PREDICTION OF PRESSURE FLUCTUATION IN SOUNDING ROCKETS AND MANIFOLDED RECOVERY SYSTEMS

JOHN F. LAUDADIO

JUNE 1972

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND
PREDICTION OF PRESSURE FLUCTUATION
IN SOUNDING ROCKETS AND MANIFOLDED
RECOVERY SYSTEMS

John F. Laudadio

June 1972

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>DEVELOPMENT OF EQUATIONS</td>
<td>2</td>
</tr>
<tr>
<td>CONCLUSIONS</td>
<td>7</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>15</td>
</tr>
<tr>
<td>APPENDIX A — 1. Computer program objective</td>
<td>A-1</td>
</tr>
<tr>
<td>2. Philosophy</td>
<td>A-1</td>
</tr>
<tr>
<td>3. Options</td>
<td>A-6</td>
</tr>
<tr>
<td>4. Special use options</td>
<td>A-8</td>
</tr>
<tr>
<td>5. Table definitions</td>
<td>A-10</td>
</tr>
<tr>
<td>6. Variable names input and output</td>
<td>A-12</td>
</tr>
<tr>
<td>7. Discussion of variable input program</td>
<td>A-15</td>
</tr>
<tr>
<td>8. Plot option</td>
<td>A-17</td>
</tr>
<tr>
<td>APPENDIX B — Test case and inputs</td>
<td>B-1</td>
</tr>
<tr>
<td>APPENDIX C — Numerical integration</td>
<td>C-1</td>
</tr>
<tr>
<td>APPENDIX D — Program Listing</td>
<td>D-1</td>
</tr>
</tbody>
</table>
Symbols

1  Tube flow
2  Orifice flow
A  Area, ft$^2$
α  Angle-of-attack, deg
$C_p$ Constant pressure specific heat, BTU/lbm - °R
$C_q$ Discharge coefficient
$C_v$ Constant volume specific heat, BTU/lbm - °R
f  Friction factor
φ  Roll angle
g  Gravitational Constant (32.174 lbm-ft/lbf-sec$^2$)
h  Specific enthalpy, BTU/lbm
K  Thermal conductivity of Air (BTU/ft-sec °R)
L  Tube length, ft
M  Mach Number
m  Mass, lbm
$\dot{m}$ Mass flow rate, lbm/sec
P  Pressure, lbf/ft$^2$
R  Tube radius, ft
$R_e$ Reynolds Number
$\rho$ Density, lbm/ft$^2$
T  Temperature, °R
t  Time, sec
U  Internal energy, BTU/lbm
V  Volume, ft$^3$
$\bar{V}$ Velocity, ft/sec

Subscripts

a  Mean value
e  Exit conditions
i  Inlet conditions
m  Manifold
n  Tube or orifice number
∞  Free Stream conditions
PREDICTION OF PRESSURE FLUCTUATION IN SOUNDING ROCKETS AND MANIFOLDED RECOVERY SYSTEMS

SUMMARY

The determination of altitude by means of barometric sensors is used in sounding rocket applications. Consequently, a method for predicting the performance of such sensing systems is needed. Herein a method is developed for predicting the pressure-time response of a volume subjected to subsonic air flow through from one to four passages. The pressure calculation is based on one-dimensional gas flow with friction.

In addition, a computer program has been developed which solves the differential equations using a self-starting predictor-corrector integration technique. The input data required are the pressure sensing system dimensions, pressure forcing function(s) at the inlet port(s), and a trajectory over the time of analysis (altitude-velocity-time), if the forcing function is trajectory dependent. The program then computes the pressure-temperature history of the gas in the manifold over the time interval specified.

INTRODUCTION

This analysis, undertaken at the time of the development of the Aerobee 350 Recovery System, has led to the development of a set of equations that describe the pressure and temperature histories within a manifold connected to a pressure source or sources. The source pressure histories are assumed to be known either as a function of Mach Number or time. In addition, a computer program which solves these equations is presented.

This manifold system is used to initiate an operating sequence of a rocket vehicle or payload recovery system at a preset altitude. The chief difficulty which the analytical method has to deal with is predicting the sensitivity of a given manifold system to sensing the true altitude in terms of the pressures existing at the external surface of the vehicle during flight. Further, the barostat connection to the pressure source is a variable. Therefore, in order to achieve the greatest generality, flow from the vehicle exterior to the manifold is assumed to pass through either tubes or orifices. Following is the development of the tube flow equations.
Assumptions

The following assumptions are made in developing the equations used in this analysis:

1. The pressure, density and temperature are distributed evenly and instantaneously throughout the manifold.
2. The pressure, density and temperature at the port(s) are known for all times.
3. The specific heats, \( C_v \) and \( C_p \) are constant.
4. The volume of the manifold is much greater than the volume of any tube leading into it.
5. Continuum flow exists throughout the system.
6. Entrance effects have a negligible effect upon the tube flow.
7. An approximate equation for compressible adiabatic flow with friction can be used to calculate a mean value for velocity, given the mean density.
8. Mass continuity is satisfied; i.e., no mass addition in the manifold other than from the tubes.
9. Steady flow exists over the integration interval.
10. The behavior of air can be closely approximated by treating it as a perfect gas.

DEVELOPMENT OF EQUATIONS

The following theoretical development considers the case of tube flow from the external vehicle surface into the manifold and then discusses the variations in procedure required to account for orifice flow. A sketch of the flow case and general nomenclature is given in Figure 1.

Where:
- \( T_m \) = Temperature in the manifold
- \( T_i \) = Temperature of the gas at the inlet port
- \( T_e \) = Temperature of the gas at the outlet port
- \( t \) = Time - sec
From the first law of thermodynamics for an open system:

$$Q + \dot{m}_i \left( h_i + \frac{\bar{V}_i^2}{2g} \right) = \dot{m}_e \left( h_e + \frac{\bar{V}_e^2}{2g} \right) + \frac{d}{dt} \left( U_m + m_m \frac{\bar{V}_m^2}{2g} \right) \quad \text{EQN 1}$$

The total internal energy and specific enthalpy in the manifold at any time are:

$$U_m = m_mC_vT_m \quad \text{EQN 2}$$

$$h_m = C_pT_m \quad \text{EQN 3}$$

Expanding the term describing the changes in the manifold.

$$\frac{d}{dt} \left( U_m + m_m \frac{\bar{V}_m^2}{2g} \right) = \frac{dU_m}{dt} + \frac{\bar{V}_m^2}{2g} \dot{m}_m + \frac{m_m}{g} \bar{V}_m \dot{\bar{V}}_m \quad \text{EQN 4}$$

Now, assuming that $\bar{V}_m$ and $\dot{\bar{V}}_m$ are negligible, we have $dU_m/dt$ as the total energy change in the manifold.

Since

$$U_m = m_mC_vT_m \quad \text{EQN 2}$$
and

\[ h = C_p T \]  
\[ EQN 5 \]

\[ \frac{dU_m}{dt} = C_v (\dot{m}_m T_m + \dot{T}_m m_m) \]  
\[ EQN 6 \]

\[ h_{flow} + \frac{\bar{V}_f^2}{2g} = h_{source} = C_p T_{source} \]  
\[ EQN 7 \]

Now, solving for \( \frac{dU_m}{dt} \) in Equation 1

\[ \frac{dU_m}{dt} = \dot{m}_i \left( h_i + \frac{\bar{V}_i^2}{2g} \right) - \dot{m}_e \left( h_e + \frac{\bar{V}_e^2}{2g} \right) + Q \]  
\[ EQN 8 \]

Substituting Equations 6 and 7 into Equation 8

\[ C_v (\dot{m}_m T_m + \dot{T}_m m_m) = (\dot{m}_i T_i - \dot{m}_e T_e) C_p + Q \]  
\[ EQN 9 \]

Solving for \( \dot{T}_m \) and collecting terms:

\[ \dot{T}_m = \frac{C_p \Sigma_n \dot{m}_n T_n + Q - C_p T_m \dot{m}_m}{C_v m_m} \]  
\[ EQN 10 \]

where \( \dot{m}_m = \Sigma \dot{m}_n \), i.e., there are no mass changes in the manifold other than those introduced by the flow.

The mass flow rate in the tubes is determined as follows. Using an approximate equation for compressible adiabatic flow with friction:

\[ P_i - P_m = \left[ 2f \rho_a \frac{L}{D} + \rho_a^2 \left( \frac{1}{\rho_m} - \frac{1}{\rho_i} \right) \right] \bar{V}_a^2 \]  
\[ EQN 11 \]

where \( \rho_a \) and \( \bar{V}_a \) are mean values and \( i \) may be replaced by \( e \) where applicable.

\( \rho_a \) is determined by taking the average of the densities of air at the end of the tube being analyzed and the air in the manifold. \( \bar{V}_a \) is then found from:

\[ \bar{V}_a = \left[ \frac{P_i - P_m}{f \rho_a \frac{L}{R} + \rho_a^2 \left( \frac{1}{\rho_m} - \frac{1}{\rho_i} \right)} \right]^{\frac{1}{2}} \]  
\[ EQN 12 \]
Knowing \( \rho_a \) and assuming a value for \( \bar{V}_a \), a Reynolds number and friction factor \( f \) is calculated for each tube and then a new \( \bar{V}_a \) is calculated from Equation 12. \( \bar{V}_a \) (calculated) is compared to \( \bar{V}_a \) (assumed) and if the relative difference is greater than one percent, the iteration is continued until the process converges. Next, \( f \) is calculated from the following empirical equations:

\[
f = 0.0008 + \frac{0.05525}{Re^{0.237}} \quad \text{Re} \geq 100000 \quad \text{EQN 13}
\]

\[
f = 0.0791/Re^{0.25} \quad 1185 < \text{Re} < 100000 \quad \text{EQN 14}
\]

\[
f = 16/Re \quad \text{Re} \leq 1185 \quad \text{EQN 15}
\]

The \( \rho_a \) and \( \bar{V}_a \) so found constitute a \( \rho_a \bar{V}_a \) couple which is used to compute the mass flow rate through the tube being analyzed.

\[
\dot{m}_a = \rho_a \bar{V}_a A \quad \text{EQN 16}
\]

\[
\frac{d\rho}{dt} = \frac{\dot{m}_{\text{manifold}}}{Vol_{\text{manifold}}} = \frac{\Sigma \dot{m}_{\text{tubes}}}{Vol_{\text{manifold}}} \quad \text{EQN 17}
\]

Equation 16 is used to compute the mass flow rate, \( \dot{m}_a \), for each tube. Since the \( d\rho/dt \) and \( dT/dt \) of the manifold are now known, we may integrate numerically Equations 10 and 17. The numerical integration technique is described in Appendix C.

Orifice Flow

For orifice flow there is one significant difference in the preceding calculations: The mass flow rate through each orifice is calculated using

\[
\dot{m}_a = C_q A \left( \frac{\Delta P}{\rho} \right)^{\frac{1}{2}} \quad \text{EQN 18}
\]

where \( C_q \) may be input as a variable and \( \rho \) is the density at the higher pressure.

For both tube flow and orifice flow the limiting condition of choked flow is assumed to occur at \( M = 1 \) and this condition is applied by the computer program to limit the flow rate when necessary.

The preceding analysis takes into account those variables which are felt to be significant for the problem analyzed. In most instances, the equations are programmed in an expanded form to facilitate checking and to provide a source of documentation to the user who may wish to identify the equations in their
programmed form. The program documentation will be found in Appendix A with a complete explanation of program variables and its use.

Results

Results from three tests are used to substantiate this analysis. First, a drop test of the Aerobee 350 recovery body at the El Centro Range in California, a DOD Parachute Test Facility. Next, data is used from the payload recovery of flight 17.05 GT-GG. Finally an experiment (Reference 10) which determines the pressure drop of a volume through various length and diameter tubes. The predictions of the manifold program will be compared to these tests and will be seen to match the test results closely.

Actual pressure data from the Aerobee 350 drop test was used as a forcing function in the computer simulation. Drogue deployment was set for 20,000 feet, a manifold pressure of 972 psf. Actual deployment took place at 18,700 feet where the manifold pressure reached 1000 psf. Measured deployment time was 33.35 seconds after test initiation while the computer simulation predicted deployment at 33.66 seconds. In addition, prior to the drop test a pre-flight analysis of the maximum range of deployment altitudes was performed for several possible test configurations using wind tunnel test data (Reference 12). Results predicted that for a desired 20,000 foot recovery initiation, drogue deployment would occur between 22,500 and 15,000 feet for all configurations. Figures 2a and 2b show the forcing pressures and computed manifold pressure for the drop test. Thus, these results show the applicability and accuracy of this analysis.

Pressure data from flight 17.05 GT-GG was also used as the forcing function in a computer program simulation. Figure 3a shows the forcing pressures and the computed manifold pressure while Figure 3b shows a comparison of the measured and calculated manifold pressures. As may be seen from Figure 3b, the computed manifold pressure very closely follows the measured manifold pressure with a slight lag. This slight lag in the prediction may be due to any or all of the following: Non-uniform tube diameter, internal roughness of the tube, bends in the tube, or the possibility that the nominal manifold system used for computation is not the same as the actual manifold system flown.

Another simulation of test data is presented in Figures 4 and 5 which show a comparison between program computations and some results obtained in Reference (10). The applied pressure differential and time shown in Figures 4 and 5 is the time to equilibrium for each applied pressure. For the experiments performed in Reference (10), the experimental error is of the order of ±15 percent so that the program computation is considered to be a reasonably accurate simulation of the test.
These comparisons of results from the Aerobee 350 drop test and Flight 17.05 GT-GG with the computations of the manifold computer program show very good agreement and tend to confirm the applicability of the analytical method. Similarly, the computed response times compare favorably with the results obtained in Reference (10) which further confirms the applicability of the equations for this specific application. Naturally, the equations and the manifold program will be subject to continued testing over diversified conditions in order to increase the confidence level as well as the proven regions of applicability.

CONCLUSIONS

This analysis, coupled with a computer program for solving the equations, can be used to predict the response time of a recovery system manifold connected to pressure sources on the surface of a re-entry body. In addition, the design of new manifoded recovery systems may be undertaken with confidence. The prediction of pressure variation in any ascending rocket vehicle is possible with the proper choice of a pressure forcing function and other system parameters. In general, as long as the assumptions of the analysis are met, the response of a volume subjected to a fluctuating pressure connected either by a tube(s) or orifice(s) may be analyzed.
Figure 2a. Manifold Pressures Aerobee 350 Drop Test 11/15/71
Figure 2b. Manifold Pressures Aerobee 350 Drop Test 11/15/71
Figure 3a. Manifold Prog Calc. Aerobee Flt 17.05
Figure 3b. Measured & Predicted Pressure for 17.05 Recovery
Figure 4. Pressure Response Through a Tube
Figure 5. Pressure Response Through a Tube

\[ \Delta P \text{ vs } t \]

TUBE SIZE \( d = 0.036" \), \( L = 135" \)
REFERENCES


2. Fluid Mechanics for Hydraulic Engineers by Rouse, Dover 1961


8. Subroutine Readin - A Generalized Data Loader Which Reads in Variable and/or Array Name (s) and Their Associated Data (Revised) by John F. Laudadio, March 17, 1969, 721.23/CGT/0195


APPENDIX A

A1 The purpose of the manifold program is to solve the set of equations presented in the body of this report. The solution to these equations is accomplished in a series of steps culminating in the numerical integration of Equations 10 and 17 from the body of this report.

\[
\dot{T}_m = \frac{C_p \sum n \dot{m}_n T_n + Q - C_p T_m \dot{m}_m}{C_v m_m} 
\]

EQN 10

\[
\frac{d}{dt} \frac{\dot{m}_m}{V_m} = \Sigma_n \frac{\dot{m}_n}{V_m} 
\]

EQN 17

The program is set up in steps so that various configuration and environmental parameters may be included in the solution of the equations, either together or separately. This flexibility allows isolation of effects of the individual parameters on the pressure system being analyzed. In order to include this flexibility and yet make the input as simple as possible two concepts have been used with regard to the input. The first is the default option which means simply that the basic options needed to analyze the simplest case will be selected automatically in lieu of instructions to the contrary. The second concept is that the user need specify only those options which most suit the user's analytical model and data. The remainder of Appendix A will be used to describe the input variables and their use.

A2 The following are program variables/input names and are defined prior to describing the input options.
FLOW CHART OF MANIFOLD PROGRAM

MAIN
CALL INPUT SUBROUTINE
TRANSFER DATA FROM INPUT ARRAY TO PROGRAM NAMES.
INPUT PRESSURE COEFFICIENT DATA.
CALL PRESSURE SUBROUTINE TO BEGIN CALCULATION.

