + DABS(VMAT(L)) / FNMAX
DELAVE = DELAVE + DABS(DELVAR(L)) / FNMAX
25 CONTINUE
IF (MISMAT .GT. 0) GO TO 31
IF (NOMISS .EQ. 0) MISMAT = 1
IF (MISMAT .EQ. 0) IGO = 1
WRITE (8,34) LOOPER
DO 27 I = 1,NMAX
WRITE (8,35) AMAT(I), (EPIAT(I,L), L=1,9), VMAT(I), DELVAR(I), VAR(I)
WRITE (8,36) ERRAVE, VMTAVE, DELAVE
IF (LOOPER .LT. ITRYS) G3 TO 15
CALL ERROR
RETURN
VMFAVX = VMTAVE
DO 30 I = 1,NMAX
AMAT(I) = - ERR(I)
30 GO TO 23
WRITE (8,37) AMAT, ERRAVE, DELVAR, DELAVE, VMAT, VMTAVE, VAR
MISMAT = MISMAT + 1
IF (VMTAVE .LT. VCHNGE * VMTAVX) GO TO 28
WRITE (8,38) IF (MISMAT .LT. NOMISS) NOMISS = 1
MISMAT = 0
LOOP = 0
IGO = 2
GO TO 5
C
C
32 FORMAT (4HOROW,I2,16H IS ZERO IN EMAT)
33 FORMAT (7HOCOLUMN,I2,16H IS ZERO IN EMAT)
34 FORMAT (8H OR 28X,23H ERROR MATRIX AFTER LOOP,I4,29X,4H VMTAVX,
1 6X,6HDELVAR,7X,14HVARIABLE$$$$$$)
35 FORMAT (1H0,F6.4,10P9.3,2P11.4,6H$$$$$$)
36 FORMAT (1H0,F6.4,32X,14H AVERAGE VALUES,31X,2F11.4,6H$$$$$$)
37 FORMAT (12H---- AMAT,10P11.5,6H$$$$$$,/,12H----DELVAR,
1 10P11.6,6H$$$$$$,/,12H---- VMAT,10P11.6,6H$$$$$$,/,2
12H---- VAR,9P11.6,6H$$$$$$)
38 FORMAT (1H0,50X,22HC HANGE TOO SMALL$$$$$$)
END
SUBROUTINE ERROR

IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL ERRER, DUMSPL, FXFN2M, FXM2CP, APTFAN, FAN
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2)
DIMENSION AWORD(2)
EQUIVALENCE (WORD(1), COM(1)), (MODE, COM(6)), (IDUMP, COM(9)),
1 (LOOPER, COM(19)), (ZF, COM(136)), (PCNF, COM(137)),
2 (ZI, COM(139)), (PCNI, COM(140)), (T4, COM(156)), (ZC, COM(300)),
3 (PCNC, COM(301)), (ISPOOL, COM(1044)), (ICOAPB, COM(1045)),
4 (ICODUC, COM(1046)), (ICOMIX, COM(1047)), (ERRE?, COM(1056)),
5 (DUMSPL, COM(1057)), (FXFN2M, COM(1058)), (FXM2CP, COM(1059)),
6 (APTFA, COM(1060)), (FAN, COM(1061))
DATA AWORD /4HCO11M, 4HON /
IF (ICOAPB .LT. 1) ICOAPB = 0
IF (ICOMIX .LT. 1) ICOMIX = 0
IF (ICODUC .LT. 1) ICODUC = 0
IF (ICOAPB .NE. 0) WRITE (6,10) ICOAPB
IF (ICODUC .NE. 0) WRITE (6,11) ICODUC
IF (ICOMIX .NE. 0) WRITE (6,12) ICOMIX
ERRER = .TRUE.
WRITE (6,2) WORD
WRITE (6,3) WORD, ZF,PCNF,ZI,PCNI,ZC,PCNC,T4,MODE
WRITE (6,4) (COM(I), I = 33,394)
WRITE (6,5) (COM(I), I = 33,394)
WRITE (6,7) LOOPER
IF (IDUMP .EQ. 0) GO TO 1
WRITE (6,6) CALL SYG (2)
CALL ENGBAL
RETURN

1

CALL ENGBAL
RETURN

C

2

FORMAT (28HOAN ERROR HAS BEEN FOUND IN ,A4,A2)
3 FORMAT (1HO,A4,A2,9X,7E15.6,I4)
4 FORMAT (2HO )
5 FORMAT (1H ,8E15.6)
6 FORMAT (1H1)
7 FORMAT (25H0FAILED TO CONVERGE AFTER,I4,6H LOOPS)
8 FORMAT (1H ,8E15.6,I12)
9 FORMAT (27H THE ERROR IN COMABN IS AT ,I3)
10 FORMAT (27H THE ERROR IN CODUCT IS AT ,I3)
11 FORMAT (27H THE ERROR IN COMIX IS AT ,I3)
12 END
SUBROUTINE ETAAB

DIMENSION TFAR(25), ETABRT(25), EM6T(7), DELM6(7), P6T(14), DELP6(14), X(3), Y(3)

DATA TFAR/.039D0, .0585D0, .0732D0, .0878D0, .0976D0, .1171D0, .1268D0, .1463D0, .1619D0, .1834D0, .1951D0, .2195D0, .2439D0, .2927D0, .3415D0, .4146D0, .4538D0, .5366D0, .6341D0, .7317D0, .8293D0, .9268D0, 1.00D0, 1.0634D0, 1.70D0/

DATA ETABRT/.94D0, -9887D0, 1.0193D0, 1.0306D0, 1.0227D0, 1.0672D0, .9377D0, .9207D0, .9354D0, .9626D0, .9773D0, .9733D0, 1.0193D0, 1.0532D0, 1.077D0, .0781D0, 1.077D0, 1.0747D0, 1.0668D0, 1.0578D0, 1.051D0, 1.0374D0, 1.0192D0, 1.00D0, .9626D0, .9151D0, .0D0, .9626D0, .9151D0/

DATA EM6T/1.00D0, 1.071D0, 1.19D0, 1.3090D0, 1.428D0, 1.547D0, 1.666D0/

DATA DELM6/0.0D0, -0.13D0, -0.041D0, -0.073D0, -0.11D0, -0.147D0, -0.187D0/

DATA P6T/-.22DO, -.2267D0, -.25DO, -.3008, -.3333D0, -.3767D0, -.4167D0/

DATA DELP6/-.142DO, -.125DO, -.1DO, -.075DO, -.062DO, -.05DO, -.041DO, -.027DO, .019D0, -.013D0, -.01D0, -.008D0, -.004DO, -.0021DO, 0.0D0/

IMPLICIT REAL*8 (A-H,O-Z)

DATA P6T/.22DO, -.2267D0, -.25DO, -.3008, -.3333D0, -.3767D0, -.4167D0/

IF (IDES .NE. 1) GO TO 5
ECIULT = ETAADS / ETAASV
FMULT = FAR7DS / FAR7SV
AMULT = AM6DS / AM6DSV
PMULT = P6DS / P6DSAV
DO 1 K = 1,25
ETABRT(K) = ETABRT(K) * EMULT
DO 3 K = 1,7
EM6T(K) = EM6T(K) * AMULT
DO 4 M = 1,14
P6T(M) = P6T(M) * PMULT
ETAASV = ETAADS
P6DSAV = P6DS
PARK7SV = PARK7DS
AM6DSV = AM6DS
RETURN

IF (IDES .NE. 1) GO TO 5
EMULT = ETAADS / ETAASV
FMULT = FAR7DS / FAR7SV
AMULT = AM6DS / AM6DSV
PMULT = P6DS / P6DSAV
DO 1 K = 1,25
ETABRT(K) = ETABRT(K) * EMULT
DO 3 K = 1,7
EM6T(K) = EM6T(K) * AMULT
DO 4 M = 1,14
P6T(M) = P6T(M) * PMULT
ETAASV = ETAADS
P6DSAV = P6DS
PARK7SV = PARK7DS
AM6DSV = AM6DS
RETURN

1 TFAR(K) = TFAR(K) * PMULT
2 0.0D0/

CONTINUE

DO 7 I = 1,3
NN = N - 1 + I
X(I) = TFAR(NN)

CALL PARABO (X,Y,FAR,ETA1)
GO TO 9

ETA1 = - 2.0D0 * FAR + .1948D0

IF (PAR .GT. 0.067D0) GO TO 8

DO 6 J = 1,25
IF (PAR .GE. TFAR(J)) N = J - 1

CONTINUE

IF (N .EQ. 0) N = 1
IF (N .GE. 20) N = 23
DO 7 I = 1,3
NN = N - 1 + I
X(I) = TFAR(NN)

CALL PARABO (X,Y,FAR,ETA1)
GO TO 9

ETA1 = - 2.0D0 * PAR + .1948D0
M = 0
DO 10 J = 1,7
IF (EM6 .GE. EM6T(J)) M = J - 1
CONTINUE
IF (M .EQ. 0) M = 1
IF (M .GE. 6) M = 5
DO 11 I = 1,3
MM = M - 1 + I
X(I) = EM6T(MM)
Y(I) = DELM6(MM)
CALL PARABO (X,Y,EM6,COR1)
L = 0
DO 12 J = 1,14
IF (P6 .GE. P6T(J)) L = J - 1
IF (L .EQ. 0) L = 1
IF (L .GE. 13) L = 12
DO 13 I = 1,3
LL=L-1+1
X(I) = P6T(LL)
CALL PARABO (X,Y,P6,COR2)
ETA = ETA1 * (1.0D0 - COR1) * (1.0D0 + COR2)
RETURN
END

SUBROUTINE FASTBK
IMPLICIT REAL*8 (A-A,Z)
LOGICAL FAN
COMMON /COMALL/ COM (1062)
EQUIVALENCE (PI, COH (33)), (T25, CON (fig)), (P25, "OM (47)), 1 (H25, COM (48)), (S25, COM (49)), (WFD, COM (72)), (XXP1, COM (76)), 2 (XWG24, COM (77)), (XP25, COM (80)), (XH25, COM (81)), (XS25, COM (82)), 3 (XP55, COM (83)), (XH55, "OM (87)), (XS55, COM (88)), 4 (XWFB, COM (89)), (XXP55, COM (90)), (XP55, COM (92)), 5 (T21, "OM (263)), (H21, "OM (264)), (S21, "OM (265)), 6 (T55, "OM (272)), (H55, "OM (273)), (S55, "OM (274)), 7 (BLF, "OM (316)), (WAP, "OM (319)), (WG24, "OM (321)), 8 (P21, "OM (377)), (P55, COM (387)), (PAN, "OM (1061))
XT55 = T55
XP55 = P55
XH55 = H55
XS55 = S55
IF (FAN) GO TO 1
T25 = T21
P25 = P21
H25 = H21
S25 = S21
WG24 = WAP - BLF
XT25 = T25  
XP25 = P25  
XH25 = H25  
XS25 = S25  
XWFB = WFB  
XWG55 = WG55  
XPA55 = FAR55  
XWFD = WFD  
XWG24 = WG24  
XPA24 = FAR24  
XP1 = P1  
CALL COMIX  
RETURN  
END

Subroutine FCNTRL

SUBROUTINE FCNTRL  
IMPLICIT REAL*8 (A-H, O-Z)  
RETURN  
END

Subroutine FRTOSD

SUBROUTINE FRTOSD  
IMPLICIT REAL*8 (A-H, O-Z)  
LOGICAL FAN  
COMMON /COMALL/ COM(1062)  
EQUIVALENCE (P1, COM(33)), (H3, COM(153)), (WAC, COM(191)),  
1 (XP1, COM(200)), (XT21, COM(231)), (XP21, COM(202)),  
2 (XH21, COM(203)), (XS21, COM(204)), (XH3, COM(205)),  
3 (XWAF, COM(206)), (XWAC, COM(207)), (XBFL, COM(203)),  
4 (XBFL, COM(209)), (T21, COM(253)), (H21, COM(264)),  
5 (S21, COM(265)), (BLF, COM(316)), (BLDU, COM(317)),  
6 (WAF, COM(319)), (P21, COM(377)), (FAN, COM(1061))  
XP1 = P1  
XWAF = WAF  
XWAC = WAC  
XBFL = BLF  
XBFLU = BLDU  
XH3 = H3  
XT21 = T21  
XP21 = P21  
XH21 = H21  
XS21 = S21  
IF (FAN) CALL CODUCT  
IF (.NOT. FAN) CALL FASTBK  
RETURN  
END

Function GUESS

FUNCTION GUESS (M, T, TD, P, PD, W, WD, D, DD, VD)  
IMPLICIT REAL*8 (A-H, O-Z)  

116
SUBROUTINE INDURY (CNI, ZI, WACI, IDES)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /COMDAT/ C0MD(5423)
DIMENSION CNX1(15), PRXXI(15,15), WACXXI(15,15), ETAXXI(15,15),
1 NPTXI(15)
DIMENSION XCNXX(15), WACAR(15)
EQUIVALENCE (CNX1X(1), COED(1381)), (PRXXI(1,1), COMD(1396)),
1 (WACXXI(1,1), COMD(1621)), (ETABXXI(1,1), COMD(1846)),
2 (NCNXI, COMD(5328)), (PRXXI(1,1), COMD(5329))
DATA XCNXX /1.00D0, 1.00D0, 1.00D0, 1.00D0, 1.00D0, 1.00D0, 1.00D0, 1.00D0,
1 2.00D0, 2.00D0, 2.00D0, 2.00D0, 2.00D0, 2.00D0, 2.00D0/,
DATA WACAR /1.00D0, 1.00D0, 1.00D0, 1.00D0, 1.00D0, 1.00D0, 1.00D0,
1 1.50D0, 1.00D0, 0.80D0, 0.60D0, 0.40D0, 0.25D0, 0.10D0, 0.05D0/
IF (IDES .NE. 1) GO TO 1
WAIDS = WACI
CNIDS = CNI
ZI = 2.00D0 / 3.50D0
1 NCNXI = 15
DO 2 I = 1,15
NPTXI(I) = 15
CNXXI(I) = XCNXX(I) * CNIDS
2 DO 2 J = 1,15
PRXXI(I,J) = FLOAT(J + 3) / 4.00D0
ETAXXI(I,J) = 1.00D0
WACXXI(J,I) = WACAR(I) * (0.993D0 + 0.001D0 * FLOAT(J)) * WAIDS
RETURN
END
SUBROUTINE MATRIX (E,V,A,N)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION E(9,9), V(9), A(9), PIV(10), T(9,10)
NN = N + 1
NM = N - 1
DO 1 I = 1, N
T(I,NN) = A(I)
DO 1 J = 1, N
T(I,J) = E(I,J)
1 CONTINUE
DO 2 J = 1, N
IF (TEMP .GT. DABS(T(J,I))) GO TO 2
TEMP = DABS(T(J,I))
2 CONTINUE
IPIV = J
DO 3 J = IPIV, NN
PIV(J) = T(IPIV,J) / T(IPIV,IPIV)
3 CONTINUE
IF (IPROM .EQ. IPIV) GO TO 6
PM = - T(IPROM,I)
DO 5 J = IP1, NN
T(IPOM,J) = T(IPOM,J) + PM * PIV(J)
5 CONTINUE
IPROM = IPROM - 1
DO 7 J = IP1, NN
T(J,I) = PIV(J)
7 CONTINUE
DO 8 I = 1, NM
J = NN - I
K = N - I
DO 8 L = J, N
T(K,NN) = T(K,NN) - T(K,L) * T(L,NN)
8 CONTINUE
DO 9 I = 1, N
V(I) = T(I,NN)
9 RETURN
END