READIN
INPUT SUBROUTINE

PRESUR

INITIALIZE CONSTANTS
INITIALIZE INTEGRATION ROUTINE

START CALCULATION
CALL INTEGRATION ROUTINE
OUTPUT
RETURN TO MAIN
WHEN FINISHED WITH PROBLEM

SINTRP

QUADRATIC INTERPOLATION FOR MOST TABLES
CSUBP
LINEAR INTERPOLATION FOR FOUR WAY CSUBP TABLE
INTEGT
INTEGRATION ROUTINES
PLOT ROUTINE
<table>
<thead>
<tr>
<th>PROGRAM VARIABLE NAME AND INPUT NAME</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT</td>
<td>Altitude Table — Ft.</td>
</tr>
<tr>
<td>CQ</td>
<td>Discharge Coefficient table for orifice flow.</td>
</tr>
<tr>
<td>FREN0</td>
<td>The independent variable for the CQ table. The function may be either Reynolds Number, the square root of Reynolds Number or pressure ratio where ( P/P_o &gt; 1 ).</td>
</tr>
<tr>
<td>NCQ</td>
<td>The number of pairs of points in the CQ-FREN0 table.</td>
</tr>
<tr>
<td>INDPVR</td>
<td>Indicates the type of independent variable FREN0 = 0 = (ReNo)², - 1 = Re No, 1 = P/Po (P/Po &gt; 1).</td>
</tr>
<tr>
<td>PRINTFQ</td>
<td>The time interval between printed output steps — should always be used, otherwise every step will be output.</td>
</tr>
<tr>
<td>BDUMP</td>
<td>BDUMP is input equal to 1 for all name load input to be printed, otherwise it is neglected.</td>
</tr>
<tr>
<td>TINIT</td>
<td>Initial time.</td>
</tr>
<tr>
<td>TEND</td>
<td>Time at which program is to stop.</td>
</tr>
<tr>
<td>NOPTS</td>
<td>Number of points in the Altitude-Velocity-Time Table, i.e., number of time points.</td>
</tr>
<tr>
<td>ALPHA</td>
<td>Input option to have the angle-of-attack vary with time. ALPHA is input only if this option is to be implemented, with ( \text{ALPHA} = \text{Number of points in the } \alpha - \text{T table} ) (maximum of 10).</td>
</tr>
<tr>
<td>ALPHAA</td>
<td>Name of the angle-of-attack array.</td>
</tr>
<tr>
<td>TALPH</td>
<td>Name of time array associated with ALPHAA.</td>
</tr>
<tr>
<td>CASES</td>
<td>Number of cases to be run.</td>
</tr>
<tr>
<td>PROGRAM VARIABLE NAME AND INPUT NAME</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>VR</td>
<td>Manifold Volume — input as inches³, converted internally to feet³.</td>
</tr>
<tr>
<td>NEWFMT</td>
<td>Used to trigger NAMELOAD subroutine to read in user input case label.</td>
</tr>
<tr>
<td>OMEGA</td>
<td>Roll rate in-CPS (optional input).</td>
</tr>
<tr>
<td>PM</td>
<td>Initial Pressure in the manifold — PSF.</td>
</tr>
<tr>
<td>RO</td>
<td>Initial air density in the manifold lbm/ft³.</td>
</tr>
<tr>
<td>TM</td>
<td>Initial air temperature in the manifold °R.</td>
</tr>
<tr>
<td>Note: Only one of RO or TM need be input; the other is calculated in the program.</td>
<td></td>
</tr>
<tr>
<td>PLTCON</td>
<td>= 1 Enables user to continue plotting from one case to the next, i.e., could also be used when SAVE = 1 is used.</td>
</tr>
<tr>
<td>IPlot</td>
<td>The array in which the variables to be plotted are specified.</td>
</tr>
<tr>
<td>PLOTFQ</td>
<td>Plot every PLOTFQ number of points (see A7 for explanation).</td>
</tr>
<tr>
<td>DELT</td>
<td>Initial time step, default = 0.0625.</td>
</tr>
<tr>
<td>DELMAX</td>
<td>Maximum step size allowed by the user; if DELMAX is not specified there is no step size limiter. It is not normally necessary to limit step size but should be used to ensure that a minimum number of points are computed for output, i.e., that the integration step is not so large that very little output is received.</td>
</tr>
<tr>
<td>IERCRT</td>
<td>Error criteria for the numerical integrator.</td>
</tr>
<tr>
<td>ERCRIT(1)</td>
<td>integration—preset.</td>
</tr>
<tr>
<td>ERCRIT(2)</td>
<td></td>
</tr>
<tr>
<td>PROGRAM VARIABLE NAME AND INPUT NAME</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>JMAX</td>
<td>Number of equations to be solved—preset.</td>
</tr>
<tr>
<td>NØNLIN</td>
<td>Non-Linear Integration Option—preset.</td>
</tr>
<tr>
<td>ICP</td>
<td>The number of different CP tables to be input, i.e., succeeding cases may use either the same or a different CP table.</td>
</tr>
<tr>
<td>KT</td>
<td>The number of tubes or orifices connected to the manifold, default = 1, Max No. = 4.</td>
</tr>
<tr>
<td>NØQ</td>
<td>Option selector for tube or orifice flow default = tube, NØQ = 2 for orifice.</td>
</tr>
<tr>
<td>RT</td>
<td>Recovery Temperature factor for aero heating in boundary layer. (Usually around 0.9).</td>
</tr>
<tr>
<td>SAVE</td>
<td>SAVE = 1 saves PM, TM and t from previous case to use in next case. This would be used when there is too much pressure forcing function data to input into the data array for a one case run. This would enable the user to just input more tables for each succeeding run. However, SAVE must be re-initialized for each case where it is desired. In addition, TEND must be respecified for each succeeding data set.</td>
</tr>
<tr>
<td>OPTSEL and CPCNTL</td>
<td>OPTSEL is used as one part of a two part option selector. OPTSEL selects the type of pressure function option. CPCNTL is the second part of the two part option selector. CPCNTL selects Mach Number or time and symmetric or non-symmetric tables CP table definition.</td>
</tr>
<tr>
<td>IXMAX</td>
<td>Number of Mach numbers or times to be input.</td>
</tr>
<tr>
<td>IYMAX</td>
<td>Number of angles-of-attack to be input.</td>
</tr>
<tr>
<td>IZMAX</td>
<td>Number of Ø-C_p pairs to be input. The C_p table is the pressure forcing function and may be defined as any of the input options d through i.</td>
</tr>
<tr>
<td>NAME</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PIPSIZ</td>
<td>Dummy array containing pipe or orifice dimensions (inches).</td>
</tr>
<tr>
<td>PIPSIZ 1</td>
<td>Radius (1)</td>
</tr>
<tr>
<td>PIPSIZ 2</td>
<td>Length (1)</td>
</tr>
<tr>
<td>PIPSIZ 3</td>
<td>Radius (2)</td>
</tr>
<tr>
<td>PIPSIZ 4</td>
<td>Length (2)</td>
</tr>
<tr>
<td>PIPSIZ 5</td>
<td>Radius (3)</td>
</tr>
<tr>
<td>PIPSIZ 6</td>
<td>Length (3)</td>
</tr>
<tr>
<td>PIPSIZ 7</td>
<td>Radius (4)</td>
</tr>
<tr>
<td>PIPSIZ 8</td>
<td>Length (4)</td>
</tr>
</tbody>
</table>

where length = 0 for orifice flow

A3 General Option Description

The different options are the following:

Type of System
- Tube flow
- Orifice flow

Parameters that may be included in the analysis
- Trajectory input
- Temperature function at the port
- Type of Pressure forcing function
- Type of discharge coefficient for orifice flow
- Roll rate. Note: This option requires that the integration step be smaller than other options require and therefore takes more computer time.
- Output all nameloaded input data
- Plot the results
- Include angle-of-attack if required by pressure forcing function tables.

The preceding options are possible functions that the user may include in his analysis. It is up to the user to define his problem within the program's framework.
Detailed Option List:

<table>
<thead>
<tr>
<th>OPTION</th>
<th>NUMBER/LETTER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Tube flow.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Orifice flow.</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>Trajectory input required, altitude, velocity, time.</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>Ambient temperature used at port.</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>Recovery temperature used at port. TR = TA (1+2<em>RT</em>M).</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>Pressure forcing function (C_p = \frac{\Delta P}{q} = f(M, a, \phi)).</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>Pressure forcing function (C_p = \frac{P_1}{P_0} = f(M, a, \phi), \left(\frac{P_1}{P_0} &gt; 1\right)).</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>Pressure forcing function (C_p = P_1 = f(M, a, \phi), \text{lb/ft}^2).</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>Pressure forcing function (C_p = \frac{\Delta P}{q} = f(t, a, \phi)).</td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>Pressure forcing function (C_p = \frac{P_1}{P_0} = f(t, a, \phi), \left(\frac{P_1}{P_0} &gt; 1\right)).</td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>Pressure forcing function (C_p = P_1 = f(t, a, \phi) (\text{lb/ft}^2)).</td>
<td></td>
</tr>
<tr>
<td>j</td>
<td>Variable Discharge Coefficient (C_Q = f(Re^{\frac{1}{2}})).</td>
<td></td>
</tr>
<tr>
<td>k</td>
<td>Variable Discharge Coefficient (C_Q = f(Re)).</td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>Variable Discharge Coefficient (C_Q = f\left(\frac{P_1}{P_0}\right), \left(\frac{P_1}{P_0} &gt; 1\right)).</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>Input constant roll rate for vehicle simulation, cps.</td>
<td></td>
</tr>
<tr>
<td>o</td>
<td>Output all input data except (C_p) table which is always output automatically.</td>
<td></td>
</tr>
<tr>
<td>OPTION NUMBER/ LETTER</td>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>Plot up to 10 dependent variables as a function of any other variable.</td>
<td></td>
</tr>
<tr>
<td>q</td>
<td>Input angle-of-attack as a function of time to be used by the Cp table. This table is needed only when Cp is also a function of angle-of-attack.</td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>Input non-symmetric Cp tables, i.e., $0 &lt; \theta &lt; 360$.</td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>Use a simple one way table input into the altitude time array (when no trajectory is needed) to solve for the pressure fluctuation in a manifold. For example, this could be used to analyze the response of a pressure measuring system used in an experiment or to determine the frequency response in a transducer-tube measuring system.</td>
<td></td>
</tr>
<tr>
<td>t</td>
<td>Hold the manifold temperature constant to affect an isothermal solution to a problem. This is accomplished by inputting the desired temperature with a minus sign.</td>
<td></td>
</tr>
<tr>
<td>u</td>
<td>Save the manifold pressure and temperature and time to use in the next case, i.e., internal initialization of next case when multiple Cp tables are required because there is too much Cp data to input to one case.</td>
<td></td>
</tr>
</tbody>
</table>

A4 Selection of options for input, i.e., the input name and value input to implement each option. Default options are those options the program will select automatically if no other option selection is made.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>INPUT NAME(S)</th>
<th>OPTION SELECTION OR INPUT VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NØQ</td>
<td>0 (default)</td>
</tr>
<tr>
<td>2</td>
<td>NØQ</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>ALT, VEL, TF</td>
<td>Input data</td>
</tr>
<tr>
<td>b</td>
<td>RT</td>
<td>0 (default)</td>
</tr>
<tr>
<td>OPTION</td>
<td>INPUT NAME(S)</td>
<td>OPTION SELECTION OR INPUT VALUE</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>c</td>
<td>RT</td>
<td>Equal to the recovery temperature factor</td>
</tr>
<tr>
<td>d</td>
<td>OPTSEL, CPCNTL</td>
<td>0, 0 (default)</td>
</tr>
<tr>
<td>e</td>
<td>OPTSEL, CPCNTL</td>
<td>-1, 0</td>
</tr>
<tr>
<td>f</td>
<td>OPTSEL, CPCNTL</td>
<td>-2, 0</td>
</tr>
<tr>
<td>g</td>
<td>OPTSEL, CPCNTL</td>
<td>0, -1</td>
</tr>
<tr>
<td>h</td>
<td>OPTSEL, CPCNTL</td>
<td>-1, -1</td>
</tr>
<tr>
<td>i</td>
<td>OPTSEL, CPCNTL</td>
<td>-2, -1</td>
</tr>
<tr>
<td>j</td>
<td>INDPVR</td>
<td>0 (default)</td>
</tr>
<tr>
<td>k</td>
<td>INDPVR</td>
<td>-1</td>
</tr>
<tr>
<td>l</td>
<td>INDPVR</td>
<td>1</td>
</tr>
<tr>
<td>n</td>
<td>OMEGA</td>
<td>Roll rate cps</td>
</tr>
<tr>
<td>o</td>
<td>BDUMP</td>
<td>1</td>
</tr>
<tr>
<td>p</td>
<td>IPLOT, NCURVS, PLTCON</td>
<td>See A7</td>
</tr>
<tr>
<td>q</td>
<td>ALPHA</td>
<td>Number of pairs of points in table. Then the table is input under arrays.</td>
</tr>
</tbody>
</table>

Note: If an option is a default selection it need not (but may) be included in the input stream.
The input names OPTSEL and CPCNTL are used jointly to define the type of pressure function being input to the program. For some options, i.e., d, e, f, it would not be necessary to input CPCNTL since 0 is the default value for CPCNTL but it might also be input solely for the sake of clarifying the input. In addition, the type of pressure function may change from case to case if so specified. Otherwise, it will remain the same as the previous case, i.e., symmetric option for case 1, non-symmetric for case 2, etc.

A5 Table Definitions

Input Names will be set off by quotes. Each type of table which may be input will be discussed. Quadratic interpolation is used unless otherwise specified. No extrapolation of any tables is performed. The end points are used as the extreme values. All tables, except the C_p table, are input via the name load subroutine, which is input using standard Fortran formats. The method of using the name load option is described in A6.

Option a calls for "Altitude", "Velocity", "Time" tables representing a trajectory experienced by the vehicle being analyzed with altitude and velocity being taken at the same time.

"ALT" is the name of the altitude table
"VEL" is the name of the velocity table
"TF" is the name of the time table

Each table may contain a maximum of 50 points. "NPTS" is input as the number of sets of ALT-VEL-TIME points. The tables used for options d-i are linearly interpolated. This pressure forcing function table is input under standard FORTRAN input rules, i.e., no name load is used. The reason is that it
would be inefficient to input the amount of data required for the \( C_p \) table under a name load option. The tables are specified as follows:

"IX" = Number of Mach Numbers or time
"IY" = Number of angles-of-attack per Mach Number or time
"IZ" = Number of roll angle-\( C_p \) pairs per angle-of-attack

The input format for IX, IY and IZ is (3I5) on the first card of the \( C_p \) table data set. IX, IY and IZ are fixed point, i.e., no decimal point, and are right justified. The \( C_p \)’s for any option are then input as follows (Format = 2E15.8).

Mach Number or time whichever is the independent variable
Angle-of-attack
Roll angle-\( C_p \) as determined by user input selection. These are input as follows. Suppose \( IX = 2 \) \( IY = 2 \) \( IZ = 2 \)

\[
\begin{align*}
M_1 \text{ or } t_1 \\
a_1 \\
\phi_1, C_p \\
\phi_2, C_p \\
a_2 \\
\phi_1, C_p \\
\phi_2, C_p \\
M_2 \text{ or } t_2 \\
a_1 \\
\phi_1, C_p \\
\phi_2, C_p \\
a_2 \\
\phi_1, C_p \\
\phi_2, C_p \\
\text{Etc.}
\end{align*}
\]

The sample problems show the placement of the data on cards and position in the input stream. The \( C_p \) tables are input after all the name load data for a particular case has been given. Up to 10 Mach Numbers or times may be input. Up to 15 angles-of-attack (per independent variable) may be input. Up to 20 \( \phi \)-\( C_p \) pairs may be input per angle-of-attack. If the user requires more space for \( C_p \) data then option u would be selected allowing the continuation of the \( C_p \) table.
Options j-i, the variable discharge coefficient table is input using the same load subroutines.

"FRENO" is the name of the independent variable
"CQ" is the name of the dependent variable

Where FRENO may be Reynolds Number, the square root of Reynolds Number, or a pressure ratio greater than 1. CQ is a value between 0 and 1. Each variable is input separately under the name load subroutine. There must be the same number of points in each table. "NCQ" is the variable that specifies this option in the input stream and is also the name used to specify the number of pairs of points in this table.

Option Q is the angle-of-attack variation table.

The name of the independent variable is "TALPH"
The name of the dependent variable is "ALPHAA"

Where TALPH is flight time and ALPHAA is angle-of-attack in degrees. This option is implemented by inputting the variable name "ALPHA" with the number of pairs of points in this table. A maximum of ten points may be input into each array.

Examples of table input will be given in the section on Input.

Note: All quadratically interpolated tables must have at least three points (per variable) input. The linearly interpolated tables must have at least two points (per variable) input.

A6 Input Description

The data are input to the manifold program via a "name load" read-in subroutine, i.e., each piece of data has a name and is identified in the input stream by this name. This also applies to most array variables in which case the input name is just the array name without subscript. A complete description of such a subroutine may be found in Reference (8). This section will describe the method of using this name loader and will include sample inputs.
Name Load Input Options:

<table>
<thead>
<tr>
<th>NAME LOAD</th>
<th>OPTION NUMBER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTION NUMBER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Read in variable names, and data. Return to calling program, i.e., manifold main program.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Option 0 plus read another option card.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Read in an array name, and data. Return to calling program.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Option 2 plus read another option card.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Read in an array name specific array locations, and data. Return to calling program.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Option 6 and read another option card.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Any of the above option numbers prefixed by a 5, i.e., 50, 52, 56, etc., specifies that a format will be input for that option. The format for</td>
<td>the variable name read-in and array name read-in must be specified separately. Also, unless respecified the last format input will apply for either succeeding variables or arrays.</td>
</tr>
</tbody>
</table>

Note 1: IF NO FORMAT is specified initially, a standard format of 10G8.4 will be used by both name load and array load options. This means that to change the format, an option number must be prefixed by a 5. However, the standard format should be sufficient for most users.

Note 2: All data used in the manifold program, when it is input via name load routine, is input in floating point. The only fixed point data input to the manifold program are the table sizes for the M, a, Ø or t, a, Ø tables which are loaded in regular FORTRAN formats as described in A5.
CARD INPUT DETAILS

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Card Type</th>
<th>Column(s)</th>
<th>Format (Per Card)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option Number</td>
<td>1</td>
<td>1-5</td>
<td>Fixed Point on Card Type 1, right justified in field</td>
</tr>
<tr>
<td>Number of Variable names (maximum of 10 per input card)</td>
<td>1</td>
<td>6-10</td>
<td></td>
</tr>
<tr>
<td>NEWFMT (=0 or blank for no label card, =2 to input a label)</td>
<td>1</td>
<td>11-15</td>
<td>3I5</td>
</tr>
<tr>
<td>Card containing Hollerith format</td>
<td>2</td>
<td>1-80</td>
<td>If NEWFMT set equal to 2</td>
</tr>
<tr>
<td>Variable names left justified</td>
<td>4</td>
<td>1-8, 9-16, etc.</td>
<td>10A8</td>
</tr>
<tr>
<td>Format for Variable name input data</td>
<td>5</td>
<td>1-80</td>
<td>Any valid Fortran format, default = 10G8.4</td>
</tr>
<tr>
<td>Array name left justified</td>
<td>7</td>
<td>1-8</td>
<td></td>
</tr>
<tr>
<td>Location at which to start loading array (necessary only if loading does not start in location 1)</td>
<td>7</td>
<td>9-15</td>
<td>A8, I7</td>
</tr>
<tr>
<td>Array Name Format</td>
<td>8</td>
<td></td>
<td>Any valid Fortran format, default = 10G8.4</td>
</tr>
</tbody>
</table>
Card Input Details

A standard format for both variable name data and array name data is preset in the Read-in Subroutine. These formats may be used for inputting data without specifying a format in the input stream, i.e., leave out card types 5 and 8 if the preset format is large enough for the data to be input. These formats should be large enough for most applications and will save the user time in setting up input data. The preset format is the same for both variable name data and array name data and is (10G8.4) which permits 10 items to a card in fields of 8 columns each. Appendix B has a sample case with a listing of the input data.

The Example inputs illustrate how the various kinds of data are input and are in no particular sequence. Line 1 indicates option number, column 5, number of variable names, columns 9 and 10, and the number 2 in column 15 used to indicate that a Hollerith label will be input. Line 2 is the Hollerith label to be read in for this case. On Line 3 are the variable names to be read in while Line 4 contains the data associated with each name. Since the name format and data format are each 8 columns long, the data appears under its name, however this need not be the case if the data input format is altered. Lines 5, 6 and 7 show the use of an array name, in this case the plot array. A part of the pressure forcing function input is shown in Lines 8 through 18. Line 8 gives the number of times (or Mach Numbers depending on the option), angles-of-attack and roll angle points to be input. Line 9 contains the first time point. Line 10 contains the first alpha. Lines 11, 12, and 13 contain the roll angle-pressure points. Line 14 has the second angle-of-attack. Lines 15, 16, and 17 contain the set of roll angle-pressure points associated with that alpha. Line 18 begins the sequence again with the second time point. Note that this $C_p$ table is a non-symmetric or 360 degree table and as such would be so specified in the input stream even though it is not shown in this particular example which is abstracted from the total input shown on a later listing.

A7  I PLOT — Name of the input array in which the plotted values are specified.

The following values may be plotted:

<table>
<thead>
<tr>
<th>Index</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Time</td>
</tr>
<tr>
<td>2</td>
<td>Manifold Pressure</td>
</tr>
<tr>
<td>3</td>
<td>Port pressure 1</td>
</tr>
<tr>
<td>4</td>
<td>Port pressure 2</td>
</tr>
<tr>
<td>5</td>
<td>Port pressure 3</td>
</tr>
<tr>
<td>6</td>
<td>Port pressure 4</td>
</tr>
</tbody>
</table>
### MANIFOLD ANALYSIS OF AEROBEE FLT 17.05, 5/25/70

<table>
<thead>
<tr>
<th>YR</th>
<th>PM</th>
<th>TM</th>
<th>TINIT</th>
<th>TEND</th>
<th>BDUMP</th>
<th>PRNTFO</th>
<th>CASES</th>
<th>KT</th>
<th>NOPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1080</td>
<td>520</td>
<td>360.2</td>
<td>362</td>
<td>1</td>
<td>.01</td>
<td>2</td>
<td>3</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLOT</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>360.2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1048.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>1062.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>1183.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>120</td>
</tr>
<tr>
<td>240</td>
</tr>
<tr>
<td>360.4</td>
</tr>
</tbody>
</table>

etc.

Cp TABLE
To set up this plot option use the following procedure:

Set:

- **NCURVES** = Number of curves (maximum of 10).

- **PLTCON** = 1 to continue plot from one case to the next, i.e., continuation of related cases. However, if unrelated cases are run set **PLTCON** = 0.

- **I PLOT** (1) = independent variable.
  (2) = dependent variable No. 1 by number.
  (3) = dependent variable No. 2 by number.
  Etc. up to 10 dependent variables.

**Note**: **PLOTFQ**: If **PLOTFQ** is equal to **PRNTFO**, every printed point will also be plotted. However, by specifying **PLOTFQ** equal to an integer number n, only every nth point will be plotted. Notice that **PRNTFO** may be any non-integer but only if **PLOTFQ** = **PRNTFO** will every printed point be plotted.

### A8 Program Output

<table>
<thead>
<tr>
<th>Output Name</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>Altitude of vehicle—ft</td>
</tr>
<tr>
<td>Velocity</td>
<td>Velocity of vehicle—ft/sec</td>
</tr>
<tr>
<td>OUTPUT NAME</td>
<td>UNITS</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>Manifold Press</td>
<td>lbf/ft(^2)</td>
</tr>
<tr>
<td>Pressure at each port</td>
<td>lbf/ft(^2)</td>
</tr>
<tr>
<td>Atmospheric Pressure</td>
<td>lbf/ft(^2)</td>
</tr>
<tr>
<td>Q₀</td>
<td>Dynamic pressure—lbf/ft(^2)</td>
</tr>
<tr>
<td>Mach Number</td>
<td>Vehicle velocity/speed of sound at ALT</td>
</tr>
<tr>
<td>Alpha</td>
<td>Angle-of-attack—degrees</td>
</tr>
<tr>
<td>Manifold Temperature</td>
<td>Degrees R (Rankine)</td>
</tr>
<tr>
<td>Pressure coefficient at each port</td>
<td>Same as input</td>
</tr>
<tr>
<td>Roll rate</td>
<td>CPS</td>
</tr>
<tr>
<td>Mass flow rate (through each port)</td>
<td>lbf/sec</td>
</tr>
<tr>
<td>Reynolds Number (through each port)</td>
<td>Based on tube diameter</td>
</tr>
<tr>
<td>Internal and external ss</td>
<td>Speed of sound—ft/sec</td>
</tr>
<tr>
<td>Temperature (at each port)</td>
<td>Degrees R (T(_{oo}) or T(_r)) depending on input, i.e., if recovery factor is used</td>
</tr>
<tr>
<td>Phase angle at each port</td>
<td>In relation to free stream, degrees</td>
</tr>
<tr>
<td>Total mass change in manifold</td>
<td>lbfm from (t_0) to (t_{present}) (T(<em>{INIT}) to T(</em>{END}))</td>
</tr>
<tr>
<td>Velocity in each tube</td>
<td>Average gas velocity—ft/sec</td>
</tr>
<tr>
<td>Friction factor in each tube</td>
<td>Average friction factor in tube</td>
</tr>
</tbody>
</table>
APPENDIX B
SAMPLE PROBLEM

As a means of illustrating both the type of problem that can be solved and the input details, the following sample problem is presented.

Example 1 input —

Data needed for program:

Port sizes and tube lengths
Type of pressure forcing function
Port locations (fore and aft) needed only if using computed pressures at the ports or abstracting data from wind tunnel reports
Trajectory data

Option Selection —

<table>
<thead>
<tr>
<th>Option Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Tube flow</td>
</tr>
<tr>
<td>a</td>
<td>Trajectory input</td>
</tr>
<tr>
<td>i</td>
<td>Pressure = f (t, a, φ)</td>
</tr>
<tr>
<td>b</td>
<td>Atmospheric ambient pressure at each port</td>
</tr>
<tr>
<td>o</td>
<td>Output all input data</td>
</tr>
<tr>
<td>r</td>
<td>Input 360 degrees Cp Tables</td>
</tr>
<tr>
<td>p</td>
<td>Plot option for Port pressure 1, 2, 3, plus manifold and ambient pressure versus time</td>
</tr>
<tr>
<td>u</td>
<td>Save PM, TM and t to use in the next case</td>
</tr>
</tbody>
</table>

B-1
Option Implementation

<table>
<thead>
<tr>
<th>Option</th>
<th>Input Names (Variables)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Input</td>
<td>VR, PM, TM, TINIT, TEND, PRNTFQ, KT, CASES</td>
</tr>
<tr>
<td>1</td>
<td>PIPSIZ</td>
</tr>
<tr>
<td>a</td>
<td>ALT, VEL, TF, NOPTS</td>
</tr>
<tr>
<td>b</td>
<td>Default, no input needed</td>
</tr>
<tr>
<td>i, r</td>
<td>OPTSEL, CPCNTL, $C_p$ Table indices and table values, number of $C_p$ tables, ICP</td>
</tr>
<tr>
<td>o</td>
<td>BDUMP</td>
</tr>
<tr>
<td>p</td>
<td>IPLOT, NCURVS, MODE, IPSKIP, PLTCON</td>
</tr>
<tr>
<td>u</td>
<td>SAVE</td>
</tr>
</tbody>
</table>

The following data sheet is a listing of the program input for Example 1.
<table>
<thead>
<tr>
<th>COLUMN 0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANIFOLD ANALYSIS OF AERODYNE FLT 17.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1.08</td>
<td>1.52</td>
<td>3.60</td>
<td>2.36</td>
<td>1.0</td>
<td>.01</td>
<td>2.0</td>
<td>3.0</td>
</tr>
<tr>
<td>NGCURVES MODE : IPS, IC &gt; 1.5 1.0 1.2 1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.5</td>
<td>1.2</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPILOT</td>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALT</td>
<td>17000.9</td>
<td>16943.7</td>
<td>16878.9</td>
<td>16814.4</td>
<td>16749.9</td>
<td>16685.4</td>
<td>16621.2</td>
<td>16557.4</td>
</tr>
<tr>
<td>VEL</td>
<td>338.5</td>
<td>338.5</td>
<td>338.5</td>
<td>338.5</td>
<td>338.5</td>
<td>338.5</td>
<td>338.5</td>
<td>338.5</td>
</tr>
<tr>
<td>TF</td>
<td>360.2</td>
<td>360.2</td>
<td>360.2</td>
<td>360.2</td>
<td>360.2</td>
<td>360.2</td>
<td>360.2</td>
<td>360.2</td>
</tr>
<tr>
<td>PIPSIZ</td>
<td>.0625</td>
<td>.0625</td>
<td>.0625</td>
<td>.0625</td>
<td>.0625</td>
<td>.0625</td>
<td>.0625</td>
<td>.0625</td>
</tr>
</tbody>
</table>

Reproduced from best available copy.
<table>
<thead>
<tr>
<th>Time (h)</th>
<th>0.0</th>
<th>120.0</th>
<th>240.0</th>
<th>361.2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>109.2</td>
<td>1071.4</td>
<td>1111.4</td>
<td>1079.0</td>
</tr>
<tr>
<td>120.0</td>
<td>1084.2</td>
<td>1084.2</td>
<td>1090.2</td>
<td>1078.6</td>
</tr>
<tr>
<td>240.0</td>
<td>1079.0</td>
<td>1084.2</td>
<td>1090.2</td>
<td>1118.3</td>
</tr>
<tr>
<td>361.2</td>
<td>1109.2</td>
<td>1071.4</td>
<td>1111.4</td>
<td>1098.7</td>
</tr>
<tr>
<td>0.0</td>
<td>1109.2</td>
<td>1071.4</td>
<td>1111.4</td>
<td>1165.2</td>
</tr>
<tr>
<td>120.0</td>
<td>1079.0</td>
<td>1084.2</td>
<td>1090.2</td>
<td>1183.5</td>
</tr>
<tr>
<td>240.0</td>
<td>1078.6</td>
<td>1084.2</td>
<td>1090.2</td>
<td>1118.3</td>
</tr>
<tr>
<td>361.4</td>
<td>1078.8</td>
<td>1084.2</td>
<td>1090.2</td>
<td>1098.7</td>
</tr>
<tr>
<td>0.0</td>
<td>1048.6</td>
<td>1165.2</td>
<td>1111.4</td>
<td>1117.4</td>
</tr>
<tr>
<td>120.0</td>
<td>1062.9</td>
<td>962.8</td>
<td>1111.4</td>
<td>1057.2</td>
</tr>
<tr>
<td>240.0</td>
<td>983.5</td>
<td>1062.9</td>
<td>962.8</td>
<td>1057.2</td>
</tr>
<tr>
<td>361.6</td>
<td>983.5</td>
<td>1062.9</td>
<td>962.8</td>
<td>1097.0</td>
</tr>
<tr>
<td>0.0</td>
<td>1057.2</td>
<td>1097.0</td>
<td>1031.5</td>
<td>1031.5</td>
</tr>
<tr>
<td>120.0</td>
<td>1057.2</td>
<td>1097.0</td>
<td>1031.5</td>
<td>1031.5</td>
</tr>
<tr>
<td>240.0</td>
<td>1078.8</td>
<td>1032.9</td>
<td>1179.4</td>
<td>1032.9</td>
</tr>
<tr>
<td>362.0</td>
<td>1078.8</td>
<td>1032.9</td>
<td>1179.4</td>
<td>1179.4</td>
</tr>
</tbody>
</table>

B-4
<table>
<thead>
<tr>
<th>TEND</th>
<th>364.10</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>1087.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1075.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1226.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>1087.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1075.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1226.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.</td>
<td>1074.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1071.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1204.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>1074.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1071.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1204.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.</td>
<td>1104.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1105.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1226.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>1109.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1105.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1226.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.</td>
<td>1074.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1097.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1200.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>1074.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1097.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1200.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.</td>
<td>1109.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1105.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1204.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>363.2</td>
<td>1078.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1050.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1192.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>1078.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120.</td>
<td>1050.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240.</td>
<td>1192.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>363.4</td>
<td>1117.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Angle (°)</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>120</td>
<td>589.5</td>
<td>1234.7</td>
<td>1117.9</td>
</tr>
<tr>
<td>240</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>360</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C

The numerical integration scheme used to solve the equations in this report is a sixth order predictor corrector with a Runga-Kutta starter. Step size is automatically computed and altered based on both the stability and truncation error. This particular predictor-corrector set was chosen rather arbitrarily during program design and is still used because it has never failed to work properly for the manifold program. The particular numerical integration program used was designed to handle up to fifteen differential equations simultaneously with added facilities for changing predictor-corrector pairs and error criteria easily as well as allowing utility in inputting the differential equations.

Documentation of the above mentioned numerical integration scheme is now in progress.

The predictor-corrector pair used is the following:

\[ Y_{n+1}^P = Y_n + \frac{\Delta t}{24} \left[ 55 \dot{Y}_n - 59 \ddot{Y}(n-1) + 37 \dddot{Y}(n-2) 
- 9 \dddot{Y}(n-3) \right] + \frac{251}{720} \Delta t^5 \dddot{Y}^{IV} \]

\[ Y_{n+1}^C = Y_n + \frac{\Delta t}{24} \left[ 9 \dot{Y}(n+1) + 19 \dot{Y}(n) - 5 \dddot{Y}(n-1)
+ \dot{Y}(n-2) \right] - \frac{19}{720} \Delta t^5 \dddot{Y}^{IV} \]

where \( \dddot{Y}^{IV} \) is the fifth derivative of \( Y \).

The four point Runga-Kutta formula used for starting the integration follows:

\[ K_1 = \Delta t \times f(x, y) \]

\[ K_2 = \Delta t \times f \left( x + \frac{\Delta t}{2}, y + \frac{K_1}{2} \right) \]

\[ K_3 = \Delta t \times f \left( x + \frac{\Delta t}{2}, y + \frac{K_2}{2} \right) \]

\[ K_4 = \Delta t \times f \left( x + \Delta t, y + K_3 \right) \]

C-1
\[ Y = \frac{1}{6}(K_1 + 2K_2 + 2K_3 \cdot K_4) \]

\[ Y_2 = Y_1 + \Delta Y \]
REAL*8 DICT(300)

COMMON/ZILCH1/YX(15), YPRIME(15), ERCRIT(2), P(4), RAD(4), AL(4),
ALT(50), VEL(50), TF(50), CQ(10), FRENQO(10), CMACH(10), ALFA(10, 15),
2ALPHA(10), TALPH(10), PRESS(6), TCON, TTUBE, THICK, CPTUBE, ROTUBE,
3FI(10,15,20), CPP(10,15,20), T, DELT, DELMAX, PM, VR, THETA, OMEGA,
4THRCON, PLTCON, FAZANG, ALPHA, CP, AA, TM, PRNTFQ, TEND, PHI, RT, INDVR,
5JMAX, NONLIN, IERCRT, IXMAX, IYMAX, IZMAX, KT, NOQ, NOPTS, NCQ, NALPHA,
6NCURVS, PLOTFO, IPLOT(15), APLOT(15)

C THE '8' ARRAY CONTAINS ALL INPUT QUANTITIES FROM READIN

C THE NAME BLANKS IS A FILLER IN VACENT LOCATIONS THAT
C MAY BE USED FOR NEW VARIABLE NAMES AS NEEDED

DATA DICT/50*'ALT', 10*'CQ', 10*'FRENO', 'NCQ', 'PRNTFQ', 'BDUMP',
1'TINIT', 'TEND', 'NOPTS', 'ALPHA', 'CASES', 'VR',
2'OMEGA', 'PM', 'THRCON', 'PLTCON', 'BLANKS', 'RO', 'TM', 'INDVR',
3'BLANKS', 'NCURVS', 'BLANKS', 'PLTCON', '7*BLANKS', 'OPTSEL',
4'TCPCNTL', 'TCON', 'TTUBE', 'THICK', 'CPTUBE', 'ROTUBE', '5*BLANKS',
5'TDEL', 'DELMAX', 'ERCRIT', 'JMAX', 'NCLIN', 'IERCRT', 'ICP', 'KT',
6'SAVE', '10*ALPHA1', '10*TALPH', 'RT', '4*BLANKS',
750*'VEL', 50*'TF', 8*'PIPSIZ', 15*'IPLOT', 27*'BLANKS'/

NDICT=300

DO 12 I=1, 300
12 B(I)=0.