Subroutine NOZCTR

SUBROUTINE NOZCTR
IMPLICIT REAL*8 (A-H,O-Z)
RETURN
END

Subroutine OUTPUT

SUBROUTINE OUTPUT
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL DUNSPPL, FXPN2M, PXM2CP, AFTEAN, PAN
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2)
DIMENSION W(6,5), AWORD1(2), AWORD2(2)

EQUIVALENCE (WORD(1), COM(1)), (IDES, COM(3)), (MODE, COM(6)), (PCNI, COM(140)), (ETAR, COM(187)), (ETAR, COM(187)), (FXFNI211, CON(1058)), (FXNZCP, COM(1059)), (FAN, COM(1061))

DATA W /4HOUTP, 4HUT, 4HCOMM, 4HON /
DATA AWORD1, AWORD2 /4HOUTP, 4HUT, 4HCOMM, 4HON /

IF (IAMTRX .EQ. 1 .AND. ITRAN .EQ. 1) GO TO 24
WRITE (6,29) TIME
WORD(1) = AWORD1(1)
WORD(2) = AWORD1(2)
IF (IDBURN .GT. 0) GO TO 2
IF (IAPTBN .GT. 0) GO TO 1
WRITE (6,7) WORD, AMC, ALTP, T4, ETAR
GO TO 3
WRITE (6,8) WORD, AMC, ALTP, T4, T7, ETAR
GO TO 3
WRITE (6,9) WORD, AMC, ALTP, T4, T24, ETAR
3 IF (FXFNI2M) WRITE (6,17)
IF (FXM2CP) WRITE (6,18)
IF (FAN) GO TO 25
WRITE (6,26) ISPOOL
GO TO 27
25 IF (.NOT. FXFNI2M) AND (.NOT. FXM2CP) AND (.NOT. DUMSPL)
1 WRITE (6,19)
IF (DUMSPL) WRITE (6,23)
IF (PCBLID .EQ. 0.0D0) WRITE (5,20)
27 CALL CONOUT(2)
WRITE (6,10) (W(I, IMSHOC),I=1,6), FG, FN, SPC
IF (IGASHX .GT. 0 .OR. .NOT. PAN) GO TO 4
WRITE (6,11) (W(I, IMSHOC),I=1,6)
WRITE (6,12) LOOPER
IF (IDES .NE. 1) GO TO 5
WORD(1) = AWORD2(1)
WORD(2) = AWORD2(2)
WRITE (6,13) WORD, ZF, PCNP, ZI, PCNI, ZC, PCNC, T4, MODE
WRITE (6, 14)
WRITE (6, 15)  (COM(I), I = 33, 334)
WRITE (6, 14)
WRITE (6, 15)  DUMPSL, FXPN2M, FXM2CP, AFTFAN, PAN
WRITE (6, 14)
WRITE (6, 16)
24 IF (IDES .EQ. 1) GO TO 6
5 A8 = A8SAV
A9 = A9SAV
A28 = A28SAV
A29 = A29SAV
IF (IDUMP .NE. 2) GO TO 5
WRITE (6, 16)
CALL SYG (2)
6 CALL ENGBAL
RETURN
C
C
C
7 FORMAT (1H , A4, A2, 14X, 7H AM=, F7.3, 6X, 7H ALTP=, F7.0,
1 6X, 7H T4=, F8.2, 25X, 7H ETAR=, F7.4)
8 FORMAT (1H, A4, A2, 14X, 7H AM=, F7.3, 6X, 7H ALTP=, F7.0,
1 6X, 7H T4=, F8.2, 5X, 7H T7=, F8.2, 5X, 7H ETAR=, F7.4)
9 FORMAT (1H, A4, A2, 14X, 7H AM=, F7.3, 6X, 7H ALTP=, F7.0,
1 6X, 7H T4=, F8.2, 5X, 7H T24=, F8.2, 5X, 7H ETAR=, F7.4)
10 FORMAT (1H0, 5HMAIN, 6A4, 9X, 3HFG=, F9.2, 18X, 3HFN=, F9.2,
1 18X, 4HSFC=, F8.5)
11 FORMAT (6H DUCT, 6A4)
12 FORMAT (16HOCONVERGED AFTER, I4, 5H LOOPS, /, 1H1)
13 FORMAT (1H , A4, A2, 9X, 7E15.6, I4)
14 FORMAT (1H )
15 FORMAT (1H , 8E15.6)
16 FORMAT (1H1)
17 FORMAT (5HOPAN AND MIDDLE SPOOL ARE ATTACHED, USE INNER AND,
1 14HOUTER TURBINES)
18 FORMAT (49HOMIDDLE AND COMPRESSOR SPOOLS ARE ATTACHED, USE,
1 125HMAIN AND OUTER TURBINES)
19 FORMAT (19HTHREE SPOOL ENGINE)
20 FORMAT (21HNO AIRFLOW INTO WING)
21 FORMAT (1H+, 22X, 14H, AFT-TURBOFAN)
22 FORMAT (14H0 AFT-TURBOFAN)
23 FORMAT (22HOMIDDLE SPOOL IS DUMMY)
26 FORMAT (1H0, I4, 15H SPOOL TURBOJET)
29 FORMAT (1H1, 20X, 7H TIME=, F7.4)
END

Subroutine PARABO

SUBROUTINE PARABO (X, Y, XD, YANS)
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION X(3), Y(3)
X1M2 = X(1) - X(2)
X1M3 = X(1) - X(3)
X2M1 = X(2) - X(1)
\[ X_{2M3} = X(2) - X(3) \]
\[ X_{3M2} = X(3) - X(2) \]
\[ X_{1SQ} = X(1) \times X(1) \]
\[ X_{2SQ} = X(2) \times X(2) \]
\[ X_{3SQ} = X(3) \times X(3) \]
\[ Y_{1M2} = Y(1) - Y(2) \]
\[ Y_{1M3} = Y(1) - Y(3) \]

\[ A = (X_{1M2} \times Y_{1M3} - X_{1M3} \times Y_{1M2}) / X_{1M2} / X_{1M3} / X_{3M2} \]
\[ B = ((X_{1SQ} - X_{2SQ}) \times Y_{1M3} - (X_{1SQ} - X_{3SQ}) \times Y_{1M2}) / X_{1M2} / X_{1M3} / X_{3M2} \]

\[ D = (Y(1) \times Y(2) \times X_{1SQ} - X(2) \times X(1) \times X_{2M1}) / (X_{2SQ} \times X_{1SQ}) \]

\[ \text{RETURN} \]

Subroutine PERF

```
SUBROUTINE PERF
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI, DUMSPL, AFTFAN, FAN
COMMON /COMALL/ COM(1062)
DIMENSION WORD(2), AWORD(2)
EQUIVALENCE (WORD(1), 'COM(1)'), (IDES, COM(3)), (IGASMX, COM(10)), (PA, PE, 4HRF)
1 (PI, COM(33)), (A29, COM(50)), (V29, COM(63)), (PS29, COM(65))
2 (WAD, COM(71)), (WFD, COM(72)), (PCNI, COM(140))
3 (WPB, COM(192)), (CS, COM(194)), (AM, COM(195))
4 (WG37, COM(210)), (A39, COM(213)), (V39, COM(221))
5 (PS39, COM(226)), (PGMWNG, COM(230)), (PGPWNG, COM(231))
6 (PFWING, COM(232)), (PMAIN, COM(233)), (PMOFPN, COM(234))
7 (DELPF, COM(236)), (DFLN, COM(237)), (DFLSPC, COM(238))
8 (CVDNWG, COM(239)), (C VN, COM(240)), (CVMNOZ, COM(241))
9 (VA, COM(242)), (VJD, COM(243)), (VJW, COM(244))
1 (VJM, COM(245)), (WFT, COM(246)), (WGT, COM(247))
2 (SPC, COM(248)), (TFAR, COM(249)), (FRD, COM(250))
3 (FGND, COM(251)), (PFMM, COM(252)), (PFDP, COM(253))
4 (FPGM, COM(254)), (PGM, COM(255)), (PGF, COM(256))
5 (FG, COM(257)), (FN, COM(258)), (FPFPN, COM(259))
6 (FCOVFN, COM(260)), (PMOFPN, COM(261)), (PMOFPD, COM(262))
7 (WA32, COM(271)), (PCBLID, COM(305)), (CNI, COM(309))
8 (WAI, COM(311)), (BLOB, COM(318)), (WAP, COM(319))
9 (WG24, COM(321)), (WG7, COM(334)), (A9, COM(356))
1 (PS9, COM(359)), (V9, COM(360)), (WPA, COM(368))
EQUIVALENCE (TIME, COM(993)), (TPRINT, COM(996))
1 (ITRAN, COM(1049)), (SI, COM(1055)), (DUMSPL, COM(1057))
2 (APFTAN, COM(1060)), (FAN, COM(1061))
DATA AWORD /4A PE, 4HRF /
WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
IF (SI) GO TO 100
G = 32.174049D0
CAPSF = 2116.2170D0
GO TO 101
```
100 G = 1.0DO
CAPSF = 1.0DO
101 WPT = WFB + WFD + WFA
WAT = WAF - BLOB
IF (APTFAN) WAT = WAT + WAI
WGT = WAT + WPT
TFAR = WPT / WAT
VA = AM * CS
PRD = VA * WAF / G
IF (APTFAN) PRD = PRD + VA * WAI / G
VJM = CVMOZ * V9
FGMM = VJM * WG7 / G
FGPM = CAPSF * (PS9 - P1) * A9
IF (IGASMX.GT. 0 .OR. .NOT. FAN) GO TO 1
VJD = CVMOZ * V29
FGMD = VJD * WG24 / G
FGPD = CAPSF * (PS29 - P1) * A29
1
VJW = 0.0DO
PGMNG = 0.0DO
PGPWNG = 0.0DO
PFWNG = 0.0DO
FNPWNG = 0.0DO
IF (PCBLID .EQ. 0.0DO) GO TO 2
VJW = CVMOZ * V39
PGMNG = VJW * WG37 / G
PGPWNG = CAPSF * (PS39 - P1) * A39
FGWING = FGMMNG + FGPWNG
FNPWNG = FGWING - VA * WA32 / G
2
FGM1 = FGMM + FGMD
PGM = FGM1 + FGMNG
FGP1 = FGM1 + FGPD
FGP = FGP1 + FGPWNG
FNMAIN = FGM1 + FGP1 - VA * (WAF - WA32) / G
IF (APTFAN) FNMAIN = FNMAIN - VA * WAI / G
PG = FGM + FGP
FN = FG - PRD
SFC = 3600.0DO * WFT / FN
FG = DELPG * FG
FN = DELEN * FN
SFC = DELSFC * SFC
FPAN = FGM + FGP - VA * WAD / G
FCORE = FNMAIN - FPAN
FPVFN = FPAN / FN
FCVFN = FCORE / FN
FWVFN = FNWING / FN
FMNOP = FNMAIN / FN
IF (IDES .EQ. 1) PDES = FN
FNQP = FN / FDES
IF (.NOT. DUMSPL) GO TO 3
PCNI = 1.0DO
CNI = 0.0DO
3
IF (ITIFAN .EQ. 1 .AND. TIME .LT. TPFINT) CALL ENGBAL
CALL OUTPUT
CALL ERASOR
RETURN
END
SUBROUTINE PROCOM (FARX,TEX,CSEX,AKEX,CPEX,REX PHI,HEX)

IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI
COMMON /COMALL/ COM(1062)
EQUIVALENCE (SI, COM(1055))

C IF SI UNITS ARE USED, CONVERT TEX TO DEGREES FANKINE
IF (SI) TEX = TEX * 9.0D0 / 5.0D0
IF (FARX .GT. .067523D0) FARX = .067523D0
IF (TEX .LT. 300.0D0) TEX = 300.0D0
IF (TEX .GT. 4000.0D0) TEX = 4000.0D0
IF (FARX .LT. O.0D0) PARX = O.0D0

C AIR PATH
CPA = ((((1.011554D-25 * TEX - 1.452677D-21) * TEX +
7.621576D-18) * TEX - 1.5128259D-14) * TEX - 6.7179376D-12) *
2 TEX + 6.5519486D-08) * TEX - 5.1536879D-05) * TEX + 2.5020051D-01
HEA = ((((1.2644425D-26 * TEX - 2.0752522D-22) * TEX +
1.270263D-18) * TEX - 3.0256518D-15) * TEX - 1.6794594D-12) *
2 TEX + 2.1839826D-08) * TEX - 2.576844D-05) * TEX +
3.020051D-01) * TEX - 1.755886D+00
SEA = 2.5020051D-01 * DLOG(TEX) + ((((1.4450767D-25 * TEX -
1.2411288D-22) * TEX + 1.5243153D-16) * TEX - 3.792364RD-15) *
2 TEX - 2.239279D-12) * TEX + 3.73816638D-02)
IF (PARX .LE. O.0D0) GO TO 5

C FUEL/AIR PATH
CPF = ((((7.267871D-25 * TEX - 1.3335668D-20) * TEX +
1.0212913D-16) * TEX - 4.2051104D-13) * TEX + 9.9685793D-10) *
2 TEX - 1.3771901D-06) * TEX + 1.225863D-03) * TEX + 7.3816638D-02
HEF = ((((9.0848388D-26 * TEX - 1.905994D-21) * TEX +
1.7021525D-17) * TEX - 2.4102208D-14) * TEX + 2.8921698D-10) *
2 TEX - 4.5906632D-07) * TEX + 6.129315D-04) * TEX +
3.816638D-02) * TEX + 3.058153D+01
SEF = 7.3816638D-02 * DLOG(TEX) + ((((1.038267D-25 * TEX -
1.2226116D-21) * TEX + 2.0425826D-17) * TEX - 1.0512776D-13) *
2 TEX + 3.3228928D-10) * TEX - 5.8859505D-07) * TEX +
3.17021525D-03) * TEX + 6.493399D-01

CPEX = (CPF + FARX * CPF) / (1.0D0 + FARX)
HEX = (HEA + FARX * HEX) / (1.0D0 + FARX)
PHI = (SEA + FARX * PHI) / (1.0D0 + FARX)
AMW = 28.97D0 -.346186DO * AMW
REX = 1.986375D0 / AMW
AKEX = CPEX / (CPEX - REX)
CSEX = DSQRT(AKEX * RX * TEX * 25031.37D0)
IF (.NOT. SI) RETURN
CPEX = CPEX * 4185.7666D0
HEX = HEX * 2325.4259D0
PHI = PHI * 4185.7666D0
REX = REX * 4185.7666D0
CSEX = CSEX * .3048D0
Subroutine PUTIN

IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI, ERRER, DUMPL, FXPN2M, FXM2CP, AFTFAN, PAN
COMMON /COLUMM/ COM(1062)
DIMENSION WORD(2), TIMEP(50)
DIMENSION XSAVE(396), XWORD(2)

EQUIVALENCE (WORD(1), COM(1)), (IDBS, COM(3)), (MODE, COM(6)),
1 (INIT, COM(7)), (DUMP, COM(9)), (IAMTP, COM(9)),
2 (IGSMMX, COM(10)), (IDBURN, COM(11)), (IAPPKN, COM(12)),
3 (IDCD, COM(13)), (IMCD, COM(14)), (NOZFLT, COM(17)),
4 (ITRYS, COM(18)), (TOLALL, COM(23)),
5 (I24, COM(42)), (A26, COM(50)), (ETRD, COM(73)),
6 (DPDUDS, COM(75)), (T2, COM(92)), (P2, COM(93)),
7 (IPCD, COM(14)), (IAlrTP, COM(9)), (ICMP, COM(10)), (I3RIIRN, COM(11)),
8 (IDUMP, COM(12)), (IAFTRH, COM(12)), (ICMP, COM(13)), (IKIT, COM(i)),
9 (ICM, COM(14)), (IPl23, COM(15)), (ITRYS, COM(16)), (TOLALL, COM(23)),
10 (12L1, COM(42)), (A2E, COM(50)), (ETP-D, COM(73)),
11 (DPDUDS, COM(75)), (T2, COM(92)), (P2, COM(93)),
12 (IPCD, COM(14)),

EQUIVALENCE (CVDNZOZ, COM(240)), (CVMNOZ, COM(241)),
1 (TFLPDS, COM(275)), (CNLDS, COM(276)), (ETLPS, COM(277)),
2 (EFIPS, COM(278)), (CNIPDS, COM(279)), (ETFIPS, COM(280)),
3 (PRCS, COM(297)), (ETACDS, COM(298)), (PCNC, COM(301)),
4 (PCBLDC, COM(302)), (PCNCDS, COM(303)), (PCBLIP, COM(304)),
5 (PCBLID, COM(305)), (WACDS, COM(310)), (ZCDS, COM(313)),
6 (AMF, COM(327)), (AF, COM(328)), (T7DS, COM(342)),
7 (T7, COM(343)), (AB, COM(346)), (ETAADS, COM(366)),
8 (DPFADS, COM(367)), (WFA, COM(368)), (ETAADS, COM(366)),
9 (VFAN, COM(395)), (VFAN, COM(395)), (VFAN, COM(395)),
10 (VFAN, COM(395)), (VFAN, COM(395)), (VFAN, COM(395)),
11 (VFAN, COM(395)), (VFAN, COM(395)), (VFAN, COM(395)),
12 (VFAN, COM(395)), (VFAN, COM(395)), (VFAN, COM(395)),

EQUIVALENCE (DUMPL, COM(1057)), (FXPN2M, COM(1058)),
1 (FXM2CP, COM(1059)), (AFTFAN, COM(1060)), (PAN, COM(1061)),
**DATA AWORD /4HN /**

C *** IDES =1 FOR CALCULATING DESIGN POINT
C *** MODE =0 FOR CONSTANT T4
C *** MODE =1 FOR CONSTANT PCNC
C *** MODE =2 FOR CONSTANT WFB
C *** MODE =3 FOR CONSTANT PCNF
C *** INIT =1 WILL NOT INITIALIZE POINT
C *** IDUMP =1 WILL DUMP LOOPING WRITE-OUTS IF ERROR OCCURS
C *** IDUMP =2 WILL DUMP LOOPING WRITE-OUTS AFTER EVERY POINT
C *** IAMTP =0 WILL USE INPUT AM AND MIL SPEC ETAR
C *** IAMTP =1 WILL USE INPUT AM AND INPUT ETAP
C *** IAMTP =2 WILL USE T2 AS T1+P2 AND STANDARD P1
C *** IAMTP =3 WILL USE P2 AND STANDARD T1
C *** IAMTP =4 WILL USE T2 AND P2
C *** IAMTP =5 WILL USE RAM2 FOR SPECIAL RECOVERY
C *** IGASMX=-1 SEPARATE FLOW, INPUT A6
C *** IGASMX=0 SEPARATE FLOW, A6=A55
C *** IGASMX=1 WILL MIX DUCT AND MAIN STREAMS, A6=A25+A55
C *** IGASMX=2 WILL MIX DUCT AND MAIN STREAMS, INPUT A6
C *** IDBURN=1 FOR DUCT BURNING, INPUT T24
C *** IDBURN=2 FOR DUCT BURNING, INPUT WFD
C *** IAFTBN=1 FOR AFTERBURNING, INPUT T7
C *** IAFTBN=2 FOR AFTERBURNING, INPUT WFA
C *** IDCD =1 DUCT NOZZLE WILL BE C-D
C *** IMCD =1 MAIN NOZZLE WILL BE C-D
C *** NOZFLT=1 FOR FLOATING MAIN NOZZLE
C *** NOZFLT=2 FOR FLOATING DUCT NOZZLE
C *** NOZFLT=3 FOR FLOATING MAIN AND DUCT NOZZLES
C *** ITRYS =N NUMBER OF PASSES THRU ENGINE BEFORE QUITTING

**NAMELIST /DATAIN/ ISPOOL,FAN,SI,DELT1,IDES,MODE,IDUMP,IAMTP,
1 IGA5MX,IBURN,IAFTBN,ICD,IMCD,N0ZFLT,ITRYS,FXFN2M,FM2CP,
2 APTFAN,DUMSPL,TOLALL,DELFG,DELSP,PCNPDS,PRFDS,ETBPS,
3 PCNDPS,PRCDPS,ETACDS,T4DS,WFDS,ETABDS,PCDPS,ETLPDS,
4 DPUPS,TD1S,ETAADS,DPFADS,A6,A3,A28,PS55,AM55,CVNOZ,CVMNOZ,T2,
5 P2,T4,WAFCSDS,WACCDS,HPEXT,AM,ALTE,ETAR,PCNP,PCNC,WPE,PCBLS,PCBL,PCNI,
6 PCBL,PCBLDU,PCBLB,PCBLP,PCBL,PCBLS,TD2,ETAD,TD7,WFA,ETAA,AM6,AM23,
7 DPEWDS,AM3,PCNIDS,PCBDPS,PCBLS,PCBLS,ZCAP,ZCAPS,ZCDZ,PCBLS,TPHEPS,CMHPS,
8 TPIEPS,CMIDP,FILPS,CMIDPS,PRIDS,ETAINS,ETIPDS,WAPIDS,PLAPS,
9 CVWNG,ITRAN,DTRNNT,TF,INIT,DT,XNHDPS,XNLPDS,PNHP,SNIHP,PNHP,
1 PMIH,SPI,SPI,VCMP,VCMP,VPTRB,VPTRB,VPTRB,VPTRB,VPTRB,VPTRB,
2 VDFUSC,ITRANX

WORD(1) = AWORD(1)
WORD(2) = AWORD(2)
ITRAN = 0
JTRAN = 0
TIME = 0.0D0
NSTEP = 0
TPRINT = 0.0D0
DTPRNT = 0.0D0
CALL ZERO
IF (KGO .EQ. 1) GO TO 5
IDES = 0
READ (5,DATAIN)
IF (.NOT. ERRER) GO TO 102
IF (IAPTBn .GT. 0 .OR. IDBURN .GT. 0 .OR. NOZFLT .GT. 0) GO TO 1 A3CD4540
102 ERRER = .FALSE. A3CD4542
C TABLE IS REFERENCED TO COMMON/ALL/FIRST ENTRY A3CD4543
IF (IDES .EQ. 0) GO TO 7 A3CD4544
IF (KKGO .NE. 2) GO TO 3 A3CD4545
TEMPL = COM(325) A3CD4546
TEMP2 = COM(326) A3CD4547
TEMP3 = COM(336) A3CD4548
TEMP4 = COM(370) A3CD4549
DO 2 I = 1,392 A3CD4550
2 COM(I+2) = XSAVE(I) A3CD4551
DO 205 I = 1,4 A3CD4552
205 COM(I+1056) = XSAVE(I+392) A3CD4553
COM(325) = TEMP1 A3CD4554
COM(326) = TEMP2 A3CD4555
COM(336) = TEMP3 A3CD4556
COM(370) = TEMP4 A3CD4557
READ (5,DATAIN) A3CD4558
C SAVE INPUT IN CASE OF LOOP ON PRESSURE PATIOS A3CD4559
3 TEMP1 = XSAVE(323) A3CD4560
TEMP2 = XSAVE(324) A3CD4561
TEMP3 = XSAVE(334) A3CD4562
TEMP4 = XSAVE(368) A3CD4563
DO 4 I = 1,392 A3CD4564
4 XSAVE(I) = COM(I+2) A3CD4565
DO 405 I = 1,4 A3CD4566
405 XSAVE(I+392) = COM(I+1056) A3CD4567
XSAVE(323) = TEMP1 A3CD4568
XSAVE(324) = TEMP2 A3CD4569
XSAVE(334) = TEMP3 A3CD4570
XSAVE(368) = TEMP4 A3CD4571
GO TO 7 A3CD4572
5 TEMP1 = COM(325) A3CD4573
TEMP2 = COM(326) A3CD4574
TEMP3 = COM(336) A3CD4575
TEMP4 = COM(370) A3CD4576
DO 6 I = 1,392 A3CD4577
6 COM(I+2) = XSAVE(I) A3CD4578
DO 605 I = 1,4 A3CD4579
605 COM(I+1056) = XSAVE(I+392) A3CD4580
COM(325) = TEMP1 A3CD4581
COM(326) = TEMP2 A3CD4582
COM(336) = TEMP3 A3CD4583
COM(370) = TEMP4 A3CD4584
WRITE (6,8) PRFDS,PRFNEW,PRCDS,PRCNEW A3CD4585
PRCDS = PRCNEW A3CD4586
PPFDS = PRFNEW A3CD4587
7 IF (DUMSPL) WAICDS = WACCDS A3CD4588
IF (IAPTBn .GT. 0 .OR. IDBURN .GT. 0 .OR. NOZFLT .GT. 0) INIT = 1 A3CD4589
IF (MODE .EQ. 0) WRITE (3,9) IDES,AM,ALT,TP,T4,T24,T7 A3CD4590
IF (MODE .EQ. 1) WRITE (3,10) IDES,AM,ALT,PCNC,T24,T7 A3CD4591
IF (MODE .EQ. 2) WRITE (8,11) IDES,AM,ALT,W,TP,T24,T7 A3CD4592
IF (DUMSPL) WAICDS = WACCDS A3CD4593
IF (IDES .NE. 1) GO TO 101

C

101 CALL COINLT
RETURN

C

8 FORMAT (18, CHANGE PR LDS FROM, F9.3, 4H TO, F9.3, 17H AND PR CD$ADCDB001
           1 FROM, F10.3, 4H TO, F10.3)  ABCD4602
9 FORMAT ((HC, 7H IDES=, I3, 10X, 7H AM=, F7.3, 6X, 7H ALTP=, ABCD4603
           1 F7.0, 6X, 7H T4=, F8.2, 5X, 7H T24=, F8.2, 5X, 7H T7=, ABCD4604
           2 F8.2, 6H$$$$$)
10 FORMAT (1HC, 7H IDES=, I3, 10X, 7H AM=, F7.3, 6X, 7H ALTP=, ABCD4605
           1 F7.0, 6X, 7H PCNC=, F8.3, 5X, 7H T24=, F8.2, 5X, 7H T7=, ABCD4606
           2 F8.2, 6H$$$$$)
11 FORMAT (1HC, 7H IDES=, I3, 10X, 7H AM=, F7.3, 6X, 7H ALTP=, ABCD4607
           1 F7.0, 6X, 7H WB=, F8.4, 5X, 7H T24=, F8.2, 5X, 7H T7=, ABCD4608
           2 F8.2, 6H$$$$$)
END

Subroutine RAM

SUBROUTINE RAM (AM, ETAR)
IMPLICIT REAL*8 (A-H, O-Z)
ETAR = 1.0DO
IF (AM .LE. 1.0DO) RETURN
IF (AM .GT. 5.0DO) GO TO 3
ETAR = 1.0DO - 0.075DO * ((AM - 1.0DO) ** 1.35DO)
RETURN
ETAR = 800.000 / ((AM ** 4) + 935.000)
RETURN
END

Subroutine RAM2

SUBROUTINE RAM2 (AM, ETAH)
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION PR IN LT(15), F MN(15), X(1), Y(3)
DATA FMN /0.000, .100, .200, .300, .400, .500, .600, 1.000, 1.200, 1.400, 1.600, 1.800, 2.000, 2.400, 2.700/
DATA PR IN LT /9.000, .9200, .9500, .9800, .9900, .9700, .9700, .9600, .9600, .9600, .9600, .9600, .9600, .9600/
M = 0
DO 1 J = 1, 15
IF (AM . GE. FMN(J)) M = J - 1
1 CONTINUE
IF (M .EQ. 0) M = 1
IF (M .GE. 14) M = 13
DO 2 I = 1, 3
MM = M - 1 + I
X(I) = FMN(MM)
2 Y(I) = PR IN LT(MM)
CALL PAPARO (X, Y, AM, ETAR)
RETURN
END
Subroutine ROLL

SUBROUTINE ROLL
IMPLICIT REAL*8 (A-H, O-Z)
COMMON /CROMALL/ COM(1062)
DIMENSION FO(50,4), SO(10,6), PDATA(5,50), TIMEPT(50)
EQUIVALENCE (FO(1,1), COM(430)), (SO(1,1), COM(631)),
1 (PDATA(1,1), COM(691)), (TIMEPT(1), COM(941))
DO 1 I = 1, 50
  FO(I, 2) = FO(I, 1)
  FO(I, 4) = FO(I, 3)
DO 2 I = 1, 10
  SO(I, 6) = SO(I, 5)
  SO(I, 5) = SO(I, 4)
  SO(I, 4) = SO(I, 3)
  SO(I, 3) = SO(I, 2)
2  DO 3 I = 1, 49
3  N1 = 51 - I
  NO = 50 - I
  TIMEPT(N1) = TIMEPT(NO)
DO 3 J = 1, 5
3  PDATA(J, N1) = PDATA(J, NO)
RETURN
END