200 CALL READIN(DICT, B, NDICT)

5555 CONTINUE

NCQ=B(71)
PRNTFQ=B(72)
BDUMP=B(73)

C T IS THE INITIALIZATION TIME FOR THE INTEGRATION

T=TINIT = INITIAL TIME

T=B(74)
TEND=B(75)

C NOPTS = THE NUMBER OF ALT-VEL TF DATA POINTS INPUT TO THE PROG

NOPTS=B(76)
ALPHA=B(77)
ICASES=B(78)
VR=B(79)
OMEGA=B(80)
PM=B(81)

C THERMAL CONSTANT K IN BTU/SEC*FT*DEG R

THRCON=B(82)

C PLTCON = 1 WILL ENABLE THE USER TO CONTINUE PLOTTING FROM
C ONE CASE TO THE NEXT, WHEN THE LAST CASE TO BE PLOTTED IS
C REACHED SET PLTCON=0 OTHERWISE PLTCON IS SET = 0 AUTOMATICALLY
C WHEN THE LAST CASE IS INPUT

PLTCON=B(83)

IF(ICASES .LE. 1) PLTCON=0.

B(83)=PLTCON

C OPTION SELECTION, 0=TUBE FLOW, 1=TUBE FLOW+HEAT, 2=ORIFICE FLOW

C IF NOQ IS NOT INPUT THE TUBE OPTION IS CHOSEN

NOQ=B(85)

C YX(1) IS THE INITIAL DENSITY - DERIVED FROM INITIAL TM & PM IN
C SUBROUTINE PRESSUR UNLESS DENSITY IS INPUT, IE DENSITY IS

D-1
NONSTANDARD AND TAKES PRECEDENCE IF BOTH TEMP AND DENSITY ARE
INPUT BY MISTAKE. PRESS MUST ALWAYS BE INPUT

YX(1)=B(86)
YX(2)=B(87)

IF( YX(2).LT. 0.) WRITE(6,1000)
1000 FORMAT("0 THE ISOHERMAL SOLUTION TO THE EQNS HAS BEEN CHOSEN")
INDPVR=B(88)

INDPVR IS A TRIGGER INDICATING THE TYPE OF INDEPENDENT VARIABLE
BEING USED FOR THE CQ TABLE, IE, -1,RENO, 0(DEFAULT) SQRT(RE),
1 PRESSURF RAT. .GT. 1.0
IF(INDPVR .NE. 0) WRITE(6,100)INDPVR
100 FORMAT("O A NONSTANDARD INDEPENDENT TABLE INPUT HAS BEEN USED FOR
CQ, IND =",12)

SET UP VARIABLES FOR PLOTTING
NPPTS=B(89)
NCURVS=B(90)

DEFAULT NPPTS
IF(NPPTS .EQ. 0 .AND. NCURVS .GT. C) NPPTS=100
MODE=B(91)
PLOTFQ=B(92)
DO 205 IX=1,15
APLOT(IX)=B(258+IX)
205 IPLOT(IX)=B(258+IX)
C NOTE THAT PRESSF(566) ARE INPUT AS VARIABLE NAMES (OPTSEL & CPCNTL) IN
C READIN BUT IS HANDLED EVERYWHERE ELSE AS PRESSF(566)
C THE NAME INPUT, CPCNTL, IS USED TO MAKE INPUT EASIER
DO 206 IX=1,6
206 PRESSF(IX)=B(99+IX)
IF(PRESSF(5) .EQ. -1.) WRITE(6,110)
110 FORMAT("0 THE SPECIAL USE OPTICK HAS BEEN CHOSEN, IE, P(K)=PO*CP")
IF(PRESSF(5) .EQ. -2.) WRITE(6,113)
113 FORMAT("0 THE SPECIAL USE OPTION HAS BEEN CHOSEN, IE, P(K)=CP")
IF(PRESSF(5) .EQ. -3.) WRITE(6,111)
111 FORMAT("0 THE SPECIAL USE OPTION HAS BEEN CHOSEN, IE, P(K)=F(TF)")
IF(PRESSF(6) .EQ. -1. .OR. PRESSF(6) .EQ. 2.) WRITE(6,112)
112 FORMAT("0 THE SPECIAL USE OPTION OF CP AS A FUNCTION OF TIME HAS
BEEH CHOSEN, IE, CP=F(T,ALPHA,PHI)")

FOR THE SPECIAL USE OPTIONS THE ATMOSPHERIC VARIABLES ARE INPUT
AS CONSTANTS IN THE PRESSF ARRAY AS FOLLOWS
RHOO=PRESSF(1),PO=PRESSF(2),TO=PRESSF(3),AB=PRESSF(4)=SOUND SPD
C
THIS DEFINES THE ATMOSPHERIC QUANTITIES WHEN THEIR CALCULATION
C IS PRECLUDED BY THE SPECIAL USE CPTICNS
C PRESSF(6)=-1,0., FOR SYMMETRIC CP TABLES AS F(T),F(MACH NO),
C PRESSF(6)=1,2., FOR NON-SYMMETRIC CP TABLES,CP=F(MACH),CP=F(T),
C SYMMETRIC TABLE = 180 DEG , NONSYMMETRIC =360 DEG INPUT TABLES
TCON =B(106)
TTUBE=B(107)
THICK=B(108)/12.
CPTUBE=B(109)
ROTTUBE=B(110)

B ARRAY LOCATIONS 111-115 INCLUSIVE ARE AVAILABLE FOR USE
DELT=B(116)
C
DEFAULT DELT IF IT IS NOT SPECIFIED
IF(DELT .EQ. 0.) DELT=.0625
C
IF HEAT OPTION IS CHOSEN DELMAX WILL BE SET EQUAL TO .001
UNLESS DELMAX IS ALREADY SET LESS THAN .001, IE,
IF(NOQ .EQ. 1 .AND.(B(117) .GT. .001 .CR.B(117).EQ.0.))B(117)=.001
DELMAX=B(117)
ERCRIT(1)=B(118)
ERCRIT(2)=B(119)
JMAX=B(120)

C DEFAULT JMAX, NOTE JMAX IS NEVER INPUT UNLESS THE NO OF
C EQUATIONS IS CHANGED
IF(JMAX .EQ. 0) JMAX=2
IF(B(120) .NE. JMAX)B(120)=JMAX
NCNLIN=B(121)
IERCRT=B(122)

C ICP DENOTES THE NUMBER OF CP TABLES TO BE USED DURING A RUN
ICP=B(123)
C KT IS THE NUMBER OF PORTS ENTERING THE MANIFOLD
KT=B(124)

C DEFAULT KT=1
IF(KT .EQ. 0) KT=1
IF(B(124) .NE. KT) B(124)=KT

SAVE=1 SAVES PM, PM, T FROM PRESENT CASE TO BE USED IN NEXT CASE
SAVE MUST BE RESET EACH TIME IT IS TO BE USED
SAVE=B(125)

IF(NoQ .LE. 1) WRITE (6,105)
IF(NoQ .GE. 2) WRITE (6,106)

105 FORMAT('0 TUBE FLOW HAS BEEN CHOSEN, NOQ=',I3)
106 FORMAT('0 ORIFICE FLOW HAS BEEN CHCSEN, NOQ=',I3)

C LOAD ALT, VEL, TIME ARRAYS
DO 1, I=1,50
ALT(I)=B(I)
VEL(I)=B(I+150)
1 TF(I)=B(I+200)
IF(ALT(I) .NE. 0. .OR. VEL(I) .NE. 0. .OR. TF(I) .NE. 0.)AND.
1NQTS .EQ. 0.) WRITE(6,225)

225 FORMAT('O THE TRAJECTORY HAS BEEN INPUT BUT NOPTS IS STILL REQ')

C LOAD ORIFICE COEFF ARRAY AS A FUNCTION OF RENO, IE, FRENO
C LOAD ANGLE OF ATTACK - TIME TABLE
DO 5, I=1,10
CQ(I)=B(I+50)
FRENO(I)=B(I+60)
ALPHA(I)=B(I+125)
5 TALPH(I)=B(I+135)
IF(ALPHA(I) .NE. 0. .OR. TALPH(I) .NE. 0. .AND. ALPHA .EQ. 0.)
1WRITE(6,226)

226 FORMAT('O ALPHAA OR TALPH HAS BEEN INPUT BUT ALPHA IS STILL REQ')
IF(CQ(I) .NE. 0. .OR. FRENO(I) .NE. 0. .AND. NCC .EQ. 0.)WRITE(6,
1107)

107 FORMAT('O ORIFICE COEFF HAVE BEEN LOADED BUT NCC (COUNT OF NCQ''S)
1HAS NOT BEEN LOADED')

IF(BDUMP .EQ. 1.) WRITE(6,10001)(R(I),I=1,300)

10001 FORMAT('H0,10G13.6)

C RT GT 0 IMPLEMENTS THE RECOVERY TEMP OPTION ,RT IS THE RECOVERY
C TEMPERATURE FACTOR ,USUALLY ABOUT .9
RT=B(146)
WRITE(6,906)

906 FORMAT('H0, 'INPUT DIM, VR IN IN., OMEGA IN CPS, RAD(K) IN IN.,
1AL(K) IN IN., ALL OTHERS ARE IN FT, LBF, LBM, SEC')
IF(BDUMP .EQ. 0) WRITE(6,902)VR,OMEGA,PM, YX(2),TINIT

902 FORMAT('H0, 'INITIAL CONDITIONS 'VR=',G13.6,'OMEGA=',G13.6,'PM=',
1 G13.6,'TF=',G13.6,'TINIT=',G13.6)

C CP TABLES INPUT HERE
IF(ICP .EQ. 0) GO TO 921
ICP=ICP-1
B(123)=ICP
READ(5,950) IXMAX, IYMAX, IZMAX
950  FORMAT(4I5)
DO 920 IX=1,IXMAX
READ(5,915) CMACH(IX)
WRITE(6,904) CMACH(IX)
904  FORMAT(1HO,8G10.5)
915  FORMAT(E15.8)
DO 920 IY=1,IYMAX
READ(5,916) (ALFA(IX,IY),(FI(IX,IY,IZ),CPP(IX,IY,IZ),IZ=1,IZMAX))
WRITE(6,916) (ALFA(IX,IY),(FI(IX,IY,IZ),CPP(IX,IY,IZ),IZ=1,IZMAX))
916  FORMAT(E15.8/(2E15.8))
920  CONTINUE
921  CONTINUE
C  PIPE SIZE (PIPSIZ ARRAY) INPUT FROM B(250-258)
C  PIPSIZ DOES NOT APPEAR AS AN ARRAY NAME IN THE PROGRAM
C  IT IS ONLY A MEMNCNIC FOR DATA INPUT
I=1
DO 305 K=2,8,2
RAD(I)=B(249+K)
AL(I)=B(250+K)
I=I+1
305  CONTINUE
VR=VR/1728.
DO 922 II=1,4
RAD(II)=RAD(II)/12.
AL(II)=AL(II)/12.
922  CONTINUE
C  INITIALIZE PHI HERE IN CASE A NON STANDARD OPTION IS USED
C  PHI IS USED IN CALCULATING THE RCLL ANGLE IN CSUBP
PHI=0.
14  CONTINUE
C  INITIALIZE MANIFOLD SUBROUTINE HERE
C  ENTER CALCULATION PHASE OF MANIFOLD PROGRAM
CALL PRESUR
IF(SAVE .NE. 1.) GO TO 55552
B(81)=PM
B(87)=TM
B(74)=T
SAVE=0.
55552 CONTINUE
ICASES=ICASES-1
B(78)=ICASES
IF(ICASES .GT. 0) GO TO 2000
IF(ICASES .LE. 0 .AND. ICP.GF.1) GO TO 55551
STOP
END
SUBROUTINE PRESUR
COMMON/ZILCH1/YX(15),YPRIME(15),ERCRT(2),P(4),RAD(4),AL(4),
1ALT(50),VEL(50),T5(50),CQ(10),FRENO(10),CMACH(10),ALFA(10,15),
2ALPHAA(10),TALPH(10),PRESSF(6),TCON,TTUBE,THICK,CPTUBE,ROTUBE,
3FI(10,15,20),CPP(10,15,20),T,DELT,DELMAX,PM,VR,THETA,OMEGA,
4THCON,PLTCN,FAZANG,ALPHA,CP,AA,TM,PRNTFO,TEND,PHI,RT,INDPVR,
5SMAX,NONLIN,ERCRT,IXMAX,IZMAX,IZMAX,KT,NOQ,NOPTS,NCQ,NALPHA,
6NCURVS,PLOTFO,PLOT(15),APLOT(15)
INTEGER ONE/I/,TWO/2/,I/58/

DP/DT IN THIS PROG IS BASED ON ENERGY AND HEAT TRANS EQNS
DIMENSION ANS(4),DM(4),TA(4),RHC(4),C(15),CCW(4),CPW(4),REW(4)
DIMENSION PHI(4),PLOT(22),VVOUT(4),QAIR(4)
DIMENSION TUBE(4),TCONV(4),AVOL(4),ALAT(4),TVOL(t)
DATA ISKIP/O/

C INITIALIZE CONSTANTS FOR CALCULATION
C DEFAULT FOR ORIFICE EQNS IF TUBE DIMENSIONS ARE ZERO
IF( AL/I) .EQ. O. .AND. NOQ .LT. 2) NOQ=2
C INITIALIZE V2 TO DETERMINE CPTICN SELECTION
V2=O.
C INITIALIZE PLOT ROUTINE HERE
IF(NCURVS .GT. O. .AND. ISKIP .EQ. O) CALL RJPLOT(APLOT,NCURVS)
C IF PLTCN IS USED FOR THIS CASE SKIP PLOT REINITIALIZING
IF(PLTCN .EQ. 1.) ISKIP=1
IF(PLTCN .EQ. 0.) ISKIP=O
IPSS=O
PI=3.14159
GAMMA=1.4
VV=O.
AJ=778.26

C OMEG=ROLL RATE IN CPS
C SAVE OMEGA IN CPS THEN CONVERT TO RAD/SEC
OMEGA=OMEGA
OMEGA=OMEGA*6.28318
GC=32.174048

C R IN (FT-LBF)/(LBM- DEG R)
R=53.36
CPC=.240
CV=.1710
C INITIALIZE TOTMAS WHICH IS THE TOTAL MASS CHANGE IN THE MANIFOLD
TOTMAS=O.
RR=((.75*VR)/PI)**(.3333)
T=ABS(YX(2))
C TISO IS THE TRIGGER FOR KEEPING THE TEMP CONST ,IE ISOTHERMAL
TISO=YX(2)
C MAKE SURE YX(2) IS POSITIVE FOR INTEGRATOR
YX(2)=ABS(YX(2))
RO=YX(1)
C IF YX(1) NE O USE DENSITY INPUT OPTION
IF(YX(1) .EQ. O.) GO TO 70
TM=PM/(R*RO)
YX(2)=TM
GO TO 75
70 TM=PM/(R*TM)
YX(1)=RO
75 CONTINUE
C ZERO OUT PORT VARIABLES
DO 11111 K=1,4
P(K)=O.
TA(K)=O.
CPW(K)=O.
DECLARE DM(K) = 0.
DECLARE REW(K) = 0.
DECLARE PHIII(K) = 0.
DECLARE VVCUT(K) = 0.
DECLARE CWIK(K) = 0.

C INITIALIZE INITIAL HEAT TRANSFER RATE AND CALCULATE CONSTANTS

DECLARE QAIR(K) = 0.
DECLARE TUBE(K) = TTUBE
DECLARE TCCNV(K) = TCON
DECLARE AVOL(K) = PI * RAD(K) * AL(K)
DECLARE ALAT(K) = 2. * PI * RAD(K) * AL(K)
DECLARE TVCL(K) = PI * ((RAD(K) + THICK) * AL(K) - AVOL(K)
WRITE(6, 10109) AVOL(K), ALAT(K), TVCL(K)

11111 RHO(K) = 0.

C SET V2 = PRESSF(5) IF PRESSF(5) LT 0
IF(PRESSF(5) LT 0.0) V2 = PRESSF(5)
C INPUT ATMOSPHERIC VARIABLES HERE IF V2 IS NEGATIVE
IF(V2 LE 0.0) GO TO 81
RHO = PRESSF(1)
P0 = PRESSF(2)
TO = PRESSF(3)
AB = PRESSF(4)
AA = 0.
H0 = 0.
V1 = C.
Q0 = 0.
CP = 0.