Subroutine SEARCH

SUBROUTINE SEARCH (P, A, B, C, D, AX, NA, BX, CX, DX, NO, NAM, NOM, NCODE)
IMPLICIT REAL*8 (A-H, O-Z)
COMMON /CROMALL/ COM(1062)
DIMENSION AX(NAM), BX(NAM, NOM), CX(NAM, NOM), DX(NAM, NOM),
1 NC(NAM), Q(9)
EQUIVALENCE (TOLALL, COM(23))
C *** NEEDS SUBROUTINE AFQUIR
C *** AX AND BX MUST BE STORED LO TO HI
C *** P=INPUT PROPORTION BETWEEN 0.0 AND 1.0
C IF NOT INPUT, P MUST EQUAL -1.
C *** FIND A
DO 1 I = 1, NA
  IH = I
  IF (A .LT. AX(I)) GO TO 2
ABCDA4664
ABCDA4665
ABCDA4666
ABCDA4667
ABCDA4668
ABCDA4669
ABCDA4670
ABCDA4671
ABCDA4672
ABCDA4673
ABCDA4674
ABCDA4675
ABCDA4676
ABCDA4677
ABCDA4678
ABCDA4679
ABCDA4680
ABCDA4681
ABCDA4682
ABCDA4683
ABCDA4684
ABCDA4685
ABCDA4686
ABCDA4687
ABCDA4688
ABCDA4689
1 CONTINUE
IF (A .GT. AX(IH)) NCODE = 2
A = AX(IH)
GO TO 3
2 IF (IH .GT. 1) GO TO 3
NCODE = 1
IH = 2
A = AX(1)
3 IL = IH - 1
LIMH = NO(IH)
LIML = NO(IL)
C *** FIND B
PRM = (A - AX(IL)) / (AX(IH) - AX(IL))
PP = P
IF (P .GE. 0.0D0) GO TO 5
BL = EXTR(BX(IL,1), PRM, BX(IH,1))
BH = EXTR(BX(IL,LIML), PRM, BX(IH,LIMH))
IF (B .GE. BL) GO TO 4
NCODE = NCODE + 10
B = BL
GO TO 5
4 IF (B .LE. BH) GO TO 5
NCODE = NCODE + 20
BH = EXTR(BX(IL,LIML-1), PRM, BX(IH,LIML-1))
CHM = EXTR(CX(IL,LIML-1), PRM, CX(IH,LIML-1))
DHM = EXTR(DX(IL,LIML-1), PRM, DX(IH,LIML-1))
CH = EXTR(CX(IL,LIML), PRM, CX(IH,LIML))
DH = EXTR(DX(IL,LIML), PRM, DX(IH,LIMH))
CSLOPE = (CH - CHM) / (BH - BHM)
DSLOPE = (DH - DHM) / (BH - BHM)
C = CH + CSLOPE * (B - BH)
D = DH + DSLOPE * (B - BH)
RETURN
5 PP = 0.5D0
Q(2) = 0.0D0
Q(3) = 0.0D0
6 BH = EXTR(BX(IH,1), PP, BX(IH,LIMH))
BL = EXTR(BX(IL,1), PP, BX(IL,LIML))
DO 7 J = 2, LIMH
JH = J
IF (BH .LT. BX(IH,J)) GO TO 8
CONTINUE
7 CONTINUE
8 JL = JH - 1
DO 9 K = 2, LIML
KH = K
IF (BL .LT. BX(IL,K)) GO TO 10
9 CONTINUE
10 KL = KH - 1
PR = (BX(IH,JL) - BH) / (BX(IH,JH) - BX(IH,JL))
CH = EXTR(CX(IH,JL), -PR, CX(IH,JH))
DH = EXTR(DX(IH,JL), -PR, DX(IH,JH))
PR = (BX(IL,KL) - BL) / (BX(IL,KH) - BX(IL,KL))
CL = EXTR(CX(IL,KL), -PR, CX(IL,KH))
DL = EXTR(DX(IL,KL), -PR, DX(IL,KH))
BT = EXTR(BL, PRM, BH)
CT = EXTR(CL, PRM, CH)
DT = EXTR(DL, PRM, DH)
IF (F .GE. 0.0D0) GO TO 13
DIR = DSQRT(B / BT)
ERR = (B - BT) / B
CALL AFQUIR (Q(1), PP, ERR, 0.0D0, 25.0D0, 1.0D0*TOLALL, DIR, PT, ICON)
GO TO (11, 13, 12), ICON

11 PP = PT
IF (PP .LT. 0.0D0) PP = 0.0D0
IF (PP .GT. 1.0D0) PP = 1.0D0
GO TO 6

12 NCODE = 7

13 B = BT
C = CT
D = DT
RETURN
END

Subroutine SYG

SUBROUTINE SYG (ICON)
DIMENSION WORD(132)
DATA ONEDOL /4H$
GO TO (1, 2), ICON
1 REWIND 8
RETURN
C TERMINATE THE FILE
2 WRITE (8, 10)
REWIND 8
C READ RECORD
3 READ (8, 11) (WORD(I), I=1, 132)
C CHECK FOR 12 LEADING DOLLAR SIGNS
DO 4 I = 1, 12
IF (WORD(I) .NE. ONEDOL) GO TO 5
4 CONTINUE
RETURN
C CHECK FOR 6 TRAILING DOLLAR SIGNS
5 DO 8 I = 1, 132
IF (WORD(I) .NE. ONEDOL) GO TO 8
K = I + 5
DO 7 J = I, K
IF (WORD(J) .NE. ONEDOL) GO TO 8
7 CONTINUE
GO TO 9
8 CONTINUE
WRITE (6, 12)
RETURN
C PRINT LINE
9 I = I - 1
WRITE (6, 11) (WORD(M), M=1, I)
GO TO 3
C
C
Subroutine THCOMP

SUBROUTINE THCOMP (PR, ETA, T, H, S, P, TO, HO, SO, PO)
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI
COMMON /COMALL/ COM(1062)
EQUIVALENCE (TOLALL, COM(23)), (SI, COM(1055))
CPG = .25DO
IF (SI) CPG = 1048.0DO
PO = P * PR
TP = T * PR ** 0.28572DO
DO 1 I = 1, 25
CALL THERMO (PO, HP, TP, SP, X1, 0, X2, 0)
DELS = SP - S
IF (DABS(DELS) .LE. .05DO * TOLALL * S) GO TO 2
1 TP = TP / DEXP(DELS / CPG)
CALL ERROR
2 HO = H + ((HP - H) / ETA)
CALL THERMO (PO, HO, TO, SO, X1, 0, X2, 1)
RETURN
END

Subroutine THERMO

SUBROUTINE THERMO (PX, HX, TX, SX, AMX, I, FAR, K)
IMPLICIT REAL*8 (A-H,O-Z)
LOGICAL SI
COMMON /COMALL/ COM(1062)
EQUIVALENCE (TOLALL, COM(23)), (SI, COM(1055))
IF (SI) GO TO 100
DEM = 1.986375D0
CPG = .25DO
PSTD = 1.0DO
GO TO 101
100 DEM = 8316.41D0
CPG = 1048.0DO
PSTD = 10.1325.0DO
GO TO 101
101 FX = 0.0DO
IF (L .EQ. 1) FX = FAR
IF (K .EQ. 1) GO TO 1
CALL PROCOM (FX, TX, CS, AK, CP, R, PHI, HX)
GO TO 3
1 TX = HX / CPG
DO 2 I = 1, 50
CALL PROCOM (FX, TX, CS, AK, CP, R, PHI, HX)
DELS = HX - H
IF (DABS(DELS) .LE. .01DO * TOLALL * HX) GO TO 3
2 TX = TX + DELH / CPG
WRITE (8, 4)
\begin{verbatim}
SUBROUTINE THTURB (DH, ETA, FAR, H, S, P, TO, HO, SO, PO)
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL SI
COMMON /COMALL/ COM(1062)
EQUIVALENCE (TOLALL, COM(23)), (SI, COM(1055))
DEM = 1.986375D0
IF (SI) DEM = 8316.41D0
HO = H - DH
HOP = H - DH / ETA
PT = P / 2.0D0
DO 1 I = 1, 25
CALL THERMO (PT, HOP, TT, ST, AMWT, 1, FAR, 1)
DELS = ST - S
IF (DABS (DELS) .LE. .05D0 * TOLALL * S) GO TO 2
CALL ERROR
CALL THERMO (PO, HO, TO, SO, X1, 1, FAR, 1)
RETURN
END

SUBROUTINE WDUCTI
IMPLICIT REAL*8 (A-H, O-Z)
LOGICAL SI
COMMON /COMALL/ COM(1062)
DIMENSION WORD (2), ERR (9)
DIMENSION Q(9), XZERO(25), A'R'DRD(2)
EQUIVALENCE (WORD (1), COM(1)), (IDES, COM(3)), (IDSHJC, COM(15))
1 (TOLALL, COM(23)), (PPR (1), COM(24)), (P1, COM(33)), (WAC, 
2 COM(191)), (WG37, COM(210)), (XZERO(1), COM(210)), (A38, 
3 COM(211)), (A38, COM(212)), (V38, COM(213)), (T38, COM(214)),
4 (U38, COM(215)), (P38, COM(216)), (TS38, COM(217)),
5 (PS38, COM(218)), (A39, COM(219)), (AM39, COM(220)),
6 (V39, COM(221)), (T39, COM(222)), (H39, COM(223)),
7 (P39, COM(224)), (TS39, COM(225)), (PS39, COM(226)),
8 (WA32DS, COM(227)), (DPWNG, COM(228)), (BPRINT, COM(229)),
9 (DPWGDS, COM(235)), (T21, COM(263)), (H21, COM(264)),
1 (WA32, COM(271)), (FCL3LD, COM(305)), (P21, COM(377)),
2 (P37, COM(393)), (U37, COM(394)), (VWDUCT, COM(404)),
3 (SI, COM(1055))
DATA AWORD /4HWDUC, 4HTI /
WORD(1) = AWORD(1)
\end{verbatim}
WORD(2) = AWORD(2)
IF (SI)   GO TO 100
RA = .0252D0
AJ = 2.719D0
GO TO 101
100 RA = 286.9D0
AJ = 1.0D0
101 IF (PCBLID .GT. 0.0D0) GO TO 3
DO 1 I = 1,25
1 XZERO(I) = C.0D0
RETURN
3 P32 = P21
H32 = H21
T32 = T21
BPRINT = WA32 / WAC
WA32C = WA32 * DSQRT(T32) / P32
IF (IDES .EQ. 1) WA32DS = WA32C
DPWING = DPWGDS / WA32C / WA32DS
DPWING = DMIN1(1.0D0,DPWING)
P36 = P32 * (1.0D0 - DPWING)
T36 = T32
H36 = H32
CALL THERMO (P36,H36,T36,S36,XX2,1,0.0D0,0)
WG37 = WA32
T37 = T36
P37 = P36
H37 = H36
S37 = S36
IF (VWDUCT .EQ. 0.0D0) GO TO 21
Q(2) = 0.0D0
Q(3) = 0.0D0
WG37P = WG37
H37P = H37
P37DOT = DERIV(22,P37)
18 CALL THERMO (P37,H37,T37,S37,XX2,1,0.0D0,0)
WG37 = WG37P - P37DOT * VWDUCT / T37 / 1.4D0 / RA
U37 = H37 - RA * AJ * T37
U37DOT = DERIV(23,U37)
H37X = (WG37P * H37P - (WG37P - WG37) * U37 - U37DOT * P37 *
1 VWDUCT / T37 / RA) / WG37
ERRW = (H37 - H37X) / H37
DIR = DSQRT(DABS(H37 / H37X))
CALL AFQUIR (Q(1),T37,ERRW,0.0D0,20.0D0,.1D0*TOLALL,DIR,T37T,IGO)
GO TO (19,21,20), IGO
19 T37 = T37T
GO TO 18
20 CALL ERROR
21 CALL CONVRG (T37,H37,P37,S37,0.0D0,WG37,P1,IDES,A38,P38R,T38,H38,
1 P38,S38,TS38,PS38,V38,AM38,ICON)
GO TO (5,5,5,4), ICON
4 CALL ERROR
5 T39 = T38
H39 = H38
P39 = P38
ABCDA931
ABCDA932
ABCDA933
ABCDA934
ABCDA935
ABCDA936
ABCDA937
ABCDA938
ABCDA939
ABCDA940
ABCDA941
ABCDA942
ABCDA943
ABCDA944
SUBROUTINE ZERO
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /COMALL/ COM(1062)
EQUIVALENCE (IDES, COM(3)), (JDES, COM(4)), (INIT, COM(7)),
1 (IDBURN, COM(11)), (IAFTBN, COM(12)), (IDSHOC, COM(15)),
2 (IPISHOC, COM(16)), (NOZPLT, COM(17))
IDES = 0
JDES = 0
INIT = 0
IDBURN = 0
IAFTBN = 0
IDSHOC = 3
IMSHOC = 3
NOZPLT = 0
COM(33) = 0.0D0
DO 5 I = 37,90
IF (I .EQ. 41 .OR. I .EQ. 50 .OR. I .EQ. 51) GO TO 5
IF (I .EQ. 60 .OR. I .EQ. 61 .OR. I .EQ. 75) GO TO 5
COM(I) = 0.0D0
5 CONTINUE
COM(94) = 0.0D0
COM(95) = 0.0D0
COM(99) = 0.0D0
DO 6 I = 102,113
IF (I .EQ. 107 .OR. I .EQ. 109 .OR. I .EQ. 110) GO TO 6
COM(I) = 0.0D0
6 CONTINUE
DO 7 I = 131,135
IF (I .EQ. 133) GO TO 7
COM(I) = 0.0D0
7 CONTINUE
DO 8 I = 149,160
IF (I .EQ. 155 .OR. I .EQ. 156) GO TO 8
COM(I) = 0.0D0
8 CONTINUE
DO 9 I = 171,174
COM(I) = 0.0D0
9 CONTINUE
DO 10 I = 190,209
10 CONTINUE
DO 11 I = 242,258
IF (I .EQ. 244) GO TO 11
COM(I) = 0.0D0
11 CONTINUE
DO 12 I = 263,265
COM(I) = 0.0D0
DO 13 I = 270,274
IF (I .EQ. 271) GO TO 13
COM(I) = 0.0D0
13 CONTINUE
DO 14 I = 290,293
COM(I) = 0.0D0
DO 15 I = 306,320
IF ((I .GE. 309 .AND. I .LE. 312) .OR. I .EQ. 314) GO TO 15
COM(I) = 0.0D0
15 CONTINUE
DO 16 I = 321,355
IF (I .EQ. 325 .OR. I .EQ. 326 .OR. I .EQ. 328) GO TO 16
IF (I .EQ. 329 .OR. I .EQ. 336 .OR. I .EQ. 341) GO TO 16
IF (I .EQ. 342 .OR. I .EQ. 346 .OR. I .EQ. 347) GO TO 16
COM(I) = 0.0D0
16 CONTINUE
DO 17 I = 358,371
IF (I .EQ. 366 .OR. I .EQ. 367 .OR. I .EQ. 370) GO TO 17
COM(I) = 0.0D0
17 CONTINUE
CALL SYG (1)
RETURN
END
APPENDIX C