81 C CONTINUE
C FAZANG = ANGLE BETWEEN KT EVENLY SPACED PORTS
FAZANG = (2. * PI) / KT
C DISCHARGE COEFFICIENT FOR AN ORIFICE - CONSTANT
CGD = .611
C INITIALIZE INTEGRATION SUBCRUTINE HERE
C LIMIT INTEGRATION STEP SIZE TO OUTPUT FREQUENCY
IF(DELMAX .GT. PRNTFQ) DELMAX = PRNTFQ
CALL STARTI (XY, YPRIME, T, DELT, DELMAX, ERcrit, JMAX, NONLIN, IERcrit)
C ALPHA INITIALLY IS THE NO OF POINTS IN THE ALPHAA-TALPH
C TABLE AND IS USED THEREAFTER TO TRANSFER ALPHA TO CSUBP
C IALF IS THE NO OF POINTS IN ALPHAA TABLE
IALF = ALPHA
C THE DERIVATIVES RCD AND TMD ARE CALCULATED IN PCALC
IS = 0
III = 1
IORDER = 3
C PLOTF IS INITIALIZED HERE
PLOTF = 0.
35 C CONTINUE
C TPRNT IS THE TIME AT THE LAST OUTPUT STEP AND IS SAVED TO
C COMPARE WITH PRNTFQ (PRINT FREQUENCY)
TPRNT = T
21 C CONTINUE
C FIND ALT & VEL AT TIME T
IF(IALF .EQ. 0) GO TO 45
C NOPRNT = 3 SUPPRESSES PRINTED OUTPUT FROM SINTRP WHEN TABLE
C BOUNDARIES ARE EXCEEDED, SINTRP Chooses END POINTS
NOPRNT = 3
CALL SINTRP(T, ALPHAA, TALPH, IALF, IS, III, IORDER, ALPHA, NOPRNT)
45 C CONTINUE
NOPRNT = 3
IF(NOPTS .EQ. 0) GO TO 80

D-6
CALL SINTRP(T, ALT, TF, NOPTS, IS, IORDER, HO, NCPRT)
NOPRT=3
IF(PRESSF(5) .EQ. -3.) GC TO 80
C SKIP AT62 AND VEL INTERPOLATION WHEN 2ND SPECIAL USE OPTION IS USED
CALL SINTRP(T, VEL, TF, NOPTS, IS, IORDER, VI, NCPRT)
CALL AT62(HO, ANS)
RHOO=ANS(1)*GC
PO=ANS(2)
TO=ANS(3)*1.80007
AB=ANS(4)
AA=ABS(VI)/AB
QQ=(RHOO*VI*VI*.5)/GC
80 CCNTINUE
30 CCNTINUE
C BEGIN CALCULATION OF DERIVATIVES ROD(RO DOT) & TMD(TM DOT)
THETA=0.
DMTOT=0.
SUM=0.
SUMP=0.
PM=R*RO*TM
C INITIALIZE HEAT TRANS TERMS BEFORE LOOP
QTCT=0.
DTA=0.
DTT=0.
10003 DO 500 K=1,KT
C V2=-1, P(K)=PO*CP, V2=-2, P(K)=CP, V2=-3, P(K)=F(TF) WITH
ICONS=2
COUNT=0.
IF(V2 .GE.-2.) CALL CSUBP
P(K)=PO+CP*QQ
C IF(PRESSF(5) LT 0) PORT PRESSURE IS ALTERED FOR SPECIAL USE OPTION
C THE FORCING FUNCTION INPUT INTO THE ALTITUDE ARRAY
IF(V2 .EQ. -1.) P(K)=PO*CP
IF(V2 .EQ. -2.) P(K)=CP
IF(V2 .EQ. -3.), P(K)=HO
C IF P(K) GOES TO 0. THE DENSITY ECN WILL BE DIVICING BY 0. ALSO
C IF(P(K) .LT. 0.) P(K)=0.
C RT IS THE RECOVERY TEMPERATURE FACTOR
1020 TA(K)=TO*(1.+2.*RT*AA*AA)
RHO(K)=P(K)/(R*TA(K))
C TE IS THE TEMP USED TO CALC THE ENERGY TERM OF DT/DT(K)
C USE TEMP OF SOURCE GAS FOR TF
TE=TA(K)
IF(P(K) .LT. PM) TE=TM
C USE AVG DENSITY FOR APPROXIMATE COMPRESSIBLE ADIABATIC
C FLOW WITH FRICTION THROUGH A PIPE, IE ASSUMED LINEAR VARIATION
C USE DENSITY FROM DIRECTION OF FLOW FOR ORIFICE FLOW
C CALCULATE THE SPEED OF SOUND -SS (FT/SEC) BASED ON THE
C TEMPERATURE FROM THE DIRECTION OF FLOW
SS=SRT(GAMMA*GC*R*TE)
C CALCULATE VISCOSITY BASED ON TE
AMuu= 2.22E-8*(TE**3/(1.+(180./TE)))
C IF NOQ= 2 OR 3 USE ORIFICE FLOW EONS 2 NO HEAT ,
C IF(NQQ .LT. 2 .OR. NOQ .GT. 3) GO TO 601
ROAVG=RHO(K)
IF(P(K) .LT. PM) ROAVG=RO
DMSAVE=0.
C ROAVGS=DENSITY IN SLUGS ,CQD = DISCHARGE COEFFICIENT
C ROAVG =DENSITY IN LB/FT**3 D(K) IN LB/SEC
ORIFICE FLOW EQUATIONS

\[ \text{ROAVG} = \frac{\text{ROAVG}}{\text{GC}} \]

\[ \text{CONTINUE} \]

\[ \text{DM}(K) = \text{CQD} \cdot \pi \cdot \text{RAD}(K)^2 \cdot \frac{\text{ROAVG} \cdot \sqrt{2 \cdot \text{ABS}(\text{P}(K) - \text{PM})}}{\text{ROAVGS}} \]

\[ \text{IF}(\text{P}(K) \cdot \text{LT.} \cdot \text{PM}) \quad \text{DM}(K) = -\text{DM}(K) \]

\[ \text{VTRIL} = \frac{\text{ABS}(\text{VV})}{\sqrt{\text{DM}(K) / (\text{ROAVG} \cdot \pi \cdot \text{RAD}(K)^2)}} \]

\[ \text{RE} = \frac{\text{ROAVG} \cdot \text{ABS}(\text{VV}) \cdot 2 \cdot \text{RAD}(K)}{\text{AMUU} \cdot \text{GC}} \]

\[ \text{IF}(\text{VV} \cdot \text{GT.} \cdot \text{SS} \cdot \text{AND.} \cdot \text{NCQ} \cdot \text{EQ.} \cdot 0) \quad \text{GO TO 600} \]

\[ \text{IF}(\text{VV} \cdot \text{NE.} \cdot 0.) \quad \text{GO TO 50} \]

\[ \text{VV} = 1. \]

\[ \text{DM}(K) = 0. \]

\[ \text{GO TO 1060} \]

\[ \text{GO TO 1060} \]

\[ \text{ITERATE TO FIND DM(K) IF CQD IS VARIABLE} \]

\[ \text{VTWST} = \text{ABS}(\text{VV}/\text{VTRIL}) \]

\[ \text{IF}(\text{ABS}(\text{VTWST} - 1.) \cdot \text{LE.} \cdot 01 \cdot \text{AND.} \cdot \text{VV} \cdot \text{GT.} \cdot \text{SS}) \quad \text{GO TO 600} \]

\[ \text{IF}(\text{ABS}(\text{VTWST} - 1.) \cdot \text{LE.} \cdot 01) \quad \text{GO TO 1060} \]

\[ \text{THE DISCHARGE COEFF TABLE IS CQD VS (RE)**.5} \]

\[ \text{CHECK TO FIND WHICH INDEP VAR IS USED FOR CQ} \]

\[ \text{IF(INDPVR)76,77,78} \]

\[ \text{RS} = \text{RE} \]

\[ \text{GO TO 79} \]

\[ \text{RS} = \frac{\text{P}(K)}{\text{PM}} \]

\[ \text{IF}(\text{PM} \cdot \text{GT.} \cdot \text{P}(K)) \quad \text{RS} = \text{PM} / \text{P}(K) \]

\[ \text{GO TO 79} \]

\[ \text{RS} = \sqrt{\text{RE}} \]

\[ \text{CONTINUE} \]

\[ \text{NOPRNT} = 3 \]

\[ \text{CALL SINTRP}(\text{RS}, \text{CQ}, \text{FREQN}, \text{NCQ}, \text{IS}, \text{III}, \text{IORDER}, \text{CQD}, \text{NOPRNT}) \]

\[ \text{DMSAVE} = \text{DM}(K) \]

\[ \text{GO TO 40} \]

\[ \text{GO TO 40} \]

\[ \text{ROAVG} = (\text{RO} + \text{RHO}(K)) \cdot 0.5 \]

\[ \text{VTRIL} = \text{ABS}(\text{VV}) \]

\[ \text{RE} = \frac{\text{ROAVG} \cdot \sqrt{2 \cdot \text{RAD}(K)}}{\text{AMUU} \cdot \text{GC}} \]

\[ \text{FRICITION COEFFICIENT FOR SMCCTH PIPES} \]

\[ \text{ALLOW CHANGE OF RE RANGE UP TO 10 STEPS} \]

\[ \text{IF}(\text{CPOINT} \cdot \text{LE.} \cdot 10) \quad \text{GO TO 704} \]

\[ \text{FORCE CALC TO REMAIN IN ONE RE RANGE} \]

\[ \text{IF(INCONS -2)702,700,701} \]

\[ \text{IF}(\text{RE} \cdot \text{LT.} \cdot 100000. \cdot \text{AND.} \cdot \text{RE} \cdot \text{GT.} \cdot 1185.) \quad \text{GO TO 700} \]

\[ \text{IF}(\text{RE} \cdot \text{LE.} \cdot 1185.) \quad \text{GO TO 701} \]

\[ \text{NIKURADSE EQN} \]

\[ \text{FF} = (1.0082 \cdot 0.0525 / \text{RE}^{2.237}) \]

\[ \text{ICCONS} = 1 \]

\[ \text{GO TO 705} \]

\[ \text{EQU FOR BLASIUS RANGE} \]

\[ \text{FF} = 0.0791 / \text{RE}^{.25} \]

\[ \text{ICCONS} = 2 \]

\[ \text{GO TO 705} \]

\[ \text{DD = 16. / \text{RE}} \]

\[ \text{ICCONS} = 3 \]

\[ \text{CONTINUE} \]

\[ \text{LOVRD} = \frac{\text{AL}(K)}{\text{RAD}(K)} \]

\[ \text{CALCULATE (1/\text{RO} - 1/\text{RHO}) FOR COMPRESSIBLE FLOW TUBE FLOW EQN} \]

\[ \text{DROCOMP} = \text{ABS}((1 / \text{RHO}(K)) - (1 / \text{RO})) \]

\[ \text{COMPRESSIBLE FRICTION FLOW EQN INVERTED TO SOLVE FOR A} \]

\[ \text{VELOCITY DENSITY-COUPLE USED TO CALC MASS FLOW RATE} \]
VV = SORT(((ABS(P(K) - PM) / (FF * ROAVG * LOVRD + ROAVG * ROCOMP)) * GC)
COUNT = COUNT + 1.
C CQD REPRESENTS THE FRICTION FACTOR FF WHEN TUBE FLOW IS USED
C
CQD = FF
IF (VV .NE. 0.) GO TO 1050
VV = 1.
DM(K) = 0.
GO TO 1060
1050 IF (ABS(VV/VTRIL-1.) -.01) 600, 600, 601
600 IF (VV-.SS) 630, 630, 640
640 VV = SS
C CQD FOR M=1 IS ONLY APPROXIMATED BY .611 IT SHOULD BE
INVESTIGATED IF ANY AMOUNT OF CHOKE ORIFICE FLOW IS ANTICIPATEC
IF (NOQ .EQ. 2 .OR. NOO .EQ. 3) VV = AB*CQD
630 IF (P(K) - PM) 602, 603, 633
602 VV = VV
603 DM(K) = ROAVG*PI*RAD(K)**2*VV
1060 DMTOT = DMTOT + DM(K)
SUM = SUM + DM(K)*(CCP* TE)
CQW(K) = CQD
CPW(K) = CP
REW(K) = RE
PHII(K) = PHI
VVCUT(K) = VV
C CALCULATE APPROXIMATION TO HFAT TRANSFER FOR PIPE FLOW
C ALAT = LATERAL AREA OF TUBE, QAIR = BTU/SEC, TCONV = AVG TEMP OF AIR IN
C TUBE, TUBE = AVG TUBE TEMP
AVOL = VOL OF AIR IN EACH TUBE, TVOL = VOL OF TUBE MATERIAL
IF (NOQ .EQ. 0.) GO TO 305
IF (TCON .LE. 0. .OR. NOO .GE. 2) GO TO 305
IF (RE .GT. 1185.) GO TO 310
C CALC LAMINAR H, THRCON = THERMAL CONDUCTIVITY OF AIR
HA = 2.182*THRCON/RAD(K)
GO TO 300
C CALC TURBULENT H
310 QAIR = HA*ALAT(K)*(TUBE(K) - TMEAN)
C CONVECTION HEAT TRANSFER FROM-TO AIR- TUBE
IF (P(K) .GT. PM) QTOT = QTOT + QAIR(K)
305 CONTINUE
500 CONTINUE
C INTEGRATE EQNS HERE, SEND VALUES FOR CALC DERIVATIVES
C INTO FUNCTION SUBROUTINE
C ROO = DMTOT/VR
C TMC = (SUM-CV*TMDMTOT+QTOT)/(CV*RO *VR)
C(1) = DMTOT
C(2) = VR
C(3) = SUM
C(4) = CV
C(5) = OW
C(5) = QTOT
CALL CONSTS(C)
20 CONTINUE

D-9
J=ONE
CALL INTEGT(J,RO,ROD,T,ISAVE,DELT)
J=ZERO
CALL INTEGT(J,TM,TMD,T,ISAVE,DELT)
IF(ISAVE .EQ. 1) GO TO 20

C COMPUTE TOTAL MASS CHANGE IN MANIFOLD UP TO THIS STEP
TOTMAS=TOTMAS+DMTOT*DELT
IF(TISO .LT. 0.) TM=ABS(TISO)
PM=RO*R*TM
IF ((T-TPRNT) .LT. PRNTFQ .AND. NCURVS .GT. 0 .AND. PLOTFQ .NE. I PRNTFQ .AND. T .LT. TEND) GO TO 10201
IF ((T-TPRNT) .LT. PRNTFQ .AND. T .LT. TEND) GO TO 21
I=I+6
IF( I .LT. 60) GO TO 505
WRITE(6,10100)
WRITE(6,10101)
WRITE(6,10106)
WRITE(6,10105)

10100 FORMAT('1 ALTITUDE',2X,'VELOCITY',7X,'TIME',3X,'MANIFOLD PRESS',
   14X,'PRESSURE (PSF) AT EACH PORT',24X,'ATMOS PRESS',4X,'QTOT')
10101 FORMAT('0',2X,'0O',8X,'MACH NO',8X,'ALPHA',2X,'MANIFOLD TEMP',5X,
   1 PRESSURE COEFFICIENT AT EACH PORT',18X,'ROLL RATE')
10106 FORMAT('0 MASS FLOW RATE IN EACH TUBE',23X,'REYNOLDS NO IN EACH T
   IUBE',27X,'INTERNAL SS',4X,'EXTERNAL SS')
10105 FORMAT('0 TEMPERATURE AT EACH PORT',26X,'PHASE ANGLE OF EACH PORT
   1',27X,'TOTAL MASS CHG IN MANIFOLD')
10103 FORMAT('0 VELOCITY IN EACH TUBE',29X,'DISCHARGE CCEFFICIENT AT EA
   ICH PORT')
10107 FORMAT('0 VELOCITY IN EACH TUBE',29X,'FRICTION FACTOR IN EACH TUB
   1E',23X,'OODD 1',8X,'QDOD 2')
10108 FORMAT('0 TCONV', 'MEAN GAS TEMPERATURE IN EACH TUBE',10X,'MEAN TUB
   1E TEMPERATURE OF EACH TUBE',17X,'CQCT 3',8X,'CDOD 4')
10109 FORMAT('0 AVOL', 'G13.6', 'ALAT', 'G13.6', 'TVCL', 'G13.6')
IF(NOQ .EQ. 0 .OR. NOQ .EQ. 1) WRITE(6,10107)
IF(NOQ .EQ. 2) WRITE(6,10103)
I=10
IF(TCON .NE. 0. .AND. NOQ .EQ. 1) WRITE(6,10108)
IF(TCON .NE. 0. .AND. NOQ .EQ. 1) I=I+1

505 CCNTINUE
WRITE(6,10110)(HO, VI, T, PM, PL1, P2, P3, P4, PO, QTG)
WRITE(6,10104)(GO, AA, ALPHAT, PM, (CPW(IZ),IZ=1,4 ),OQMEG
WRITE(6,10104)(DM(IZ),IZ=1,4 ),(REW(IZ),IZ=1,4 ),SS,AB
WRITE(6,10104)(TA(IZ),IZ=1,4 ),(PHII(IZ),IZ=1,4 ),TOTMAS
WRITE(6,10104)(VVOUT(IZ),IZ=1,4 ),(CCW(IZ),IZ=1,4)
IF(TCON .NE. 0. .AND. NOQ .EQ. 1) WRITE(6,10102)(CAIR(1),QAIR(2)
IF(TCON .NE. 0. .AND. NOQ .EQ. 1) WRITE(6,10104)(TCONV(IZ),IZ=1,4
1),(TUBE(IZ),IZ=1,4),(CAIR(3),QAIR(4)
IF(TCON .NE. 0. .AND. NOQ .EQ. 1) I=I+1
10102 FORMAT('1',104X,ZG13.6)
10104 FORMAT('1X,10G13.6')
10110 FORMAT('H0,10G13.6')
IF(NCURVS .EQ. 0) GO TO 90
C VARIABLE PLOT POINT CAPABILITY IS CHECKED HERE
10201 IF(PLOTFQ .EQ. PRNTFQ) GO TO 10202
PLOTF=PLOTF+1.
IF(PLOTF .LT. PLOTFQ) GO TO 10203
PLOTF=C.
10202 CCNTINUE
PLCT(1)=T.