SAMPLE CASE INPUT AND OUTPUT

PCNF
CF
ZF
PRF
WAPC
WAP
PCNC
CNC
ZC
PRC
WACC
WAC
T2
P2
T21
P21
T3
P3
PCBLP
BLF
PCBLC
BLC
PCBLOB
BLOB
PCBLLP
BLHP
PCBLLP
BLLP
T4
P4
WA3
WFB
WG4
FAR4
ETAB
DPCOM
TPFHP
CNHP
DHTCHP
DHTC
T5
P5
TPFLP
CNLP
DHTCLP
DHTF
T55
P55
PCBLDU
BLDU
T24
P24
T25
P25
SDATAIN ISPOOL=2, FAN=.TRUE., SI=.FALSE., IDES=1, MODE=0, I DUMP=1, IAMTP=0, IGASMX=2, ITRYS=200, PXFM2M=.FALSE., PXM2CP=.FALSE., APTFAN=.FALSE., DUNSPL=.TRUE., TOLALL=.001, DELPG=.0, DELFN=.1, DELSFC=.0, PCNFDS=102.31, PRFDS=2.996, ETAFDS=.8499, PCNCDS=.9873, PFCD=8.462, ETAED=81.36, TUSD=2892.04, ETAARDS=1.00, DPEDS=.0561, ETipDS=.8713, ELPDS=.9021, DPEDU=1.058, T7DS=3583.6,
ETAADS=.8430, DPAFDS=.0599, AM55=.283, AM6=.243, CVMN0Z=.9494, WAPCDS=221.573,
WACCDS=54.988, HPEXT=0.0, AM=0.0, ALTP=0.0, PCBLF=0.0, PCBLC=.16, PCBLDU=.208,
PCBLOB=0.0, PCBLLF=.726, PCBLLL=.966, AM23=.170, ZPDS=.8333, ZCDS=.8143,
TPHPDS=50.0, CNHPDS=2.0, TPLPDS=130.0, CNLPDS=2.3, XNHPS=10070.0, XNLPS=9651.0,
PMHP=3.30, PMILP=4.50, VFAN=2.31, VCOMP=1.65, VCOMB=1.65, VMHPTRB=.505,
VLPTRB=.518, VAPTBN=49.77, VFDUCT=10.08  &END
SDATAIN MODE=2, WFB=2.75, ITRAN=0, IDES=0  &END
SDATAIN MODE=2, WFB=2.75, ITRAN=1, IAMTRX=1  &END
<table>
<thead>
<tr>
<th>Design Type</th>
<th>Design Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan Design</td>
<td>( \Phi_{NPCF} = 3.47523476 \times 10^3 ), ( \Phi_{WPFC} = 2.35527831 \times 10^3 ), ( T_{20} = 2.51865928 \times 10^2 )</td>
</tr>
<tr>
<td>Middle Spool Design</td>
<td>( \Phi_{NPCF} = 3.10000000 \times 10^2 ), ( \Phi_{WPFC} = 2.35527831 \times 10^2 ), ( T_{22} = 3.74236432 \times 10^2 )</td>
</tr>
<tr>
<td>Compressor Design</td>
<td>( \Phi_{NPCF} = 1.09221258 \times 10^2 ), ( \Phi_{WPFC} = 2.35527831 \times 10^2 ), ( T_{21} = 3.74236432 \times 10^2 )</td>
</tr>
<tr>
<td>Combustor Design</td>
<td>( \Phi_{NPCF} = 3.47523476 \times 10^3 ), ( \Phi_{WPFC} = 2.35527831 \times 10^3 ), ( T_{20} = 2.51865928 \times 10^3 )</td>
</tr>
<tr>
<td>High Pressure Turbine Design</td>
<td>( \Phi_{NPCF} = 3.47523476 \times 10^3 ), ( \Phi_{WPFC} = 2.35527831 \times 10^3 ), ( T_{22} = 3.74236432 \times 10^3 )</td>
</tr>
<tr>
<td>Low Pressure Turbine Design</td>
<td>( \Phi_{NPCF} = 1.09221258 \times 10^2 ), ( \Phi_{WPFC} = 2.35527831 \times 10^2 ), ( T_{21} = 3.74236432 \times 10^2 )</td>
</tr>
<tr>
<td>Turbine/Duct Area Design</td>
<td>( \Phi_{NPCF} = 3.47523476 \times 10^3 ), ( \Phi_{WPFC} = 2.35527831 \times 10^3 ), ( T_{20} = 2.51865928 \times 10^3 )</td>
</tr>
<tr>
<td>Afterburner Design</td>
<td>( \Phi_{NPCF} = 3.47523476 \times 10^3 ), ( \Phi_{WPFC} = 2.35527831 \times 10^3 ), ( T_{20} = 2.51865928 \times 10^3 )</td>
</tr>
</tbody>
</table>

**Middle Spool Is Dummy**

No airflow into wing. The output is in English units.
### Table

<table>
<thead>
<tr>
<th>T7</th>
<th>WPA</th>
<th>WGT</th>
<th>PAB7</th>
<th>ETA</th>
<th>DPAPFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.141349E 06</td>
<td>0.000000</td>
<td>0.224322E 03</td>
<td>0.124102E-01</td>
<td>0.000100</td>
<td>0.599303E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PS2</th>
<th>AM8</th>
<th>V9</th>
<th>PS9</th>
<th>AM8</th>
<th>V9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.137620E 01</td>
<td>0.000000</td>
<td>0.167416E 04</td>
<td>0.137620E 01</td>
<td>0.000000</td>
<td>0.157115E 34</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PS2B</th>
<th>AM8</th>
<th>V2B</th>
<th>PS29</th>
<th>AM8</th>
<th>V29</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BYPASS</th>
<th>V9EXT</th>
<th>WPT</th>
<th>WST</th>
<th>VA</th>
<th>FBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.609702E 00</td>
<td>0.000000</td>
<td>0.274975E 01</td>
<td>0.224322E 03</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CVMD2</th>
<th>VJD</th>
<th>PGM</th>
<th>FIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.994400E 00</td>
<td>0.158944E 04</td>
<td>0.000000</td>
<td>0.119319E 05</td>
</tr>
</tbody>
</table>

### Main Sonic Convergent Nozzle

- PG = 13429.44
- FN = 13429.44
- SFC = 3.73742

**Converged After 1 Loops**
### OUTPUT

<table>
<thead>
<tr>
<th>AM</th>
<th>ALT</th>
<th>T4</th>
<th>ETAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
<td>2892.17</td>
<td>1.000</td>
</tr>
</tbody>
</table>

**MIDDLE SPOOL IS DUMMY**

**NO AIRFLOW INTO WING**

**THE OUTPUT IS IN ENGLISH UNITS**

<table>
<thead>
<tr>
<th>PCNP</th>
<th>CNF</th>
<th>ZP</th>
<th>PBF</th>
<th>WAF</th>
<th>WAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.102310E 03</td>
<td>0.102310E 01</td>
<td>0.933378E 00</td>
<td>0.299619E 01</td>
<td>0.221553E 03</td>
<td>0.221553E 03</td>
</tr>
<tr>
<td>PCNC</td>
<td>CN</td>
<td>ZC</td>
<td>PRC</td>
<td>WAC</td>
<td>WAC</td>
</tr>
<tr>
<td>0.139165E 03</td>
<td>0.397289E 00</td>
<td>0.914304E 00</td>
<td>0.946167E 01</td>
<td>0.549979E 02</td>
<td>0.137553E 03</td>
</tr>
<tr>
<td>T2</td>
<td>P2</td>
<td>T21</td>
<td>P21</td>
<td>T3</td>
<td>T3</td>
</tr>
<tr>
<td>0.5118270E 03</td>
<td>3.100000E 01</td>
<td>3.742991E 03</td>
<td>0.299619E 01</td>
<td>0.145756E 04</td>
<td>0.253531E 02</td>
</tr>
<tr>
<td>PCBLP</td>
<td>BLP</td>
<td>PCBLP</td>
<td>BLC</td>
<td>PCBLP</td>
<td>BLP</td>
</tr>
<tr>
<td>0.00003000</td>
<td>0.000000</td>
<td>0.160000E 00</td>
<td>0.222424E 02</td>
<td>0.100000E 02</td>
<td>0.000000</td>
</tr>
<tr>
<td>PCBLHP</td>
<td>BLHP</td>
<td>PCBLLP</td>
<td>BLP</td>
<td>T4</td>
<td>P4</td>
</tr>
<tr>
<td>0.725300E 00</td>
<td>0.719897E 02</td>
<td>0.650000E 01</td>
<td>0.125316E 01</td>
<td>0.249213E 04</td>
<td>0.239313E 02</td>
</tr>
<tr>
<td>WAD</td>
<td>WPD</td>
<td>WSC</td>
<td>FAP4</td>
<td>ETA4</td>
<td>DPCOM</td>
</tr>
<tr>
<td>0.115628E 03</td>
<td>0.275000E 01</td>
<td>0.118378E 03</td>
<td>0.237931E 01</td>
<td>0.100000E 01</td>
<td>0.560995E 01</td>
</tr>
<tr>
<td>TFFHP</td>
<td>CHP</td>
<td>DHTCHP</td>
<td>DHTC</td>
<td>T5</td>
<td>T5</td>
</tr>
<tr>
<td>0.500000E 01</td>
<td>0.199997E 02</td>
<td>0.734133E 01</td>
<td>0.212365E 03</td>
<td>0.211358E 04</td>
<td>0.566689E 01</td>
</tr>
<tr>
<td>TFFLP</td>
<td>CNLP</td>
<td>DHTCLP</td>
<td>DHTF</td>
<td>T55</td>
<td>T56</td>
</tr>
<tr>
<td>0.130000E 03</td>
<td>3.229996E 01</td>
<td>0.422981E 01</td>
<td>0.895316E 02</td>
<td>0.174131E 03</td>
<td>0.254131E 01</td>
</tr>
<tr>
<td>PCBLDU</td>
<td>PBLDU</td>
<td>T24</td>
<td>P24</td>
<td>T25</td>
<td>P25</td>
</tr>
<tr>
<td>0.209300E 00</td>
<td>0.719897E 03</td>
<td>0.282123E 03</td>
<td>0.781999E 03</td>
<td>0.292123E 01</td>
<td></td>
</tr>
<tr>
<td>WAD</td>
<td>WPD</td>
<td>WSC</td>
<td>FAP4</td>
<td>ETA4</td>
<td>DPCOM</td>
</tr>
<tr>
<td>0.894967E 02</td>
<td>0.000000</td>
<td>0.584957E 02</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.583915E 01</td>
</tr>
<tr>
<td>ETA4</td>
<td>ETAC</td>
<td>ETAHP</td>
<td>ETA4LP</td>
<td>AM55</td>
<td>AM55</td>
</tr>
<tr>
<td>0.149999E 00</td>
<td>0.813597E 00</td>
<td>0.817123E 00</td>
<td>0.321000E 00</td>
<td>0.283237E 00</td>
<td>0.383356E 00</td>
</tr>
<tr>
<td>T6</td>
<td>P6</td>
<td>PS6</td>
<td>AM6</td>
<td>W6</td>
<td>W6</td>
</tr>
<tr>
<td>0.141358E 04</td>
<td>0.271363E 01</td>
<td>0.250914E 01</td>
<td>0.243014E 00</td>
<td>0.437924E 03</td>
<td>0.293189E 03</td>
</tr>
<tr>
<td>T7</td>
<td>WPA</td>
<td>WGC</td>
<td>PAP7</td>
<td>ETA4</td>
<td>DPA4P</td>
</tr>
<tr>
<td>0.141358E 04</td>
<td>0.000000</td>
<td>0.224319E 00</td>
<td>0.124115E 03</td>
<td>0.703900</td>
<td>0.599099E 01</td>
</tr>
<tr>
<td>PS9</td>
<td>AM9</td>
<td>V9</td>
<td>PS9</td>
<td>AM9</td>
<td>V9</td>
</tr>
<tr>
<td>0.137622E 01</td>
<td>0.100000E 01</td>
<td>0.157421E 04</td>
<td>0.137622E 01</td>
<td>0.100000E 01</td>
<td>0.157421E 04</td>
</tr>
<tr>
<td>PS2</td>
<td>AM2</td>
<td>V2</td>
<td>PS2</td>
<td>AM2</td>
<td>V2</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.150000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>PYP4SS</td>
<td>PPEXT</td>
<td>WPT</td>
<td>WGT</td>
<td>ETA6</td>
<td>DPA4P</td>
</tr>
<tr>
<td>0.509619E 00</td>
<td>0.000000</td>
<td>0.225500E 01</td>
<td>0.224319E 00</td>
<td>0.300000</td>
<td>0.300000</td>
</tr>
<tr>
<td>CVEN</td>
<td>DJN</td>
<td>CVN</td>
<td>VJD</td>
<td>PGM</td>
<td>PPD</td>
</tr>
<tr>
<td>0.993400E 00</td>
<td>0.156949E 00</td>
<td>0.300000</td>
<td>0.103220E 05</td>
<td>0.214772E 04</td>
<td></td>
</tr>
</tbody>
</table>

**MAIN SONIC CONVERGENT NOZZLE**

| FDP | 13429.68 | FPN | 13429.53 | SPF | 1.73717 |

**CONVERGED AFTER 44 LOOPS**
<table>
<thead>
<tr>
<th>THIS IS THE S.S. SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1399E 04 9.8740E 03 1.3374E 02 2.5511E 00 2.5380E 02 2.53538 01 5.8599E 12 2.3212E 00</td>
</tr>
<tr>
<td>1.1999E 04 9.8740E 03 1.3374E 02 2.5511E 00 2.5380E 02 2.5994E 02 1.3374E 01 2.5511E 00</td>
</tr>
<tr>
<td>THIS VARIABLE IS NO. 1</td>
</tr>
<tr>
<td>THIS COLUMN OF X =</td>
</tr>
<tr>
<td>1.1857E 01 1.3973E 01 2.2132E 00 1.2314E 03 2.7948E 01 1.3374E 01 2.5994E 02 1.3374E 01</td>
</tr>
<tr>
<td>ISIN = 1</td>
</tr>
<tr>
<td>THE NEXT TRY FOR PVQDOT = 2.1000E-02</td>
</tr>
<tr>
<td>1 THE VARIABLE IS NO. 1</td>
</tr>
<tr>
<td>THIS COLUMN OF X =</td>
</tr>
<tr>
<td>1.1857E 01 1.3973E 01 2.2132E 00 1.2314E 03 2.7948E 01 1.3374E 01 2.5994E 02 1.3374E 01</td>
</tr>
<tr>
<td>ISIN = 1</td>
</tr>
<tr>
<td>THE NEXT TRY FOR PVQDOT = 6.4553E-03</td>
</tr>
<tr>
<td>1 THE VARIABLE IS NO. 1</td>
</tr>
<tr>
<td>THIS COLUMN OF X =</td>
</tr>
<tr>
<td>1.1857E 01 1.3973E 01 2.2132E 00 1.2314E 03 2.7948E 01 1.3374E 01 2.5994E 02 1.3374E 01</td>
</tr>
<tr>
<td>ISIN = 0</td>
</tr>
<tr>
<td>THE NEXT TRY FOR PVQDOT = 6.5346E-03</td>
</tr>
<tr>
<td>2 THE VARIABLE IS NO. 3</td>
</tr>
<tr>
<td>THIS COLUMN OF X =</td>
</tr>
<tr>
<td>1.8732E 01 1.9736E 01 1.3139E 01 1.3463E 01 1.4334E 01 1.3973E 01 2.0512E 01 1.3973E 01</td>
</tr>
<tr>
<td>ISIN = 1</td>
</tr>
<tr>
<td>THE NEXT TRY FOR PVQDOT = 2.1000E-02</td>
</tr>
<tr>
<td>1 THE VARIABLE IS NO. 3</td>
</tr>
<tr>
<td>THIS COLUMN OF X =</td>
</tr>
<tr>
<td>1.8732E 01 1.9736E 01 1.3139E 01 1.3463E 01 1.4334E 01 1.3973E 01 2.0512E 01 1.3973E 01</td>
</tr>
<tr>
<td>ISIN = 1</td>
</tr>
<tr>
<td>THE NEXT TRY FOR PVQDOT = 2.1000E-02</td>
</tr>
<tr>
<td>2 THE VARIABLE IS NO. 3</td>
</tr>
<tr>
<td>THIS COLUMN OF X =</td>
</tr>
<tr>
<td>1.8732E 01 1.9736E 01 1.3139E 01 1.3463E 01 1.4334E 01 1.3973E 01 2.0512E 01 1.3973E 01</td>
</tr>
<tr>
<td>ISIN = 1</td>
</tr>
<tr>
<td>THE NEXT TRY FOR PVQDOT = 2.1000E-02</td>
</tr>
<tr>
<td>1 THE VARIABLE IS NO. 3</td>
</tr>
<tr>
<td>THIS COLUMN OF X =</td>
</tr>
<tr>
<td>1.8732E 01 1.9736E 01 1.3139E 01 1.3463E 01 1.4334E 01 1.3973E 01 2.0512E 01 1.3973E 01</td>
</tr>
<tr>
<td>ISIN = 0</td>
</tr>
</tbody>
</table>
3 THE VARIABLE IS NO. 21

<table>
<thead>
<tr>
<th>COLUMN OF X</th>
<th>0.0000</th>
<th>-1.8190E-12</th>
<th>-3.1338E-02</th>
<th>-3.0195E-02</th>
<th>-1.0558E-14</th>
<th>5.5943E-14</th>
<th>-8.3919E-15</th>
<th>-3.1974E-14</th>
</tr>
</thead>
</table>

IGIN = 1

THE NEXT TRY FOR PVRDOT = 3.0010E-02
3 THE VARIABLE IS NO. 21

<table>
<thead>
<tr>
<th>COLUMN OF X</th>
<th>0.0000</th>
<th>-1.8190E-12</th>
<th>-4.7036E-02</th>
<th>-4.5319E-02</th>
<th>-1.0558E-14</th>
<th>5.5943E-14</th>
<th>-8.3919E-15</th>
<th>-3.1974E-14</th>
</tr>
</thead>
</table>

IGIN = 2

THE NEXT TRY FOR PVRDOT = 6.0020E-32
3 THE VARIABLE IS NO. 21

<table>
<thead>
<tr>
<th>COLUMN OF X</th>
<th>3.1549E-03</th>
<th>-7.9375E-03</th>
<th>-6.610E-03</th>
<th>-2.0000E-03</th>
<th>-4.0000E-03</th>
<th>-6.0000E-03</th>
<th>-8.0000E-03</th>
<th>-1.0000E-03</th>
</tr>
</thead>
</table>

IGIN = 5

THE NEXT TRY FOR PVRED = 1.2004E-01
3 THE VARIABLE IS NO. 21

|-------------|------------|-------------|------------|------------|-------------|------------|------------|------------|

IGIN = 0

THE NEXT TRY FOR PVRED = 5.6049E-10
3 THE VARIABLE IS NO. 21

|-------------|------------|-------------|------------|------------|----------|-----------|------------|------------|

IGIN = 0

THE NEXT TRY FOR PVRED = 2.0000E-02
4 THE VARIABLE IS NO. 18

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0000E-02</td>
<td>2.9749E-11</td>
<td>2.0000E-02</td>
<td>2.0000E-02</td>
<td>2.0000E-02</td>
<td>2.0000E-02</td>
<td>2.0000E-02</td>
<td>2.0000E-02</td>
<td>2.0000E-02</td>
</tr>
<tr>
<td>Column</td>
<td>Values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-6.0803 × 10^{-3} -9.3137 × 10^{-3} -6.4744 × 10^{-5} +0.5996 × 10^{-4} -5.9319 × 10^{-5} -1.5342 × 10^{-4} 2.9764 × 10^{-1} 3.3245 × 10^{-1} -7.7204 × 10^{-4} -1.5171 × 10^{-4} -7.1944 × 10^{-2} 2.0837 × 10^{-3}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-2.2128 × 10^{-2} -0.9459 × 10^{-2} -8.2344 × 10^{-4} 2.9540 × 10^{-3} -5.6105 × 10^{-1} -1.3536 × 10^{-3} 2.0673 × 10^{0} 2.0963 × 10^{0} -3.3099 × 10^{-3} -1.2482 × 10^{-3} -5.0414 × 10^{-1} 1.9802 × 10^{-2}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-7.2034 × 10^{-2} -6.7192 × 10^{-2} 1.1057 × 10^{-3} 5.0756 × 10^{-5} 1.3121 × 10^{-4} 3.0843 × 10^{-4}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-2.2217 × 10^{-2} -4.9741 × 10^{-2} -8.2946 × 10^{-4} 1.9523 × 10^{-3} -5.6526 × 10^{-4} -1.3631 × 10^{-3}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-7.5936 × 10^{-2} -7.0390 × 10^{-2} 1.1584 × 10^{-3} 5.0756 × 10^{-5} 1.3121 × 10^{-4} 3.0843 × 10^{-4}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-7.5376 × 10^{-2} -7.0390 × 10^{-2} 1.1584 × 10^{-3} 5.0756 × 10^{-5} 1.3121 × 10^{-4} 3.0843 × 10^{-4}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column of X</td>
<td>ISIN</td>
<td>The Next Try for PV ddot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>-------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.4957E-01</td>
<td>9</td>
<td>7.0137E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0298E-01</td>
<td>5</td>
<td>6.5376E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9766E-01</td>
<td>0</td>
<td>1.7537E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.3094E-03</td>
<td>1</td>
<td>-3.8958E-00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2.4177E-00</td>
<td>2</td>
<td>8.1555E-03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5173E-02</td>
<td>3</td>
<td>1.6364E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5373E-01</td>
<td>4</td>
<td>3.8307E-03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5490E-01</td>
<td>0</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>2.2441E-01</td>
<td>1</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>2.0392E-01</td>
<td>2</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>5.3177E-03</td>
<td>3</td>
<td>1.9766E-01</td>
</tr>
<tr>
<td>-2.4177E-00</td>
<td>4</td>
<td>8.1555E-03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7439E-02</td>
<td>1</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>4.5692E-02</td>
<td>2</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>4.7439E-02</td>
<td>3</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>4.5692E-02</td>
<td>4</td>
<td>1.9766E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5012E-01</td>
<td>1</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>6.3555E-02</td>
<td>2</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>1.1190E-00</td>
<td>3</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>6.3555E-02</td>
<td>4</td>
<td>1.9766E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9915E-02</td>
<td>1</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>4.8059E-02</td>
<td>2</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>4.9915E-02</td>
<td>3</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>4.8059E-02</td>
<td>4</td>
<td>1.9766E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9258E-01</td>
<td>1</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>3.1296E-01</td>
<td>2</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>3.1296E-01</td>
<td>3</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>3.1296E-01</td>
<td>4</td>
<td>1.9766E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8568E-01</td>
<td>1</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>2</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>3</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>4</td>
<td>1.9766E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8568E-01</td>
<td>1</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>2</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>3</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>4</td>
<td>1.9766E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column of X</th>
<th>ISIN</th>
<th>The Next Try for PV ddot</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8568E-01</td>
<td>1</td>
<td>7.0137E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>2</td>
<td>6.5490E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>3</td>
<td>2.2088E-01</td>
</tr>
<tr>
<td>1.8568E-01</td>
<td>4</td>
<td>1.9766E-01</td>
</tr>
</tbody>
</table>
THE VARIABLE IS NO. 11

THIS COLUMN OF X =
-1.9214E 01 -1.4498E 01 -1.9529E 01 -3.2118E-03 -5.8079E-01 -4.2135E-01 -1.1405E-01
-1.335E-01 -5.3335E-01 -2.3360E-02 -1.0775E-01 -7.8516E-03 -9.1385E-03

IGIN = 3

THE NEXT TRY FOR PVRDOT = 9.6865E-01

THE VARIABLE IS NO. 11

THIS COLUMN OF X =
-2.0750E 01 -1.5653E 01 -2.1079E 01 -5.3335E-01 -2.3360E-02 -1.1635E-01
-8.4824E-03 -9.8565E-03

IGIN = 0

THE NEXT TRY FOR PVRDOT = 9.6455E-01

THE VARIABLE IS NO. 20

THIS COLUMN OF X =

IGIN = 1

THE NEXT TRY FOR PVRDOT = 3.0010E-02

THE VARIABLE IS NO. 20

THIS COLUMN OF X =
0.0000 -1.8190E-12 8.9617E-03 9.5175E-06 1.0412E-02 -5.9343E-14 5.8817E-14
-9.3318E-15 3.137E-18

IGIN = 2

THE NEXT TRY FOR PVRDOT = 6.0320E-02

THE VARIABLE IS NO. 20

THIS COLUMN OF X =
5.4728E-03 6.0113E-01 2.3183E-02 -2.9048E-04 2.1800E-02 2.3942E-04
1.1102E-15 7.5995E-15

IGIN = 3

THE NEXT TRY FOR PVRDOT = 1.2865E 00

THE VARIABLE IS NO. 20

THIS COLUMN OF X =
1.1725E 01 9.8375E 00 4.7984E-01 -5.2937E-03 -5.9440E-01 -4.7602E-02
-7.9595E-03 2.1043E-01 -8.3829E-01 -4.7408E-02

IGIN = 4

THE NEXT TRY FOR PVRDOT = 8.1376E-01

THE VARIABLE IS NO. 20

THIS COLUMN OF X =
1.7218E-02 -2.9457E-02 -5.9257E-03 -7.3874E-03 -7.7876E-03 -5.9902E-03

IGIN = 5

THE NEXT TRY FOR PVRDOT = 8.2120E-01

THE VARIABLE IS NO. 20

THIS COLUMN OF X =
7.6608E 00 6.4536E 00 3.9678E-01 -1.1322E-03 -3.9983E-02 -4.3099E-03
1.2239E-01 -5.5419E-03 -2.8775E-02