• D-10
PLOT(2)=PM
PLCT(3)=P(1)
PLOT(4)=P(2)
PLOT(5)=P(3)
PLOT(6)=P(4)
PLOT(7)=PD
PLCT(8)=HQ
PLOT(9)=V1
PLOT(10)=QTOT
PLOT(11)=QO
PLOT(12)=AA
PLOT(13)=ALPHA
PLOT(14)=TM
PLCT(15)=REW(1)
PLOT(16)=REW(2)
PLOT(17)=REW(3)
PLOT(18)=REW(4)
PLOT(19)=CQW(1)
PLOT(20)=CQW(2)
PLOT(21)=CQW(3)
PLOT(22)=CQW(4)
DO 10200 IPL= 1,NCURVS
10200 APLOT(IPL)=PLOT(IPL(IPL(IPL))
   CALL RJPLOT(APLOT,0)
10203 IF((T-TPRNT) .LT. PRNTFQ) GO TC 21
90    CONTINUE
   IF(T .LT. TEND) GO TO 35
   IF(PLTCON .EQ. 1.) GO TO 10209
   CALL RJPLOT(APLOT,-1)
10209 RETURN
ENC
/*
// EXFC LINK
//SYSLMOD DD UNIT=DISK, VOL=SER=M2SCR6, DSN=NQJFLMFD, DISP=(NEW,KEEP)
//LINK.OBJECT DD *
** DATA DMS,N,JLMFD
** DATA CMS,N,RJPLT
** DATA DMS,N,USUBS
ENTRY MAIN
NAME NQJFLMFDIR)
*/
** DATA DMS,N,GCA19

D-11
SUBROUTINE CSUBP

COMMON/ZILCH,YX15,YPRIME15,ERCRT(2),P(4),RAD(4),AL(4),
1ALT(50),VEL(50),TF(50),CQ(10),FREN(10),CMACH(10),ALFA(10,15),
2ALPHAA(10),TALPH(10),PRESSF(6),TCCN,TTUBE,THICK,CPTUBE,ROTUBE,
3FI(10,15,20),CPP(10,15,20),T,DELT,DELMAX,P,VR,THETA,OMEGA,
4THRCN,PLTCN,FAZANG,ALPHA,CP,AA,TH,PRNTFQ,TEND,PHI,RT,INDPVR,
5JMAX,NLIN,ERCRT,IXMAX,IZMAX,KT,NOQ,NCPTS,NCQ,NALPHA,
6NPPTS,NCURVS,MODE,IPSKIP,IPLOT(7)

C PRESSURE COEFFICIENT SUBROUTINE
C CP IS DETERMINED FROM M NO,ALPHA,AND PHI FROM EXPERIMENTAL CURVES
C THREE WAY LINEAR INTERPOLATION USED TO FIND CP
C PHI IS THE ROTATION PLUS PORT POSITION AND IS SET .LE. 180 DEG
C SINCE CP CURVES ARE CONSIDERED SYMMETRIC
C AT IS EITHER MACH NO OR TIME DEPENDING ON THE VALUE OF
C PRESSF(6), IE, PRESSF(6)=0,1 FOR MACH NO, =-1,2 FOR TIME
AT=AA
IF(PRESSF(6) .EQ. -1. .OR. PRESSF(6) .EQ. 2.) AT=T
PI=3.14159
ALPHA1=ALPHA
IXMIN=1
IYMIN=1
IZMIN=1
IX=1
IY=1
IZ=1
MINTRP=0
C IF TABLE END POINTS ARE HIT MINTRP =2 STOPS DOUBLE INTERPOLATION
IF(AT .LT. CMACH(IXMAX) .AND. AT .GT. CMACH(IXMIN)) GO TO 501
IF(AT .LE. CMACH(IXMIN)) GO TO 504
IX=IXMAX
CMACH(IX)=CMACH(IXMAX)
MINTRP=2
GO TO 503
504 IX= IXMIN
CMACH(IX)=CMACH(IXMIN)
MINTRP=2
GO TO 503
501 CCNTINUE
IF(AT .GE. CMACH(IX) .AND. AT .LT. CMACH(IX+1)) GO TO 502
IX=IX+1
GO TO 501
502 CCNTINUE
C INTERPOLATE ON MACH NO, MINTRP=1 TRIGGERS SECOND INTERPOLATION
405 IF(MINTRP .EQ. 1) IX=IX+1
503 CCNTINUE
IF(ALPHA1 .LT. ALFA(IX,IYMAX) .AND. ALPHA1 .GT. ALFA(IX,IYMIN))
1 GC TO 511
IF(ALPHA1 .LE. ALFA(IX,IYMIN)) GO TO 514
IY=IYMAX
ALFA(IX,IY)= ALFA(IX,IYMAX)
GO TO 512
514 IY=IYMIN
ALFA(IX,IY)= ALFA(IX,IYMIN)
GO TO 512
511 IF(ALPHA1 .GE. ALFA(IX,IY) .AND. ALPHA1 .LT. ALFA(IX,IY+1))GO T0512
IY=IY+1
GO TO 511
512 CCNTINUE
C PHI= VEHICLE ROLL ANGLE
PHI=OMEGA*T+THETA

D-12
RESET=PHI/(2.*PI)
IRESET=RESET
PHI=(RESET-IRESET)*2.*PI

C FOR NONSYMMETRIC CP DATA SKIP ASSUMPTION OF SYMMETRY
IF( PHI .GT. PI AND PRESSF61 .LT. 1.) PHI=2.*PI-PHI

C PHI IN DEG = PHI(RAD)*57.296(DEG/RAD)

540 IF(PHI .GE. FI(IX, IY, IZ) .AND. PHI .LE. FI(IX, IY, IZ+1)) GO TO 550
IZ=IZ+1
GO TO 540

550 CONTINUE
C TEMPORARY PROCEDURE FOR SOLVING END POINTS
IF(IY .LT. IYMAX) GO TO 555
IY=IYMAX-1
555 IF(IZ .LT. IZMAX) GO TO 560
IZ=IZMAX-1

560 CONTINUE
DIV1=FI(IX, IY+1, IZ+1)-FI(IX, IY+1, IZ)
DIV2=FI(IX, IY, IZ+1)-FI(IX, IY, IZ)
SLOPE1=(CPP(IX, IY+1, IZ+1)-CPP(IX, IY+1, IZ))/DIV1
SLOPE2=(CPP(IX, IY, IZ+1)-CPP(IX, IY, IZ))/DIV2
CP1=SLOPE1*(PHI-FI(IX, IY+1, IZ))+CPP(IX, IY+1, IZ)
CP2=SLOPE2*(PHI-FI(IX, IY+1, IZ))+CPP(IX, IY, IZ)
DIFF=(CP1-CP2)/(ALFA(IX, IY+1)-ALFA(IX, IY))

563 ALPH=ALPHA1
IF(ALPH .LE. ALFA(IX, IY+1)) GO TO 570
ALPH=ALFA(IX, IY+1)
570 CP=DIFF*(ALPH-ALFA(IX, IY))+CP2
IF(MINTRP .GT. 0) GO TO 400
MINTRP=1
CP=CP
GO TO 405

400 IF(MINTRP .EQ. 2) GO TO 410
C CP VALUE BETWEEN MACH NUMBERS
CP=((CP-CP1)*(AT-CMACH(IX-1)))/(CMACH(IX)-CMACH(IX-1))+CP1

410 CONTINUE
C FAZANG IN RAD =2.*PI/KT) WHERE KT = NO CF PORTS
THETA=THETA+FAZANG
RETURN
END

D-13
SUBROUTINE READIN(DICT,B,NDICT)
REAL*8 DICT(NDICT),A(100),ARRAY
INTEGER OPTION
DIMENSION B(NDICT), INPUT(100),FMT(20),FMTA(20),
IFMTW(20),FMTHOL(20)
C INITIALIZE READ AND WRITE FORMATS
DATA FMT/'(10G',',8.5'),18*' '/
DATA FMTA/'(lOG',',8.5'),18*' '/
DATA FMTW/(10G',',13.6'),') ',17*' '/
C NDICT IS THE NUMBER OF DICTIONARY ENTRIES, IE HIGHEST INDEX USED
DO 1000 I=1,100
A(I)=0.
INPUT(I)=0
1000 CONTINUE
1 FORMAT(1615)
2 FORMAT(10A8)
3 FORMAT(20A4)
7 FORMAT(1HO,1OA8)
9 FCRMATIA8,I7,13I5,(/,16I5))
10 FORMAT('O THE NAME ',A8,'IS NOT IN THE DICTIONARY, IT IS MISSPELLED OR MISPLACED IN THE FIELD. PLEASE CORRECT AND RESUBMIT PROGRAM')
C ISTOP IS USED TO STOP THIS PROGRAM WHEN INPUT NAMES ARE UNRECOGNIZABLE, ALL NAMES ARE PROCESSED BEFORE STOPPING
ISTOP=0
C READ IN THE NUMBER OF VARIABLE NAMES OR NO OF VARIABLES IN AN ARRAY
101 CONTINUE
NEWFMT=0
READ(5,1)OPTION,NOVAR,NEWFMT
C NEWFMT SETS UP OUTPUT FORMATS
C 0 - NO FMT, 1- WRITE FMT, 2- HOLLERITH FMT, 3- BOTH 1&2
IF(NEWFMT .EQ. 2 .OR. NEWFMT .EQ. 0.53) READ(5,3) FMTHOL
IF(NEWFMT .EQ. 2 .OR. NEWFMT .EQ. 0.53) WRITE(6,FMTHOL)
IF(NEWFMT .EQ. 51 .OR. NEWFMT .EQ. 0.53) READ(5,3) FMTW
C OPTIONS- 0= READ VAR NAME(S) -RETURN, 1=0+READ ANOTHER OPTION
C AFTER THE DATA, 2=READ IN AN ARRAY NAME-RETURN, 3=2+READ ANOTHER OPTION, 6=READ IN AN ARRAY NAME + SPECIFIC ARRAY LOCATIONS, 7=
6 READ IN ANOTHER OPTION, 4 PREFIXES ANY PREVIOUS OPTION NO
C WHEN IT IS DESIRED TO WRITE THAT INPUT DATA, IE 40,46, ETC
C 5 PREFIXES ANY PREVIOUS OPTION TO INPUT A NEW DATA FORMAT
C IE, RE-DEFINE A PREVIOUSLY DEFINED DATA FORMAT, 50,56, 540, 543
C NOTE OPTION 4 MAY NOT PRECEDE OPTION 5 IF 450 IS INVALID
ISKIP=5
MINUS=50
IF(OPTION .LT. 50) GO TO 77
IF(OPTION .GE. 500) MINUS=500
C SET OPTION
OPTION=OPTION-MINUS
ISKIP=0
77 CONTINUE
IF(OPTION .EQ. 2 .OR. OPTION .EQ. 3) GO TO 102
IF(OPTION .EQ. 42 .OR. OPTION .EQ. 43) GO TO 102
IF(OPTION .EQ. 6 .OR. OPTION .EQ. 7 .OR. OPTION .EQ. 46 .OR.
1 OPTION .EQ. 47) GO TO 103
C READ VARIABLE NAMES INTO THE A ARRAY BY A FORMAT
READ(5,2)(A(I),I=1,NOVAR)
DO 5 I=1,NOVAR
DO 6 K=1,NDICT
IF(DICT(K) .EQ. A(I)) INPUT(I)=K
IF(DICT(K) .EQ. A(I)) GO TO 5
5 CONTINUE
D-14
C IF THIS WRITE IS IMPLEMENTED AN UNRECOGNIZABLE NAME HAS BEEN FOUND, NAME PROCESSING WILL CONTINUE UNTIL END OF NAMES
WRITE(6,10) Ai(I)
ISTOP=1
INPUT(I)=1
CONTINUE
C READ THE FORMAT FOR THE VARIABLES TO BE READIN
IF(ISKIP .NE. 5) READ(5,3) FMT
READ(5,FMT)( B(INPUT(I)),I=1,NOVAR)
IF(OPTION .EQ. 1) GO TO 101
IF(OPTION .EQ. 40) WRITE(6,FMTW)( B(INPUT(I)),I=1,NOVAR)
IF(OPTION .EQ. 41) WRITE(6,FMTW)( B(INPUT(I)),I=1,NOVAR)
IF(OPTION .EQ. 41) GO TO 101
IF(ISTOP .EQ. 1) GO TO 75
RETURN
C NOARRY IS THE NO OF ARRAY VALUES TO BE READ IN STARTING AT ARRAY LOCATION ISTART
102 NOARRY=NOVAR
READ(5,9)ARRAY, ISTART
IF(ISTART .LT. 1) ISTART=1
DO 50 K=1,NDICT
IF(ARRAY .EQ. DICT(K)) GO TO 55
CONTINUE
C IF THIS WRITE IS IMPLEMENTED AN UNRECOGNIZABLE NAME HAS BEEN FOUND, NAME PROCESSING WILL CONTINUE UNTIL END OF NAMES
WRITE(6,10) ARRAY
ISTOP=1
K=1
55 KA=K+ISTART-1
NPLACE=KA+NOARRY-1
IF(ISKIP .NE. 5) READ(5,3) FMTA
READ(5,FMTA)( B(I),I=KA,NPLACE)
C LOAD B ARRAY WITH ARRAY VALUES FOR TRANSMISSION TO USERS PROG
IF(OPTION .EQ. 3) GO TO 101
IF(OPTION .EQ. 42) WRITE(6,FMTW)( B(I),I=KA,NPLACE)
IF(OPTION .EQ. 43) WRITE(6,FMTW)( B(I),I=KA,NPLACE)
IF(OPTION .EQ. 43) GO TO 101
IF(ISTOP .EQ. 1) GO TO 75
RETURN
103 CONTINUE
READ(5,9) ARRAY,(INPUT(I),I=1,NOVAR)
DO 60 I=1,NDICT
IF(ARRAY .EQ. DICT(I)) GO TO 61
CONTINUE
C IF THIS WRITE IS IMPLEMENTED AN UNRECOGNIZABLE NAME HAS BEEN FOUND, NAME PROCESSING WILL CONTINUE UNTIL END OF NAMES
WRITE(6,10) ARRAY
ISTOP=1
I=1
61 CONTINUE
ISAVE=I-1
DO 62 I=1,NOVAR
INPUT(I)=INPUT(I)+ISAVE
CONTINUE
IF(ISKIP .NE. 5) READ(5,3) FMTA
READ(5,FMTA)( B(INPUT(I)),I=1,NOVAR)
C LOAD B ARRAY
IF(OPTION .EQ. 46 .OR. OPTION .EQ. 47) WRITE(6,FMTW)(B(INPUT(I)),1 I=1,NOVAR)
IF(OPTION .EQ. 7 .OR. OPTION .EQ. 47) GO TO 101
75 IF(ISTOP .EQ. 1) STOP
RETURN
END
SUBROUTINE SINTRP (A,B,C ,IMAX,IS, I, IORDER, D, NOPRNT)
IMPLICIT REAL*8(A-H,O-Z)
REAL*4 A,B,C,D
DIMENSION B(100),C(100)
DIMENSION X(100),Y(100),NUM( 20),CENDM( 20, 20),U(20)
REAL NUM,NUMT
XI=A
22223  CONTINUE
   DO 60 K=1,IMAX
   Y(K)=B(K)
   X(K)=C(K)
60   CONTINUE
   IF(IORDER .LE. 1) IORDER=2
C SEARCH AND INTERPOLATE ROUTINE
C IS=0 FOR BOTH SEARCH AND INTERP
C IS=1 FOR SEARCH ONLY
C IS=2 FOR INTERP ONLY
C NOPRNT=3 SUPRESSES THE WRITTING OF THE
C DIAGONISTIC MESSAGE ,NOPRNT RETURNS A VALUE OF -1,0,OR 1 DEPENDING
C ON WHETHER THE INTERPCLATION TCCK PLACE OUTSIDE THE TABLE ,LOWER
C (-1),UPPERFND(1),OR INSIDE THE TABLE(0)
   IF(IS .EQ. 2) GO TO 20
      I=1
   IF(XI .LT. X(IMAX) .AND. XI .GT. X(1)) GO TO 10
   IF(XI .GT. X(1)) GO TO 15
      I=IMAX
      GO TO 20
15   I=IMAX
      GO TO 20
10   IF(XI .LE. X(I+1) .AND. XI .GT.X(1)) GO TO 20
      I=I+1
      GO TO 10
20   CONTINUE
   IF(IS .EQ. 1) RETURN
C TABLE WITHIN A TABLE INDEX LOCATION
   IT=1
   IF(IS .EQ. 2 .AND. I .GT. 1) IT=I
   IF(IORDER .GT. IMAX) IORDER=IMAX
   IORDR=IORDER
22   IF((I+IORDER-1) .LE. IMAX) GO TO 21
      IORDER=IORDER-1
   IF(IORDER .NE. 1) GO TO 22
46   IF(NOPRNT .EQ. 3) GO TO 45
      WRITE(6,101) Y(IT),X(IT),Y(IMAX),X(IMAX),XI,IMAX
101  FORMAT(1HO,'AN ERRONEOUS VALUE HAS BEEN SUPPLIED TO SINTRP, IE XI
1 IS OUTSIDE THE TABLE BEING USED,Y1=' ,D15.8,'X1=',D15.8,/,'
2 ' ,D15.8,'XMAX=',D15.8,'XI=',D15.8,'IMAX=',I5, ' THE APPROPRIATE
3 ENC POINT IS USED FOR THE INTERPOLATED VALUE')
45   CONTINUE
   IT=IMAX
   IF(XI .LT. X(IMAX)) II=1
   UX=Y(II)
   IORDER=IORDR
   NOPRNT=-1
   IF( II .EQ. IMAX) NOPRNT=1
   D=UX
   RETURN
21   IF(XI .LT. X(II)) GO TO 46
C SAVE THE ORIGINAL INDEX I TO RETURN TO CALLING PROGRAM
   IRETRN=I
THE INDEX I REPRESENTS THAT POINT IN THE TABLE WHERE THE POLYNOMIAL WAS STARTED.

IF(IORDER .NE. IORDRS) I=I-(IORDRS-IORDER)

RETURN IORDER TO ORIGINAL VALUE
IORDER=IORDRS
NOPRNT=0
ISAVE =I
DC 25 II=1,IORDER
I=ISAVE+II-1
NUM(II)=XI-X(I)
DC 25 LL=1,IORDER
L=LL+ISAVE-1
DENUM(II,LL)=X(II)-X(L)

SET DIAGONAL =1 SO THAT DIVISICA BY 0 DOES NOT OCCUR
IF(I .EQ. L) DENOM(II,LL)=1.

CONTINUE
DO 26 I=1,IORDER
TERM =1.
DO 27 L=1,IORDER
II=L
IF( I .NE. L) GO TO 30
NUMT=NUM(II)
NUM(II)=1.
30 TERM=TERM*(NUM(II)/DENUM(II,L))
IF(I .EQ. L) NUM(II)=NUMT
CONTINUE
U(I)=TERM
CONTINUE
I=ISAVE

VALUE OF INDEP VAR AT XI
UX=0.
DO 40 K=1,IORDER
UX=UX+U(K)*Y(I)
I=I+1.
CONTINUE
I=IRETRN
D=LX
CONTINUE
RETURN
END
SUBROUTINE AT62 (ZFT, ANS)

REAL MOLWT, LOGPR

DIMENSION H(23), TBASE(22), TGRAD(22), PBASFI22), MCLWT(23), ANS(4)

DATA H/0., 11., 20., 32., 47., 52., 61., 79., 88., 743., 90., 100., 110., 120., 1
150., 160., 170., 190., 230., 300., 400., 500., 600., 700., TBASE/288.15, 216
2.65, 216.65, 228.65, 270.65, 270.65, 252.65, 180.65, 180.65, 180.65, 210.65
3., 2EC.65, 360.65, 960.65, 1110.65, 1210.65, 1350.65, 1550.65, 1830.65, 2160
4., 65, 2420.65, 2590.65, TGRAD/6.5, 0., 1., 2.8, 0., -2., -4., 0., 0., 3., 5., 1
50., 20., 15., 10., 6., 4., 3., 2., 1., 1./PBASFI1., 2.32361E-01, 5.40
6326E-02, 8.56663E-03, 1.09455E-03, 5.82289E-04, 1.79718E-04, 1.0241E-05
7, 1.6223E-06, 1.6223E-06, 2.9681E-07, 7.2582E-08, 2.4887E-08, 4.9955E-09
8, 3.6460E-09, 7.7561E-09, 1.6632E-09, 6.8694E-10, 1.8592E-10, 3.9777E-11
9, 1.814E-11, 3.405E-12, MOLWT/10.289.96, 28.89, 28.56, 28.07, 26.92, 26
B/34.1628/, CRHO/.3236458E-03/, PSFA/2116.22/

IF (ZFT < 0.) ZFT = 0.

IF (ZFT .GT. 2.E06) ZFT = 2.E06

Z = 3048E-03 * ZFT

IF (Z .GE. 90.) GO TO 5

Z = Z / (1. + (Z / RE))

DO 1 J = 1, 9

IF (Z .GE. H(J) .AND. Z .LE. H(J + 1)) GO TO 2

2 CONTINUE

TKELV = TBASE(J) + TGRAD(J) * (Z - H(J))

IF (ABS(TGRAD(J)) .LT. 5) GO TO 3

PSF = PBASFI(J) * PSEA * (((TBASE(J) / TKELV)**(C/TGRAD(J)))) ** GO TO 4

3 CONTINUE

PSF = PBASFI(J) * PSEA * EXP(-C * (Z - H(J)) / TRASE(J))

4 CONTINUE

SLGFT3 = CRHU * PSF / TKELV

VSOUND = SQRT(4325.746 * TKELV)

GO TO 8

5 CONTINUE

DO 6 K = 10, 22

IF (Z .GE. H(K) .AND. Z .LE. H(K + 1)) GO TO 7

6 CONTINUE

7 CONTINUE

TP = TBASE(K) + TGRAD(K) * (Z - H(K))

A = 1. + (H(K) / RE)

B = TBASE(K) / (TGRAD(K) * RE)

X = (Z - H(K)) / RE

LOGPR = (-C / TGRAD(K)) * (1. / (A - B)) * (1. / (1. / (A - 1.)) + (1. / (A - 1.))) + ALOG

1/(A + B + X)) / (B * (A + X)))

PSF = PBASFI(K) * PSEA * EXP(LOGPR)

SLGFT3 = CRHU * PSF / TP

DMOLWT = ((MOLWT(K + 1) - MOLWT(K)) / (H(K + 1) - H(K))) * (Z - H(K))

TKELV = (MOLWT(K) + DMCLWT) / 28.9644 * TM

VSOUND = 894.50

8 CONTINUE

ANS(1) = SLGFT3

ANS(2) = PSF

ANS(3) = TKELV

ANS(4) = VSOUND

RETURN

END
SUBROUTINE START

IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 WW,FDT,TT,DELT,YX(15),XX,DELMX,ERCRT(2),YDOT(15),C(15)
COMMON/ZILCH/ FDOT(10,15),W(10,15),CST(15),YY(10,15),Z(15),T(10),
1 ZIP(15),FDOT(10,15),PCER,DELT,DELMX,SAVE,SI,SAVE1, SII,
2 ISKIP(15),K,N,J,MAXMT,IDUBLE,ICHECK,NODBLE,NONLIN,ITERAT
C W=DEP VAR,T= INDEP VAR, DELT=CHANGE IN INDEP VAR.
C JMAX= NO OF EONS ,K,N,1, ARE FREE INDICES
C DELMAX = MAXIMUM ALLOWABLE TIME STEP-USER INPUT
C YX= INITIAL VALUES OF THE INDEPENDENT VARIABLE
C YPRIM= INITIAL VALUES OF YDOT FOR THE DEPENDENT VARIABLE
C DELT DELTA T ,STEP SIZE INITIAL VALUE
C ECRIT(I)= ERROR CRITERIA FOR STABILITY, ECRIT(2)=PRED-CORR CRITERIA
C NONLIN= INDICATES WHETHER OR NOT ITERATION TECHNIQUE IS TO BE USED
C IERCRT== INDICATES WHETHER OR NOT THE DEFAULT OPTION FOR THE ERROR
C CRITERIA IS TO BE USED, IERCRT=0,1,2,3 - DEFAULT OPTION
C SII CHANGED, PCER CHANGED, BOTH SII AND PCER CHANGED
C DEFAULT OPTION FOR ERROR CRITERIA
SII=.01
PCER=.01
IF(IERCRT .EQ. 0) GO TO 5
IF(IERCRT .EQ. 1) SII=ERCRT(1)
IF(IERCRT .EQ. 2) PCER=ERCRT(2)
IF(IERCRT .NE. 3) GO TO 5
SII=ERCRT(1)
PCER=ERCRT(2)
5 CONTINUE
DO 20 I=1,JMAX
W(I,1)=YX(I)
IF(NONLIN .GT. 0) FDOT(I,1)=YDOT(I)
ISKIP(I)=1
ZIP(I)=0.
20 CONTINUE
DEL=DELT
SAVE1=0.
K=1
IDUBLE=1
NODBLE=0
ISTEP=1
ISAVE=SAVE1
ICHECK=0
RELTIM=T(1)
RETURN
ENTRY CONSTS(C)
C CST TRANSFERS CONSTS UR VAR TO THE DIFF EQN WHILE INTEG IS IN PROCESS
DO 30 IC=1,15
CST(IC)=C(IC)
30 RETURN
ENTRY INTEGR(JJ,WW,FDT,TT,ISAVE, DELT)
J=JJ
10 CALL INTGRT
ISAVE=SAVE1
NN=N
IF( J .LT. JMAX AND. ISKIP(JMAX) .GT. 4) NN=N+1
WW=W(NN,J)
FDT=FDOT(NN,J)
TT=T(NN)
DELT=DELT
22221 CONTINUE
RETURN
END
SUBROUTINE FUNCT
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/ZILCH/,FDOT(10,15),W(10,15),CST(15),YY(10,15),Z(15),T(10),
1 ZIP(15),FDOTS(10,15),PCER ,DELT,DELMAX,DEL,SAVE,S1,SAVE1, SII,
2 ISKIP(15),K,N,J,JMAX,ISTEP,IDUBLE,ICHECK,NODBLE,NONLIN,ITERAT
DIMENSION Y(20),YDOT(20)
C THE EQNS ARE OF THE FORM YDOT=F(X,Y), WHERE X IS THE INDEP VAR AND
C Y IS THE DEP VAR
X=T(K)
DO 50 I=1,JMAX
  Y(I)=W(K,I)
  IF(K.LT.8) Y(I)=W(I,I)
  YDOT(I)=FDOT(K,I)
  IF(K.LT.8) YDOT(I)=FDOT(I)
50 CONTINUE
100 GO TO (1,2,3,4,5,6,7,8,9,10),J
1 CONTINUE
C PLACE THE 1ST EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  YDOT(1)=CST(1)/CST(2)
  FDOT(K,J)=YDOT(J)
  RETURN
2 CONTINUE
C PLACE THE 2ND EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  YDOT(2)=(CST(3)-CST(4)*Y(2)+CST(5)/CST(4)*Y(1)+CST(2))
  FDOT(K,J)=YDOT(J)
  RETURN
3 CONTINUE
C PLACE THE 3RD EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
4 CONTINUE
C PLACE THE 4TH EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
5 CONTINUE
C PLACE THE 5TH EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
6 CONTINUE
C PLACE THE 6TH EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
7 CONTINUE
C PLACE THE 7TH EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
8 CONTINUE
C PLACE THE 8TH EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
9 CONTINUE
C PLACE THE 9TH EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
10 CONTINUE
C PLACE THE 10TH EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
  FDOT(K,J)=YDOT(J)
  RETURN
END
SUBROUTINE ERROR  
IMPLICIT REAL*8 (A-H,O-Z)  
COMMON/ZILCH/ FOOT(10,15),W(10,15),CST(15),YY(10,15),Z(15),T(10),  
1 ZIP(15),FOOTS(10,15),PCER ,DELT,DELMAX,DEL,SAVE,S1,SAVE1, SII,  
2 ISKIP(15),K*N,J,JMAX,ISTEP,DOUBLE,ICHECK,NODBLE,NONLIN,ITERAT  
DIMENSION B(10),AK(10)  
DIMENSION Y(20),YDOT(20)  
C  DFDY REPRESENTS THE PARTIAL OF YDOT WITH RESPECT TO Y  
C  THE EQNS ARE OF THE FORM DFDY=F(X,Y), WHERE X IS THE INDEP VAR AND  
C  Y IS THE DEP VAR  
C  YDOT REPRESENTS THE DERIVATIVE OF THE FUNCTION BEING WORKED ON  
C  X=T(N+1)  
DO 50 I=1,JMAX  
30  YDOT(I)=WIN+1, I)  
C  AK IS A CHECK FOR STABILITY,B CHECKS FOR TRUNCATION ERROR  
C  AK=ABS(D(F)/DY)*DELT  
120  GO TO (1,2,3,4,5,6,7,8,9,10, J,  
1  CONTINUE  
C  PLACE THE 1ST EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
DFDY=0.  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
2 CONTINUE  
C  PLACE THE 2ND EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
DFDY=0.  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
3 CONTINUE  
C  PLACE THE 3RD EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
DFDY=(CST(1)+CST(2)*DCOS(4*Y(4))e**2*Y(3)-(CSTI3)+CSTI4)*DSIN(4.*  
1 Y(4)))**2*Y(3)-(CST(1)+CST(2)*DCOS(4.*  
1 Y(4)))**2*Y(3)-(CST(3)+CST(4)*DSIN(4.*Y(4))  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
4 CONTINUE  
C  PLACE THE 4TH EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
DFDY=0.  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
5 CONTINUE  
C  PLACE THE 5TH EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
6 CONTINUE  
C  PLACE THE 6TH EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
7 CONTINUE  
C  PLACE THE 7TH EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
8 CONTINUE  
C  PLACE THE 8TH EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
9 CONTINUE  
C  PLACE THE 9TH EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD  
AK(J)=DABS(DFDY*DELT)  
GO TO 100  
10 CONTINUE  

D-21
C PLACE THE 10TH EON, DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
AK(J)=DABS(DFDY*DELT)
100 CCNTINUE
C Z(J)= PRED(J)/CORR(J)
B(J)=DABS(Z(J))
56 IF(J LT. JMAX) RETURN
SAVE=B(1)
SI=AK(1)
C DETERMINATION OF LARGEST ERROR TERM DETERMINES STEP SIZE
JAX=JMAX-1
IF(JAX EQ. 0) GO TO 205
DO 35 L=1,JAX
IF(SAVE-B(L+1)) 40,36,36
35 CCNTINUE
205 CCNTINUE
SAVE=(DABS(SAVE)-1.)
RETURN
END
SUBROUTINE DERIV5(X,ZPC,W,FDOT,T,J,N)
IMPLICIT REAL*8 (A-H,O-Z)
EXP(IQQQ)=DEXP(IQQQ)
ALOG(IQQQ)=DLOG(IQQQ)
ABS(IQQQ)=DABS(IQQQ)
DERIV (A,B,C,D,E,F,G,H,O,P)=A/F+B/G+C/H+D/O+E/P
DIMENSION W(10,15),FDOT(10,15),Y(15),T(10)
DIMENSION DIV(20)
DO 25 I=1,10
25 DIV(I)=1.
C IF THE DIVIDED DIFFERENCE EQNS. ARE NOT USED FOR THE 5TH DERIV
C OF THE JTH EQN SET DIV(J)=0.
IF(DIV(J) .EQ. 0.) GO TO 30
FN=FDOT(N,J)
FN1=FDOT(N-1,J)
FN2=FDOT(N-2,J)
FN3=FDOT(N-3,J)
FN4=FDOT(N-4,J)
DENOM1=(T(N)-T(N-1))*(T(N)-T(N-2))*(T(N)-T(N-3))*(T(N)-T(N-4))
DENOM2=(T(N-2)-T(N-1))*(T(N-2)-T(N-3))*(T(N-2)-T(N-3))*(T(N-2)-
1 T(N-4))
DENOM3=(T(N-3)-T(N-1))*(T(N-3)-T(N-2))*(T(N-3)-T(N-3))*(T(N-3)-
1 T(N-4))
DENOM4=(T(N-4)-T(N-1))*(T(N-4)-T(N-2))*(T(N-4)-T(N-3))*(T(N-4)-
1 T(N-4))
IF(DENOM1 .NE. 0. .AND. DENOM2 .NE. 0. .AND. DENOM3 .NE. 0. .AND. 
DENC4 .NE. 0. ) GO TO 30
1 +W(N-3,J))/12. )
RETURN
30 CCNITUE
DO 20 1=1,15
20 Y(I)=W(I,J)
C ZPC REPRESENTS THE 5TH DERIVATIVE OF THE FUNCTION
C THE USER MAY DEFINE ZPC ANALYTICALLY OR USE THE STATEMENT
C FUNCTION DERIV WHICH FINDS ZPC VIA THE 4TH DIVIDED DIFFERENCE
C OF THE FUNCTION'S DERIVATIVE,FDOT
C CONTINUE STATEMENT REPRESENTING THE EQN NUMBER
GO TO (1,2,3,4,5,6,7,8,9,10),J
1 CONTINUE
ZPC=DERIV (FN,FN1,FN2,FN3,FN4,DENOM,DENOM1,DENOM2,DENOM3,DENOM4)
RETURN
2 CONTINUE
ZPC=DERIV (FN,FN1,FN2,FN3,FN4,DENOM,DENOM1,DENOM2,DENOM3,DENOM4)
RETURN
3 CONTINUE
ZPC=DERIV (FN,FN1,FN2,FN3,FN4,DFNOM,DENOM1,DENOM2,DENOM3,DENOM4)
RETURN
4 CONTINUE
ZPC=DERIV (FN,FN1,FN2,FN3,FN4,DFNOM,DENOM1,DENOM2,DENOM3,DENOM4)
RETURN
5 CONTINUE
RETURN
6 CONTINUE
RETURN
7 CONTINUE
RETURN
8 CONTINUE
RETURN
9 CONTINUE
RETURN
10 CONTINUE
RETURN
END
SUBROUTINE RKUTTA
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/ZILCH/ FDOT(10,15),W(10,15),CYT(15),YY(10,15),Z(15),T(10),
1 ZIP(15),FDOTS(10,15),PCEP,DEL,DELMAX,DEL,SAVE,SAVEL,STI,
2 ISP(15),K,N,J,JMAX,ISTEP,DOUBLE,ICHECK,NODEL,NONLIN,ITERAT
DIMENSION A(4)
DO 10 I=1,4
CALL FUNCT
A(I) = DELT * FDOT(K,J)
GO TO (11,13,10),I
1 IF(J.EQ.1) T(K+1) = T(1)+DELT*.5
2 W(K+1,J) = W(1,J) + A(I)*.5
K=K+1
GO TO 10
3 IF(J.EQ.1) T(K+1) = T(1)+DELT
W(K+1,J) = W(1,J) + A(I)
K=K+1
10 CONTINUE
N=ISKIP(J)+4
W(N,J)=W(1,J)+(A(1)+2.*(A(2)+A(3))+A(4))/6.
IF(J.EQ.1) T(N) = T(1)+DELT
W(K,J)=W(N,J)
T(K)=T(N)
CALL FUNCT
FDOT(N,J)=FDOT(K,J)
K=1
IF(ISKIP(1).GE.2) GO TO 50
IF(ICHECK.EQ.1) GO TO 15
CALL STPCHK(W,FDOT,T,DEL,DEL,J,JMAX,ICHECK,N)
SAVE1=1.
RETURN
15 IF(ICHECK.EQ.2) GO TO 20
IF(ICHECK.EQ.1) CALL STPCK1(W,FDOT,T,N,J,JMAX,ICHECK)
SAVE1=1.
RETURN
20 IF(J.LT.JMAX) RETURN
CALL STPCK2(W,FDOT,T,SAVE1,N,J,JMAX,ICHECK)
C IF ICKECK=3 DELT WAS TOO LARGE ,CALC WILL BE REDONE WITH DELT/2
IF (ICHECK.NE.3) GO TO 50
ICHECK=0
RETURN
50 IF(J.LT.JMAX) RETURN
30 DO 25 J=1,JMAX
W(1,J)=W(N,J)
FDOT(1,J)=FDOT(N,J)
ISKIP(J) = ISKIP(J)+1
C THE FOLLOWING STATMENT LOADS THE 9TH ARRAY LOCATION OF FDOT(9,J)
C AND W(9,J) SO THAT WHEN THE PRED-CORR TAKES OVER THE 9TH
C LOCATION WILL CONTAIN MEANINGFUL VALUES
IF(N.EQ.8) W(N+1,J)=W(N,J)
IF(N.EQ.8) FDOT(N+1,J)=FDOT(N,J)
100 FORMAT(1H0,3D15.8,2I5)
25 CONTINUE
K=1
T(1)=T(N)
J=JMAX
RETURN
END