IGIN = 0

THE NEXT TRY FOR PVRDOT = 8.2120E-01
<table>
<thead>
<tr>
<th>ISIN</th>
<th>PVRDOT</th>
<th>THIS COLUMN OF X</th>
<th>THE VARIABLE IS NO.</th>
<th>ISIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.0010E-02</td>
<td>0.0000 -1.8190E-12 -2.6038E-03 1.4415E-04 -3.7263E-04 -5.1384E-03 5.0135E-04 -1.6247E-02 9.3429E-06 -3.0326E-03</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3.8510E-02</td>
<td>-2.7209E-01 2.8711E-01 -2.6378E-03 1.1297E-04 -6.6521E-04 -5.6752E-03 2.4645E-04 -1.0459E-02 1.3156E-03 -2.3190E-03</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2.8510E-02</td>
<td>-5.6660E-03 -3.8532E-03 2.1629E-04 -1.1102E-15 1.8515E-04</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>RUN</td>
<td>THE VARIABLE IS NO. 13</td>
<td>THE NEXT TRY FOR PVRDOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------------------------</td>
<td>-------------------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THIS COLUMN OF X =</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>THE VARIABLE IS NO. 13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0000  -1.8190E-12</td>
<td>-3.9080E-14 -5.3562E-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8.315E-03 -5.5843E-14</td>
<td>1.0558E-14 5.5833E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.0558E-14 -1.0558E-14</td>
<td>-1.102E-15 7.223E-15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISIN</td>
<td>THE NEXT TRY FOR PVRDOT</td>
<td>THE VARIABLE IS NO. 17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------------------</td>
<td>------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.2294E-31</td>
<td>THIS COLUMN OF X = 0.0000 -1.8190E-12 -3.9080E-14 1.0397E-06 -2.2858E-03 -5.5843E-14 5.5833E-14 -8.3818E-16 -3.1974E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.4008E-01</td>
<td>THIS COLUMN OF X = 0.0000 -1.8190E-12 -3.9080E-14 1.0397E-06 -2.2858E-03 -5.5843E-14 5.5833E-14 -8.3818E-16 -3.1974E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4.8316E-31</td>
<td>THIS COLUMN OF X = 0.0000 -1.8190E-12 -3.9080E-14 1.0397E-06 -2.2858E-03 -5.5843E-14 5.5833E-14 -8.3818E-16 -3.1974E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9.6032E-01</td>
<td>THIS COLUMN OF X = 0.0000 -1.8190E-12 -3.9080E-14 1.0397E-06 -2.2858E-03 -5.5843E-14 5.5833E-14 -8.3818E-16 -3.1974E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4.9236E-30</td>
<td>THIS COLUMN OF X = 0.0000 -1.8190E-12 -3.9080E-14 1.0397E-06 -2.2858E-03 -5.5843E-14 5.5833E-14 -8.3818E-16 -3.1974E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8.2593E-00</td>
<td>THIS COLUMN OF X = 0.0000 -1.8190E-12 -3.9080E-14 1.0397E-06 -2.2858E-03 -5.5843E-14 5.5833E-14 -8.3818E-16 -3.1974E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8.9522E-00</td>
<td>THIS COLUMN OF X = 0.0000 -1.8190E-12 -3.9080E-14 1.0397E-06 -2.2858E-03 -5.5843E-14 5.5833E-14 -8.3818E-16 -3.1974E-14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td>Values</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>0.0300 -1.1890E-12 -3.9080E-14 -3.3678E-05 -5.0423E-04 -5.5843E-14 -1.0658E-14 5.6943E-14 -8.9818E-16 -3.1974E-14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 3.0110E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>0.0300 -1.1890E-12 -3.9080E-14 -1.3305E-05 -7.5660E-04 -5.6843E-14 -1.0658E-14 5.5943E-14 -8.3121E-16 -3.1374E-14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 6.0020E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>0.0000 -1.8190E-12 -3.9080E-14 -2.6512E-05 -1.5132E-03 -5.6843E-14 -1.0658E-14 5.5943E-14 -9.5819E-15 -3.1395E-14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 1.2274E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>0.0000 -1.8190E-12 -3.9080E-14 -5.3256E-05 -3.0265E-03 -5.6843E-14 -1.0658E-14 5.5943E-14 -9.3918E-15 -3.1374E-14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 2.400P8-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>3.2847E-01 1.7515E-02 -1.8378E-01 -1.7853E-04 -7.2608E-01 1.4806E-03 2.0191E-03 -2.9532E-02 -2.1343E-03 -3.3925E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 4.38E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>3.2847E-01 1.7515E-02 -1.8378E-01 -1.7853E-04 -7.2608E-01 1.4806E-03 2.0191E-03 -2.9532E-02 -2.1343E-03 -3.3925E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 8.9666E-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>3.2847E-01 1.7515E-02 -1.8378E-01 -1.7853E-04 -7.2608E-01 1.4806E-03 2.0191E-03 -2.9532E-02 -2.1343E-03 -3.3925E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 1.1739E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>The variable is No. 12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This column of X =</td>
<td>3.2847E-01 1.7515E-02 -1.8378E-01 -1.7853E-04 -7.2608E-01 1.4806E-03 2.0191E-03 -2.9532E-02 -2.1343E-03 -3.3925E-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGIN</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The next try for PVRDOT = 1.4289E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
13 THE VARIABLE IS NO. 12
13 THIS COLUMN OF X =
-4.03826E-01 -3.6915E-01 -1.1374E-02 5.0338E-02 -7.5181E-03 -6.2548E-03
IGIN = 0
13 THE NEXT TRY FOR PVRDOT = 1.4298E 01
14 THE VARIABLE IS NO. 10
14 THIS COLUMN OF X =
0.0000 -3.6380E-12 -4.1595E-03 8.2562E-04 9.1864E-03 5.2849E-03 1.0375E-03 2.1499E-03 -3.1199E-03
1.4298E 01
5.4629E-03 3.1605E-03 -5.0135E-04 -1.1116E-03 -9.9622E-04 -8.1112E-04
IGIN = 1
15 THE NEXT TRY FOR PVRDOT = 3.0310E-02
14 THE VARIABLE IS NO. 10
14 THIS COLUMN OF X =
0.0000 -3.6380E-12 -4.1595E-03 4.0059E-04 1.2569E-02 5.2849E-03 1.1735E-03 2.0335E-02 -2.1499E-03 -3.1199E-03
1.4298E 01
IGIN = 2
15 THE NEXT TRY FOR PVRDOT = 6.0020E-02
14 THE VARIABLE IS NO. 10
14 THIS COLUMN OF X =
-5.2617E-01 1.0839E-01 -1.1255E-02 6.5169E-02 3.8066E-03 2.7031E-02 -7.4835E-03 4.7507E-02 4.7522E-03 9.4907E-03
1.4298E 01
2.5057E-02 2.5627E-02 -1.6906E-02 7.2159E-02 -4.8537E-04 -6.3162E-04
IGIN = 3
15 THE NEXT TRY FOR PVRDOT = 2.5934E-01
14 THE VARIABLE IS NO. 10
14 THIS COLUMN OF X =
-3.4046E-00 -3.4046E-00 1.5466E-01 7.4266E-03 -3.4580E-02 -2.7733E-03 -2.5063E-03
1.5466E-01 1.5466E-01 7.4266E-03
IGIN = 4
15 THE NEXT TRY FOR PVRDOT = 3.5684E-01
14 THE VARIABLE IS NO. 10
14 THIS COLUMN OF X =
-4.7554E-00 4.7632E-00 -1.2546E-03 2.9223E-03 -1.0608E-01 4.9563E-02 3.1869E-01 3.1787E-03 5.8224E-02
1.4298E 01
2.1431E-01 2.1422E-01 1.0029E-01 4.7645E-02 3.1374E-03 3.5904E-03
IGIN = 0
15 THE NEXT TRY FOR PVRDOT = 3.5684E-01
14 THE VARIABLE IS NO. 10
14 THIS COLUMN OF X =
1.4298E 01
1.1092E-13 1.7053E-13 2.6645E-15 1.3558E-14 -1.1102E-15 7.5959E-15
IGIN = 1
15 THE NEXT TRY FOR PVRDOT = 3.0010E-02
15 THE VARIABLE IS NO. 4
15 THIS COLUMN OF X =
0.0000 -1.8190E-12 4.0510E-04 4.3128E-06 3.0962E-03 5.5843E-14 -1.0558E-14 3.5918E-14 1.2130E-15 3.1974E-14
1.4298E 01
1.7331E-13 1.7053E-13 2.6645E-15 1.3558E-14 -1.1102E-15 7.5959E-15
IGIN = 0
15 THE NEXT TRY FOR PVRDOT = 3.0010E-02
15 THE VARIABLE IS NO. 4
15 THIS COLUMN OF X =
0.0000 -1.8190E-12 1.7331E-03 -3.9092E-05 5.7380E-03 2.2212E-03 -2.1672E-04 2.2212E-03 -2.1672E-04
9.0495E-03
IGIN = 2
15 THE NEXT TRY FOR PVRDOT = 6.0020E-02
15 THE VARIABLE IS NO. 4

PHIS COLUMN OF X =
  0.0000 -1.8190E-12 3.4516E-03 -3.0159E-05 1.0778E-02 4.3333E-03 -8.3254E-04 1.5665E-02 2.1972E-05 2.6165E-03
IGIN = 3
THE NEXT TRY FOR PVRDOT = 1.2004E-01

15 THE VARIABLE IS NO. 4

THIS COLUMN OF X =
IGIN = 4
THE NEXT TRY FOR PVRDOT = 1.1332E-03

15 THE VARIABLE IS NO. 4

THIS COLUMN OF X =
  6.5595E-00 5.8194E-01 2.0431E-02 6.7292E-02 -8.7452E-03 1.7541E-01 -3.7133E-03 3.3237E-03
IGIN = 5
THE NEXT TRY FOR PVRDOT = 2.4309E-31

15 THE VARIABLE IS NO. 4

THIS COLUMN OF X =
IGIN = 0
THE NEXT TRY FOR PVRDOT = 3.0010E-02

16 THE VARIABLE IS NO. 16

THIS COLUMN OF X =
  0.0000 -1.8190E-12 -3.9080E-14 5.5080E-07 -1.8722E-09 -1.0658E-14 5.5843E-14 -8.3318E-15 -3.3178E-14
IGIN = 1
THE NEXT TRY FOR PVRDOT = 6.0320E-02

16 THE VARIABLE IS NO. 16

THIS COLUMN OF X =
  0.0000 -1.8190E-12 -3.9080E-14 9.2647E-07 -2.8093E-04 -5.6843E-14 -1.0658E-14 5.6843E-14 -8.3818E-16 -3.1974E-14
IGIN = 2
THE NEXT TRY FOR PVRDOT = 6.0320E-02

16 THE VARIABLE IS NO. 16

THIS COLUMN OF X =
  0.0000 -1.8190E-12 -3.9080E-14 1.6529E-06 -5.6165E-04 -5.5834E-14 -1.0658E-14 5.5834E-14 -8.3318E-15 -3.3178E-14
IGIN = 3
THE NEXT TRY FOR PVRDOT = 1.2004E-01

16 THE VARIABLE IS NO. 16

THIS COLUMN OF X =
  0.0000 -1.8190E-12 -3.9080E-14 3.3059E-06 -1.1237E-03 -5.6843E-14 -1.0658E-14 5.6843E-14 -8.3818E-16 -3.1974E-14
IGIN = 4
THE NEXT TRY FOR PVRDOT = 2.4309E-01
16  THE VARIABLE IS NO. 16

<table>
<thead>
<tr>
<th>Column</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
<th>Value 7</th>
<th>Value 8</th>
<th>Value 9</th>
</tr>
</thead>
</table>

This column of x =

\[
\begin{align*}
0.0000 & -1.01902E-12 & -3.3080E-14 & 5.31704E-06 & 4.8161E-04 & -5.684E-14 & -1.0658E-14 & 5.693E-14 & -3.1974E-14 \\
\end{align*}
\]

IGIN = 5

The next try for PVBDOT = \( 4.8016E-01 \)

16  THE VARIABLE IS NO. 16

This column of x =

\[
\begin{align*}
0.0000 & -1.01902E-12 & -3.3080E-14 & 5.31704E-06 & 4.8161E-04 & -5.684E-14 & -1.0658E-14 & 5.693E-14 & -3.1974E-14 \\
\end{align*}
\]

IGIN = 6

The next try for PVBDOT = \( 9.6032E-01 \)

16  THE VARIABLE IS NO. 16

This column of x =

\[
\begin{align*}
0.0000 & -1.01902E-12 & -3.3080E-14 & 5.31704E-06 & 4.8161E-04 & -5.684E-14 & -1.0658E-14 & 5.693E-14 & -3.1974E-14 \\
\end{align*}
\]

IGIN = 7

The next try for PVBDOT = \( 1.4345E 01 \)

16  THE VARIABLE IS NO. 16

This column of x =

\[
\begin{align*}
0.0000 & -1.01902E-12 & -3.3080E-14 & 5.31704E-06 & 4.8161E-04 & -5.684E-14 & -1.0658E-14 & 5.693E-14 & -3.1974E-14 \\
\end{align*}
\]

IGIN = 8

The next try for PVBDOT = \( 2.3795E 01 \)

16  THE VARIABLE IS NO. 16

This column of x =

\[
\begin{align*}
0.0000 & -1.01902E-12 & -3.3080E-14 & 5.31704E-06 & 4.8161E-04 & -5.684E-14 & -1.0658E-14 & 5.693E-14 & -3.1974E-14 \\
\end{align*}
\]

IGIN = 0

The next try for PVBDOT = \( 2.3785E 01 \)
<p>| TINV = | 9.7238E-04  | 1.2748E-02  | 3.5923E-01  | 5.4851E-03  | 2.0928E-01  | 1.7539E-03  |
|       | 1.2748E-02  | 9.7238E-04  | 1.2351E-02  | 1.1710E-01  | 1.8105E-01  | 1.5640E-02  |
|       | -3.3361E-01 | -1.2767E-01 | -1.3682E-03 | -4.9360E-05 | 1.6744E-03  | -2.6140E-03 |
|       | 7.0890E-05  | -2.3399E-03 | -5.2399E-05 | -1.9218E-03 | 2.4037E-02  | 5.2152E-03  |
|       | 2.4475E-02  | -1.2030E-02 | -1.3412E-05 | -2.6140E-03 | 1.2744E-02  | 4.4875E-02  |
|       | 1.0473E-04  | 4.4543E-06  | -1.5891E-05 | 1.2744E-02  | 9.7238E-04  | 1.2744E-02  |
|       | 9.8543E-00  | 1.0060E-01  | -1.4693E-04 | -2.7753E-03 | 5.7015E-02  | 5.7822E-02  |
|       | -8.8330E-02 | -2.5628E-01 | -4.0857E-05 | -4.0857E-05 | 5.7015E-02  | 5.7822E-02  |
|       | -1.3470E-02 | -1.2703E-02 | -1.9610E-05 | -1.9610E-05 | 5.7015E-02  | 5.7822E-02  |
|       | -1.3319E-01 | -9.9943E-02 | -2.1580E-05 | -2.1580E-05 | 5.7015E-02  | 5.7822E-02  |
|       | -1.3507E-03 | -1.1015E-03 | -3.7344E-03 | -3.7344E-03 | 5.7015E-02  | 5.7822E-02  |
|       | -3.6525E-02 | -2.7753E-02 | -3.7105E-04 | -3.7105E-04 | 5.7015E-02  | 5.7822E-02  |
|       | -2.5628E-01 | -1.4693E-01 | -5.1475E-05 | -5.1475E-05 | 5.7015E-02  | 5.7822E-02  |
|       | -1.3319E-01 | -9.9943E-02 | -2.1580E-05 | -2.1580E-05 | 5.7015E-02  | 5.7822E-02  |
|       | -3.6525E-02 | -2.7753E-02 | -3.7105E-04 | -3.7105E-04 | 5.7015E-02  | 5.7822E-02  |
|       | -1.3507E-03 | -1.1015E-03 | -3.7344E-03 | -3.7344E-03 | 5.7015E-02  | 5.7822E-02  |
|       | -1.3319E-01 | -9.9943E-02 | -2.1580E-05 | -2.1580E-05 | 5.7015E-02  | 5.7822E-02  |
|       | -3.6525E-02 | -2.7753E-02 | -3.7105E-04 | -3.7105E-04 | 5.7015E-02  | 5.7822E-02  |</p>
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
OUTPUT

\[\text{AN}= 0.330 \quad \text{ALTP}= 0. \quad \text{T4}= 2990.59 \quad \text{ETAP}= 1.0000\]