D-24
SUBROUTINE STPCHK(W,FDOT,T,DELT,DEL,J,JMAX,ICHECK,N)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(15),B(15),W(10,15),FDOT(10,15),T(10),D(15),E(15)
A(J)=W(N,J)
B(J)=FDOT(N,J)
D(J)=W(I,J)
E(J)=FDOT(I,J)
IF(J.LT.JMAX)RETURN
C=T(N)
ICHECK=1
F=T(1)
DELT=DELT*.5
RETURN
ENTRY STPCK1(W,FDOT,T,N,J,JMAX,ICHECK)
W(I,J)=W(N,J)
FDOT(I,J)=FDOT(N,J)
IF(J.LT.JMAX)RETURN
ICHECK=2
T(I)=T(N)
RETURN
ENTRY STPCK2(W,FDOT,T,SAVE1,N,J,JMAX,ICHECK)
DO 1 I=1,JMAX
CHECK=DABS(A(I)-W(N,I))
IF(CHECK.GT.1.D-05*DABS(W(N,I))) GO TO 5
1 CCNTINUE
DELT=DEL
ICHECK=0
SAVE1=0.
RETURN
5 CCNTINUE
DO 21=1,JMAX
W(1,1)=D(I)
FDOT(1,1)=E(I)
2 CCNTINUE
T(I)=F
DELT=DELT
ICHECK=3
SAVE1=1.
RETURN
END
SLBRoutines INTGRT
IMPLICIT REAL*8 (A-H,O-Z)
EXP(QQQ)=DFXP(QOOQ)
ALOG(QQQ)=DLOG(QQQ)
ABS(QQQ)=DABS(QQQ)
COMMON/ZILCH/ FDOT(10,15),W(10,15),CST(15),YY(10,15),Z(15),T(10),
IZIP(15),FDOTS(10,IS5),PCER ,DELT,DELMAX,DEL,SAVE,SI,SAVEL, SII,
2 ISKIP(15),K,N,J,JMAX,ISTEP,DOUBLE,NODE,STI, MOD, REAL MODIFR
DIMENSION PRFD(15),CORR(15)

22223 CONTINUE
IF(ISKIP(J) .GT. 4) GO TO 20
C SAVE ORIGINAL PRED-CORR ERROR CRITERIA IN CASE OF LATER CHANGE
PCERS=PCER
C SET ITER =0 INITIALLY
ITER=0
160 CALL RKUTTA
RETURN
20 N=8
K=N
C FIFTH DERIVATIVE FOR THE ERROR TERM
NZPC=N
TTEMP=T(N)
CALL DERIV5(TTEMP,ZPC,W,FDOT,T,J,NZPC)
C PREDICTOR (J) IE FOR THE JTH EQN
PRED(J)=W(N,J)+(DELT/24.)*((55.*FDOT(N,J)-59.*FDOT(N-1,J)+37. *
1 FDOT(N-2,J)-9.*FDOT(N-3,J)) +((251./720.)*DELT**5*ZPC
MODIFR=PRED(J)
W(N+1,J)=MODIFR
IF(J.EQ.1) T(N+1)= T(N)+DELT
TTEMP=T(N+1)
ITERAT=0
600 CONTINUE
N=5
K=N
CALL FUNCT
NZPC=N
CALL DERIV5(TTEMP,ZPC,W,FDOT,T,J,NZPC)
N=8
C CORRECTOR (J) IE FOR THE JTH EQN
CORR(J)= W(N,J)+(DELT/24.)*((9.*FDOT(N+1,J)+19.*FDOT(N,J)-5.*FDOT(N-
1,J)+FDOT(N-2,J)) -((19./720.)*DELT**5*ZPC
C ITERATE ON CORRECTOR TO FIND CORRECT VALUE OF FUNCTION
WTEMP=W(N+1,J)
W(N+1,J)=CORR(J)
ITERAT=ITERAT+1
IF(ITERAT.LT.1000) GO TO 700
IF(ITER, EQ. 5) CALL DUMP
ITER=ITER+1
WRITE(6,1100C) PREO(J),CORR(J),FCOT(9,J),
I TTTEMP,J
700 CONTINUE
CHECK=ABS(WTEMP-W(N+1,J))
IF(CHECK .GT.1.E-07*ABS(W(N+1,J)))GO TO 600
11000 FORMAT(1HO,4D16.8,15)
ZIP(J)= PRED(J) -CORR(J)
IF(CORR(J) .NE. 0.)GO TO 300
Z(J)=0.
GO TO 305
300 Z(J)=PRED(J)/CORR(J)
D-26
305  CONTINUE
   CALL ERROR
   IF(J.LT.JMAX.AND.NONLIN.LE.1) RETURN
C THE IFS ON NONLIN SET UP THE ITERATION OF A SYSTEM OF NONLINEAR EQUATIONS
   IF(NONLIN.EQ.1) CALL SYSIT
135  CONTINUE
   IF(NONLIN.GE.2) CALL SYSIT(600,125,135)
125  CONTINUE
C STEP SIZER CHECKS ERROR TERMS HERE
C ERROR CRITERIA DETERMINES WHETHER TO DOUBLE OR HALF
C SI AND SAVE CHECK STABILITY AND TRUNCATION ERROR RESPONSE
   IF(SI.GT.SII) GO TO 211
C PCER IS THE ERROR CRITERIA DEPENDING ON THE PRED AND CORR.
C PCER IS MADE STRONGER IF SI=0. (DFDY=0) SINCE IT IS THEN THE ONLY
C ERROR CRITERIA FOR CHECKING
   IF(SI.EQ.0.AND.PCER.GT..0001) PCER=.0001
   IF(ARS(SAVE).LE.PCER) GO TO 50
211  CALL STPSIZ
   IDUBLE=1
   NODBLE=0
   GO TO 106
50   IF(NODBLE.EQ.1) GO TO 105
   IF(ISTEP.LE.5)GO TO 105
   IF(SI.GT.5*SII.AND.SAVE.GT.5*PCER) GO TO 105
   CALL DOUBLE
106  IF(DEL.EQ.DEFLT) GO TO 105
   DELT=DEL
   SAVE1=1.
   GO TO 110
105  CONTINUE
   DO 115 J=1,JMAX
   DO 115 N=1,8
   W(N,J)=W(N+1,J)
   IF(J.EQ.1) T(N)=T(N+1)
   FDOT(N,J)=FDOT(N+1,J)
115  CONTINUE
   ISTEP=ISTEP+1
   J=JMAX
116  CONTINUE
   SAVE1=0.
   IDUBLE=1
   IF(ISTEP.GT.100) ISTEP=5
C RESET PCER IN CASE OF CHANGE
   PCER=PCERS
C RESET ITER IF CHANGE HAS OCCURRED
   IF(ITER.GT.0) ITER=0
C THE FOLLOWING STATEMENTS CHECK THE STEP SIZE AND LIMIT DELT WHEN
C DELMAX.GT.0, IE., STEP LIMITER
   IF (DELMAX.EQ.0) GO TO 110
   IF(DELT.EQ.DELMAX.OR.2.*DELT.GT.DELMAX) NODBLE=1
110  CONTINUE
22221  CONTINUE
   RETURN
END
SUBROUTINE SYSIT
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/ZILCH/ FDOT(10,15),W(10,15),CST(15),YY(10,15),Z(15),T(10),
1 ZIP(15),FDOTS(10,15),PCHR ,DELT,DELMAX,DEL,SAVE,SI,SAVE1, SII,
2 ISKIP(15),K,N,J,JMAX,ISTEP,IDOUBLE,ICHECK,NGDUBLE,NONLIN,ITERAT
DIMENSION YDOT(15)
C SYSIT PERFORMS ITERATION ON SYSTEMS OF NONLINEAR DIFFERENTIAL EQN
1 IF(NONLIN .EQ. 3) NONLIN=4
2 DO 3 J=1,JMAX
3 YDOT(J)=FDOT(9,J)
3 CALL FUNCT
DO 5 I=1,JMAX
CHECK=DABS(YDOT(I)-FDOT(9,I))
IF(CHECK .GT. 1.E-07*DABS(FDOT(9,I))) GC TO 1
5 CCNTINUE
IF(NONLIN .EQ. 3) GO TO 6
J=C
IF(NONLIN .EQ. 4) RETURN 3
NCNLIN=2
RETURN
ENTRY SYSIT1(**,*,*)
J=J+1
IF(J .LE. JMAX) RETURN 1
C ITERATION TO CHECK COMPATABILITY OF FDOT WITH NEW VALUES FOR W
NCNLIN=3
GO TO 2
6 CCNTINUE
NCNLIN=1
J=JMAX
RETURN 2
END
SUBROUTINE STPSIZ
IMPLICIT REAL*8 (A-H,O-Z), COMPLEX/ZILCH/
COMMON/ZILCH/ FDOT(10,15), W(10,15), CST(15), YY(10,15), Z(15), T(10),
IZIPI(15), FDOTS(10,15), PDLST, D,PCER, DEL, DELMAX, DEL, SAVE, SI, SAVE1, SII,
IZSPI(15), FDOTS(10,15), PCER, OELT, CELMAX, DEL, SAVE, SI, SAVE1, SII,
ISKIP(15), K, N, J, JMAX, ISTEP, IDUBE, ICHECK, NDOBLE, NONLIN, ITERAT
DIMENSION WS(8), TX(8), FS(8), A(10), B(10)
DEL=DELT*.5
IS=2
IORDER=4
NOPRNT=0
IMAX=4

C IF DOUBLING AND HALVING OCCURS ON SUCCESSIVE STEPS USE ACTUAL
C VALUES FOR W(7, J) AND FDOT(7, J)
IF(IDUBLE .EQ. 1) GO TO 20
DO 25 J=1, JMAX
  W(4, J)=YY(4, J)
  W(5, J)=YY(5, J)
  W(6, J)=YY(6, J)
  W(7, J)=YY(7, J)
  FDOT(2, J)=FDOTS(2, J)
  FDOT(3, J)=FDOTS(3, J)
  FDOT(4, J)=FDOTS(4, J)
  FDOT(5, J)=FDOTS(5, J)
  FDOT(6, J)=FDOTS(6, J)
  FDOT(7, J)=FDOTS(7, J)
  FDOT(9, J)=FDOTS(9, J)
25 CONTINUE
GO TO 30

20 CONTINUE
DO 1 J=1, JMAX
100 FORMAT(IX, 8D15.8)
DO 11 I=5, 8
  WS(I-4)=W(I, J)
  FS(I-4)=FDOT(I, J)
11 CONTINUE
MM=4
dc 2 m=2, 6, 2
mm=MM+1
w(M, J)=W(MM, J)
fdot(M, J)=FDOT(MM, J)
3 CONTINUE
MM=1
dc 4 m=3, 7, 2
w(M, J)=A(MM)
fdot(M, J)=R(MM)
mm=MM+1
4 CONTINUE
1 CONTINUE
30 CONTINUE
IORDER=2
IMAX=8
DO 5 K=5, 7

D-29
XI = T(K) + DEL
CALL SINTP (XI, T1, T1, IMAX, IS, K, IORDER, UX, NOPRNT)
TX(K) = UX
CONTINUE
MM = 5
I = 4
DO 6 K = 1, 7
GO TO (15, 16, 15, 16, 15, 16, 15, K)
T(K) = TX(I)
I = I + 1
GO TO 6
T(K) = T(MM)
MM = MM + 1
CONTINUE
J = JMAX
RETURN
ENTRY DOUBLE
C STEP SIZE = DEL * 2
DEL = DEL * 2.
DO 55 J = 1, JMAX
DOUBLE = 2
YY(4, J) = W(4, J)
YY(5, J) = W(5, J)
YY(6, J) = W(6, J)
YY(7, J) = W(7, J)
FOOTS(2, J) = FDOT(2, J)
FOOTS(3, J) = FDOT(3, J)
FOOTS(4, J) = FDOT(4, J)
FOOTS(5, J) = FDOT(5, J)
FOOTS(6, J) = FDOT(6, J)
FOOTS(7, J) = FDOT(7, J)
FOOTS(9, J) = FDOT(9, J)
DO 55 I = 1, 3
GO TO (65, 70, 75, I)
N = 7
NN = 6
GO TO 80
N = 6
NN = 4
GO TO 80
N = 5
NN = 2
CONTINUE
W(N, J) = W(NN, J)
IF (J .EQ. 1) T(N) = T(NN)
FDOT(N, J) = FDOT(NN, J)
CONTINUE
ISTEP = 1
J = JMAX
RETURN
ENC

D-30
SUBROUTINE SINTERP (A, B, C, IMAX, IS, IORDER, D, NOPRNT)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION B(100), C(100)
DIMENSION X(100), Y(100), NUM(20), DENOM(20, 20), U(20)
REAL NUM, NUMT
X=1
DO 60 K=1, IMAX
Y(K)=B(K)
X(K)=C(K)
60 CONTINUE
IF(IORDER .LE. 1) IORDER=2
C SEARCH AND INTERPOLATE ROUTINE
C IS=0 FOR BOTH SEARCH AND INTERP
C IS=1 FOR SEARCH ONLY
C IS=2 FOR INTERP ONLY.
C NOPRNT=3 SUPPRESSES THE WRITTING OF THE
C DIAGNOSTIC MESSAGE, NOPRNT RETURNS A VALUE OF -1, 0, OR 1 DEPENDING
C ON WHETHER THE INTERPOLATION TOOK PLACE OUTSIDE THE TABLE, LOWER
C (-1), UPPER END (0), OR INSIDE THE TABLE (0)
IF(IS .EQ. 2) GO TO 20
I=1
IF(XI .LT. X(MAX) .AND. XI .GT. X(1)) GO TO 10
IF(XI .GT. X(1)) GC TO 15
I=1
GO TO 20
15 I=IMAX
GO TO 20
10 IF(XI .LE. X(I+1) .AND. XI .GT. X(I)) GO TO 20
I=I+1
GO TO 10
20 CONTINUE
IF(IS .EQ. 1) RETURN
C TABLE WITHIN A TABLE INDEX LOCATION
IT=1
IF(IS .EQ. 2 .AND. I .GT. 1) IT=I
IF(IORDER .GT. IMAX) IORDER=IMAX
IORDRS=IORDER
22 IF((I+IORDER-1) .LE. IMAX) GO TO 21
IORDER=IORDER-1
IF(IORDER .NE. 1) GO TO 22
46 IF(NOPRNT .EQ. 3) GO TO 45
WRITE(6,101) Y(II), X(II), Y(MAX), X(MAX), XI, IMAX
101 FORMAT(IHO, 'AN ERRONEOUS VALUE HAS BEEN SUPPLIED TO SINTERP, IE XI
1 IS OUTSIDE THE TABLE BEING USED, Y=', D15.8, 'X=', D15.8, 'YMAX=
2 ', D15.8, 'XMAX=', D15.8, 'XI=', D15.8, 'MAX=', I15, '/ , THE APPROPRIATE
3 END POINT IS USED FOR THE INTERPOLATED VALUE')
45 CONTINUE
II=IMAX
IF(XI .LT. X(MAX)) II=1
UX=Y(II)
IORDER=IORDRS
NOPRNT=-1
IF(II .EQ. IMAX) NOPRNT=1
D=UX
RETURN
21 IF(XI .LT. X(II)) GO TO 46
C SAVE THE ORIGINAL INDEX I TO RETURN TO CALLING PROGRAM
IRETN=I
C THE INDEX I REPRESENTS THAT POINT IN THE TABLE WHERE THE
C POLYNOMIAL WAS STARTED

D-31
IF(IORDER .NF. IORDRS) I=I-(IORDRS-IORDER)
C NTH ORDER INTERPOLATOR
50 CONTINUE
C RETURN IORDER TO ORIGINAL VALUE
IORDER=IORDRS
NOPRNT=0
ISAVE =I
DO 25 II=1,IORDER
I=ISAVE+II-1
NUM(II)=XI-X(I)
DO 25 LL=1,II
L=LL+ISAVE-1
DENOM(II,LL)=X(I)-X(L)
C SET DIAGONAL =1 SO THAT DIVISION BY 0 DOES NOT OCCUR
IF(I .EQ. L) DENOM(II,LL)=1.
25 CONTINUE
DO 26 I=1,IORDER
TERM =1.
DO 27 L=1,IORDER
II=L
IF(I .NE. L) GO TO 30
NUMT=NUM(II)
NUM(II)=1.
30 TERM=TERM*(NUM(II)/DENOM(I,L))
IF(I .EQ. L) NUM(II)=NUMT
27 CONTINUE
U(I)=TERM
26 CONTINUE
I=ISAVE
C VALUE OF INDEP VAR AT XI
UX=0.
DO 40 K=1,IORDER
UX=UX+U(K)*Y(I)
I=I+1
40 CONTINUE
I=IRETN
D=LX
RETURN
END
** DATA DMS,N,GCB19
\*/

NASA-GSFC