MIDDLE SPOOL IS DUMMY

NO AEROFLOW INTO WING

THE OUTPUT IS IN ENGLISH UNITS

<table>
<thead>
<tr>
<th>PCNF</th>
<th>CNF</th>
<th>ZF</th>
<th>PRF</th>
<th>WAF</th>
<th>WAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.102552E+03</td>
<td>0.102551E+01</td>
<td>0.834571E+00</td>
<td>0.300924E+01</td>
<td>3.222308E+03</td>
<td>0.222303E+03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCNC</th>
<th>CNC</th>
<th>ZC</th>
<th>PRC</th>
<th>WAC</th>
<th>WAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.118482E+03</td>
<td>0.989871E+00</td>
<td>0.814729E+00</td>
<td>0.848952E+01</td>
<td>5.513288E+02</td>
<td>0.139473E+03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T2</th>
<th>P2</th>
<th>T21</th>
<th>P21</th>
<th>T3</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.518670E+03</td>
<td>0.100000E+01</td>
<td>0.744583E+03</td>
<td>0.300924E+01</td>
<td>0.147148E+04</td>
<td>0.255408E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCLLP</th>
<th>BLP</th>
<th>PCBLC</th>
<th>BLC</th>
<th>PCBLO</th>
<th>BLOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.160847E+00</td>
<td>0.160000E+00</td>
<td>0.221552E+02</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCLLP</th>
<th>9LHP</th>
<th>PCLLP</th>
<th>BLLP</th>
<th>T4</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.726000E+00</td>
<td>0.160847E+02</td>
<td>0.660000E-01</td>
<td>0.146224E+03</td>
<td>0.293059E+04</td>
<td>0.293059E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W43</th>
<th>WPB</th>
<th>WQA</th>
<th>PAK</th>
<th>ETA</th>
<th>DPCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.115315E+03</td>
<td>0.277750E+01</td>
<td>0.119392E+03</td>
<td>0.238793E-08</td>
<td>0.100000E+01</td>
<td>0.560359E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TFFHP</th>
<th>CNHP</th>
<th>DHTCP</th>
<th>DHTC</th>
<th>T5</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.499981E+02</td>
<td>0.200241E+01</td>
<td>0.734620E+00</td>
<td>0.213088E+03</td>
<td>0.210970E+04</td>
<td>0.570541E+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TFFLP</th>
<th>CNLP</th>
<th>DHTCLP</th>
<th>DHTC</th>
<th>T55</th>
<th>P55</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.130779E+03</td>
<td>0.230199E+01</td>
<td>0.423525E+01</td>
<td>0.293510E+02</td>
<td>0.178415E+04</td>
<td>0.270347E+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCBLDU</th>
<th>BLDU</th>
<th>T24</th>
<th>P24</th>
<th>T25</th>
<th>P25</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.204000E+00</td>
<td>0.460828E+01</td>
<td>0.743340E+03</td>
<td>0.293418E+01</td>
<td>0.293418E+01</td>
<td>0.293418E+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WAD</th>
<th>WPB</th>
<th>WQA</th>
<th>PAK</th>
<th>ETA</th>
<th>DPCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.884383E+02</td>
<td>0.160847E+02</td>
<td>0.560359E+01</td>
<td>0.560359E+01</td>
<td>0.560359E+01</td>
<td>0.560359E+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ETA</th>
<th>STAC</th>
<th>ETACHP</th>
<th>STATLP</th>
<th>AM55</th>
<th>AM25</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.847718E+00</td>
<td>0.313972E+00</td>
<td>0.877127E+00</td>
<td>0.920172E+00</td>
<td>0.283951E+00</td>
<td>0.391747E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T6</th>
<th>P6</th>
<th>P56</th>
<th>AM6</th>
<th>V6</th>
<th>W66</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.141910E+04</td>
<td>0.272325E+01</td>
<td>0.262221E+07</td>
<td>0.243037E+00</td>
<td>0.439073E+00</td>
<td>0.225077E+03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T7</th>
<th>WPA</th>
<th>WG7</th>
<th>PAP7</th>
<th>ETA</th>
<th>DPAPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.141910E+04</td>
<td>0.000000</td>
<td>0.225077E+03</td>
<td>0.124944E+08</td>
<td>0.303330E+00</td>
<td>0.549672E+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PS8</th>
<th>AM8</th>
<th>V8</th>
<th>PS9</th>
<th>AM9</th>
<th>V9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.133947E+01</td>
<td>0.100000E+01</td>
<td>0.167738E+04</td>
<td>0.138387E+01</td>
<td>0.100000E+01</td>
<td>0.157739E+04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PS28</th>
<th>AM28</th>
<th>V28</th>
<th>PS29</th>
<th>AM9</th>
<th>V29</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BYPASS</th>
<th>HPEXZ</th>
<th>WPT</th>
<th>WST</th>
<th>VA</th>
<th>FRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.605403E+00</td>
<td>0.000000</td>
<td>0.277750E+01</td>
<td>0.225077E+03</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CVWNOZ</th>
<th>VJN</th>
<th>CVJNZ</th>
<th>VJD</th>
<th>FSM</th>
<th>FIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.949400E+02</td>
<td>0.159259E+04</td>
<td>0.159259E+04</td>
<td>0.000000</td>
<td>0.111435E+05</td>
<td>0.239548E+04</td>
</tr>
</tbody>
</table>

MAIN SONIC CONVERGENT NOZZLE

| FG = 13536.00 | FN = 13536.00 | SFC = 3.73370 |

CONVERGED AFTER 9 LOOPS
OUTPUT

AM  =  0.000  ATTP  =  0.  T4  =  2891.96  ETAP  =  1.0000

MIDDLE SPOOL IS DUMMY

NO AIRFLOW INTO WING
THE OUTPUT IS IN ENGLISH UNITS

PCNF  0.102579E 03  CNF  0.102579E 01  ZF  0.825774E 00  PRF  0.293938E 01  WAPC  0.222789E 03  WAF  0.222789E 03
PCNC  0.119421E 03  CWC  0.984202E 00  ZC  0.814219E 00  PRC  0.999040E 01  WACC  0.151172E 02  WAC  0.137799E 03
T2  0.518670E 03  T21  0.100000E 01  T22  0.743148E 03  T23  0.293938E 01  T3  0.146874E 04  T33  0.253972E 02
PCBLF  0.000000E 00  BLF  0.000000E 00  PCBLC  0.160000E 00  BLC  0.230462E 02  PCBLDB  0.300000E 00  BLOB  0.000000E 00
PCBLHP  0.726000E 00  BLHP  0.160044E 00  PCBLHP  0.660000E 01  BLHP  0.145948E 01  TPB  0.399196E 04  P3  0.239530E 02
WA3  0.115734E 03  WFB  0.118448E 03  WGA  0.118448E 03  FAB  0.100000E 01  ETAB  0.551209E 01
TFFHP  0.499974E 02  CHHP  0.200047E 01  DHTHP  0.735434E 01  DHTC  0.212685E 03  T5  0.555558E 01
TFFLP  0.130333E 03  CHLP  0.230656E 01  DHTLP  0.425347E 01  DHTF  0.894282E 02  T55  0.257101E 02
PCBLDU  0.208000E 00  BLDU  0.208000E 00  PCBLDU  0.781733E 03  T24  0.281184E 01  T25  0.281184E 01
WAD  0.895909E 02  WFD  0.000000E 00  WG24  0.895309E 02  PAR24  0.000000E 00  ETAD  0.529283E 01
ETAF  0.847070E 00  ETAC  0.319197E 00  ETAFHP  0.817121E 03  ETATLP  0.902439E 03  AM55  0.399315E 03
T6  0.160908E 04  P6  0.160908E 04  PS6  0.259177E 01  AM6  0.245431E 03  V6  0.225534E 03
T7  0.140908E 04  WFA  0.140908E 04  W7  0.000000E 00  W25  0.000000E 00  ETAA  0.654355E 01
PS8  0.136756E 01  AM8  0.136756E 01  V8  0.136756E 01  PS8  0.136756E 01  AM9  0.136756E 01
PS28  0.000000E 00  AM28  0.000000E 00  V28  0.000000E 00  PS29  0.000000E 00  AM29  0.000000E 00
BYPASS  0.616972E 00  HPEXT  0.000000E 00  WBT  0.275000E 03  WST  0.225534E 03  VA  0.000000E 00  FPD  0.000000E 00
CVMNOZ  0.949400E 00  VMN  0.949400E 00  CVMN27  0.300000E 00  V30  0.000000E 00  FM  0.311248E 05  FPD  0.231661E 04

MAIN SONIC CONVERGENT NOZZLE
PG  =  13441.39  FH  =  13441.39  SFC  =  0.73643

CONVERGED AFTER 10 LOOPS
### B

-3.3971E+02  -1.8844E+01
-7.6357E+00  8.1031E+01
2.7458E+01  -3.6972E+01
-7.3679E-01  -7.4372E+01
3.0562E+01  -5.9852E+01
9.4730E+00  -1.0116E+00
3.3675E+04  -4.6482E+01
1.9901E+00  -2.8772E+00
2.0804E+03  1.1746E+03
5.3638E+03  -1.0983E+02
9.7612E+00  -1.6070E+00
-1.0343E+01  -3.4356E+00
-5.7335E+00  4.6618E+01

### D

AR

-5.9400E+00  1.9103E+00  3.2736E+00  -4.8070E+00  1.5055E+01  -2.9179E+02  2.0989E+03  3.5559E+01  -4.1958E+03
-5.6609E-01  -5.4974E+00  -8.6566E+00  -5.3708E+01  -1.6537E+02  1.2807E+03  1.2051E+03  9.515E+02  -3.5698E+03
3.5318E+02  1.7256E+00  -7.5892E+00  5.8571E+00  2.6599E+01  3.7053E+00  -1.6503E+02  3.0925E+00  -6.3803E+01
-1.0296E-03  -2.0348E-03  -2.7554E-03  -2.7788E+00  4.0726E+00  -1.4569E+02  5.9931E+00  -1.0465E+01  2.3315E+02
-5.3101E-02  3.6428E+00  3.9359E+00  2.7322E+00  -7.1227E+00  1.7047E+00  1.5611E+00  2.3756E+00  -3.3977E+03
1.2848E+00  1.3302E+00  1.1671E+00  -3.1032E+00  2.5177E+00  -3.9499E+00  3.4759E+00  1.3313E+01  2.3358E+03
6.7089E-01  -1.2438E+01  -1.3578E-01  6.1651E+00  -1.2405E+02  2.4472E+00  -1.9902E+02  2.4235E+00  9.3698E+02
4.1728E-01  8.1583E+00  -2.3698E+00  4.3836E+00  1.3473E+01  3.0396E+00  -3.7222E+03  -2.2243E+02  -5.7709E+01
1.4055E+03  1.1603E+01  6.6580E+00  5.1543E+00  1.2537E+02  1.9911E+01  -1.5713E+01  2.3179E+02  -5.9552E+02

BR

2.3662E+02  2.3575E+01
-2.0947E+02  1.8122E+01
6.5854E+01  -4.5792E+00
-6.7336E+01  -7.4075E+01
1.2093E+02  -2.0314E+03
3.0576E+02  -5.6996E+00
-2.9802E+01  -4.3995E+00
3.3730E+04  -5.8471E+00
-2.3988E-01  -3.8622E+00

CR

-5.5621E+02  -2.8563E+06  6.7167E+00  -4.3107E+01  6.5199E+06  3.5167E+05  5.8821E+05  -2.6394E+05  1.3592E+22
1.2340E-02  5.1938E+02  -1.2284E+00  7.9221E+03  -7.6378E+02  9.1569E+01  -9.2352E+01  4.1135E+01  -2.3809E+02
1.2340E-02  5.1938E+02  -1.2284E+00  7.9221E+03  -7.6378E+02  9.1569E+01  -9.2352E+01  4.1135E+01  -2.3809E+02

DR

2.6715E+01  -2.4538E+00
2.9270E+01  4.4590E+03
2.9270E+01  4.4590E+03
REFERENCES


Nonlinear
Linear

Time (SEC)

(a-1) With 3-percent step change in main fuel flow.

Core Rotor Speed RPM

(a-2) With 3-percent step change in nozzle area.
b) With 3-percent step change in main fuel flow.

b-2) With 3-percent step change in nozzle area.

(b) Comparison of reduced-order linear and nonlinear runs.

Figure 1. - Response of state 1 - core rotor speed.
Comparison of full-order linear and nonlinear runs.

(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
\(0-1\) With 3-percent step change in main fuel flow.

\(0-2\) With 3-percent step change in nozzle area.

\(b\) Comparison of reduced-order linear and nonlinear runs.

Figure 2. - Response of state 2 - fan rotor speed.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.

(a) Comparison of linear and nonlinear runs.
Comparison of reduced-order linear and nonlinear runs.

Figure 3. - Response of state 3 - compressor-exit pressure.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.

(a) Comparison of full-order linear and nonlinear runs.
Comparison of reduced-order linear and nonlinear runs.

Figure 4. - Response of state 4 - compressor-exit internal energy.

(b) With 3-percent step change in main fuel flow.

(c) With 3-percent step change in nozzle area.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.

Comparison of full-order linear and nonlinear cases.
(b-1) With 3-percent step change in main fuel flow.

(b-2) With 3-percent step change in nozzle area.

Figure 5. Response of state 5 - afterburner-exit pressure.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
Comparison of reduced-order linear and nonlinear runs.

Figure 6. - Response of state 6 - combustor-exit internal energy.
8.1

(a-1) With 3-percent step change in main fuel flow.
(a-2) With 3-percent step change in nozzle area.

Nonlinear
Linear
(b) Comparison of reduced-order linear and nonlinear runs.

Figure 7. - Response of state 7: afterburner-exit internal energy.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
Comparison of reduced-order linear and nonlinear runs.

Figure 8. Response of state 8 - combustor-exit internal energy.
Comparison of full-order linear and nonlinear runs.

(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
Figure 9. - Response of state 9 - duct-burner-exit pressure.

(a) With 3-percent step change in main fuel flow.
(b) Comparison of reduced-order linear and nonlinear runs.
Compared to full-order linear and nonlinear runs.

(a-1) With 3 percent step change in main fuel flow.

(a-2) With 3 percent step change in nozzle area.
Comparison of reduced-order linear and nonlinear runs.

Figure 10. - Response of state 10 - fan-exit internal energy.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.

(a) Comparison of full-order linear and nonlinear runs.
(a-1) With 3-percent step change in main fuel flow.

(b) Comparison of reduced-order linear and nonlinear runs.

Figure 11. Response of state II - fan-exit pressure.
(b) Nonlinear and Linear results are compared in the graphs below:

- **Graph 1:** Duct burner exit internal energy vs. time (SLC).
  - (i) With 3-percent step change in main fuel flow.
  - (ii) With 3-percent step change in nozzle area.

- **Graph 2:** Duct burner exit internal energy vs. time (SLC).
Figure 12. - Response of state 12 - duct-burner-exit internal energy.

(b-1) With 3-percent step change in main fuel flow.

(b-2) With 3-percent step change in nozzle area.

Comparison of reduced-order linear and nonlinear runs.
a-1) With 3-percent step change in main fuel flow.

a-2) With 3-percent step change in nozzle area.
Figure 13. Response of state 13—high turbine-exit pressure.

(a) With 3-percent step change in main fuel flow.

(b) Comparison of reduced-order linear and nonlinear runs.

(b-2) With 3-percent step change in nozzle area.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
(b) Comparison of reduced-order linear and nonlinear runs.

Figure 14. - Response of state 14 - high turbine-exit internal energy.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
(a-1) With 3-percent step change in main fuel flow.

(b-1) Comparison of reduced-order linear and nonlinear runs.

Figure 15. - Response of state 15 - low turbine-exit pressure.

(b-2) With 3-percent step change in nozzle area.
LOA WOIR LEC EX T 4. L ENGE
Nonlinear
Linear

(a-1) With 3-percent step change in main fuel flow.
(a-2) With 3-percent step change in nozzle area.
Figure 16. - Response of state 16 - low turbine-exit internal energy.

b) Comparison of reduced-order linear and nonlinear runs.

-1) With 3-percent step change in main fuel flow.

-2) With 3-percent step change in nozzle area.
(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
b) Comparison of reduced-order linear and nonlinear runs.

Figure 17. - Response of output 1 - specific fuel consumption.
Comparison of full-order linear and nonlinear runs.

(a-1) With 3-percent step change in main fuel flow.

(a-2) With 3-percent step change in nozzle area.
(b-1) With 3-percent step change in main fuel flow.

(b-2) With 3-percent step change in nozzle area.

Comparison of reduced-order linear and nonlinear runs.

Figure 18. - Response of output 2 - gross thrust.
Figure 19. - Response of output 3 - net thrust.

(a) With 3-percent step change in main fuel flow.

(b) Comparison of reduced-order linear and nonlinear runs.

(c) With 3-percent step change in nozzle area.
A digital computer program, DYGABCD, has been developed that will generate linearized, dynamic models of simulated turbofan and turbojet engines. DYGABCD is based on an earlier computer program, DYNGEN, that is capable of calculating simulated nonlinear steady-state and transient performance of one- and two-spool turbojet engines or two- and three-spool turbofan engines. Most control design techniques require linear system descriptions. For multiple-input/multiple-output systems such as turbine engines, state space matrix descriptions of the system are often desirable. DYGABCD computes the state space matrices commonly referred to as the A, B, C, and D matrices - required for a linear system description. The report discusses the analytical approach and provides a users manual, FORTRAN listings, and a sample case. NASA TN D-7901, describing DYNGEN, is a necessary adjunct to this report.