NASA TM X- 65991

X-742-72-229

PREDICTION OF PRESSURE FLUCTUATION IN SOUNDING ROCKETS AND MANIFOLDED RECOVERY SYSTEMS

JOHN F. LAUDADIO

JUNE 1972

GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-742-72-229

PREDICTION OF PRESSURE FLUCTUATION IN SOUNDING ROCKETS AND MANIFOLDED RECOVERY SYSTEMS

John F. Laudadio

June 1972

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

CONTENTS

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, 1bm
$\mathring{m}$	Mass flow rate, lbm/sec
P	Pressure, lbf/ft^2
R	Tube radius, ft
R_e	Reynolds Number
ρ	Density, lbm/ft^2
T	Temperature, °R
t	Time, sec
U	Internal energy, BTU/lbm
$\underline{V}$	Volume, ft^3
$\overline{V}$	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

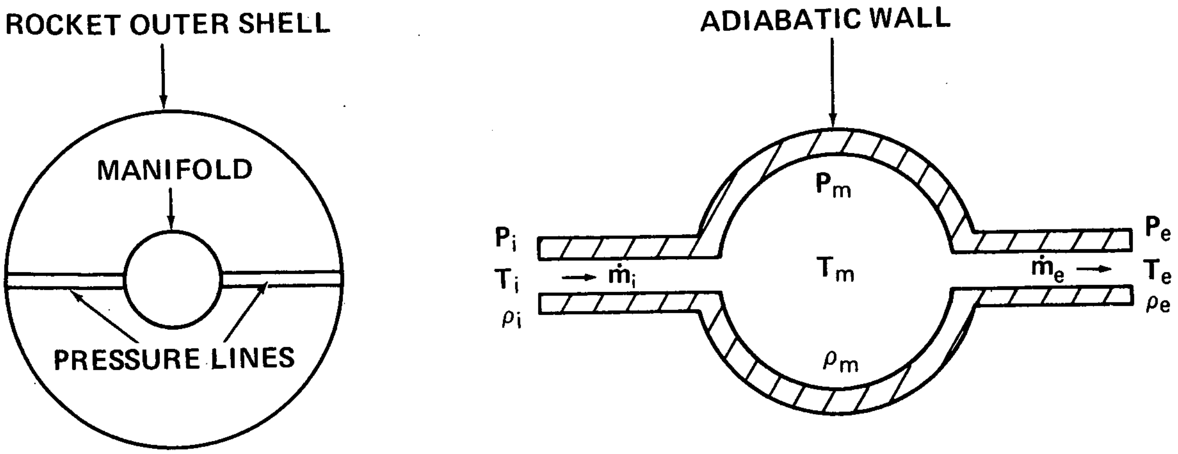


Figure 1

P_m = Pressure in the manifold
$\dot{m}_{i,e}$ = Mass flow rate

From the first law of thermodynamics for an open system:

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

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

$$U_m = m_m C_v T_m \qquad \text{EQN 2}$$

$$h_m = C_p T_m \qquad \text{EQN 3}$$

Expanding the term describing the changes in the manifold.

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

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

$$U_m = m_m C_v T_m \qquad \text{EQN 2}$$

and

$$h = C_p T \qquad \text{EQN 5}$$

$$\frac{dU_m}{dt} = C_v\ (\dot{m}_m T_m + \dot{T}_m m_m) \qquad \text{EQN 6}$$

$$h_{flow} + \frac{\bar{V}^2{}_{flow}}{2g} = h_{source} = C_p T_{source} \qquad \text{EQN 7}$$

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

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

Substituting Equations 6 and 7 into Equation 8

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

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

$$\dot{T}_m = \frac{C_p\ \Sigma\ {}_n\dot{m}_n T_n + Q - C_p T_m \dot{m}_m}{C_v m_m} \qquad \text{EQN 10}$$

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.

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

where ρ_a and $\bar{V}_a$ are mean values and i may be replaced by e where applicable.

ρ_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:

$$\bar{V}_a = \left[\frac{P_i - P_m}{f\rho_a \frac{L}{R} + \rho_a{}^2\left(\frac{1}{\rho_m} - \frac{1}{\rho_i}\right)}\right]^{1/2} \qquad \text{EQN 12}$$

Knowing ρ_a and assuming a value for $\overline{V}_a$, a Reynolds number and friction factor f is calculated for each tube and then a new $\overline{V}_a$ is calculated from Equation 12. $\overline{V}_a$ (calculated) is compared to $\overline{V}_a$ (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:

$$f = 0.0008 + 0.05525/Re^{0.237} \qquad Re \geqslant 100000 \qquad \text{EQN 13}$$

$$f = 0.0791/Re^{0.25} \qquad 1185 < Re < 100000 \qquad \text{EQN 14}$$

$$f = 16/Re \qquad Re \leqslant 1185 \qquad \text{EQN 15}$$

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

$$\dot{m}_a = \rho_a\overline{V}_a A \qquad \text{EQN 16}$$

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

Equation 16 is used to compute the mass flow rate, $\dot{m}_n$, 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

$$\dot{m}_a = C_q A \left(\frac{\Delta P}{\rho}\right)^{1/2} \qquad \text{EQN 18}$$

where C_q may be input as a variable and ρ 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 manifolded 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.

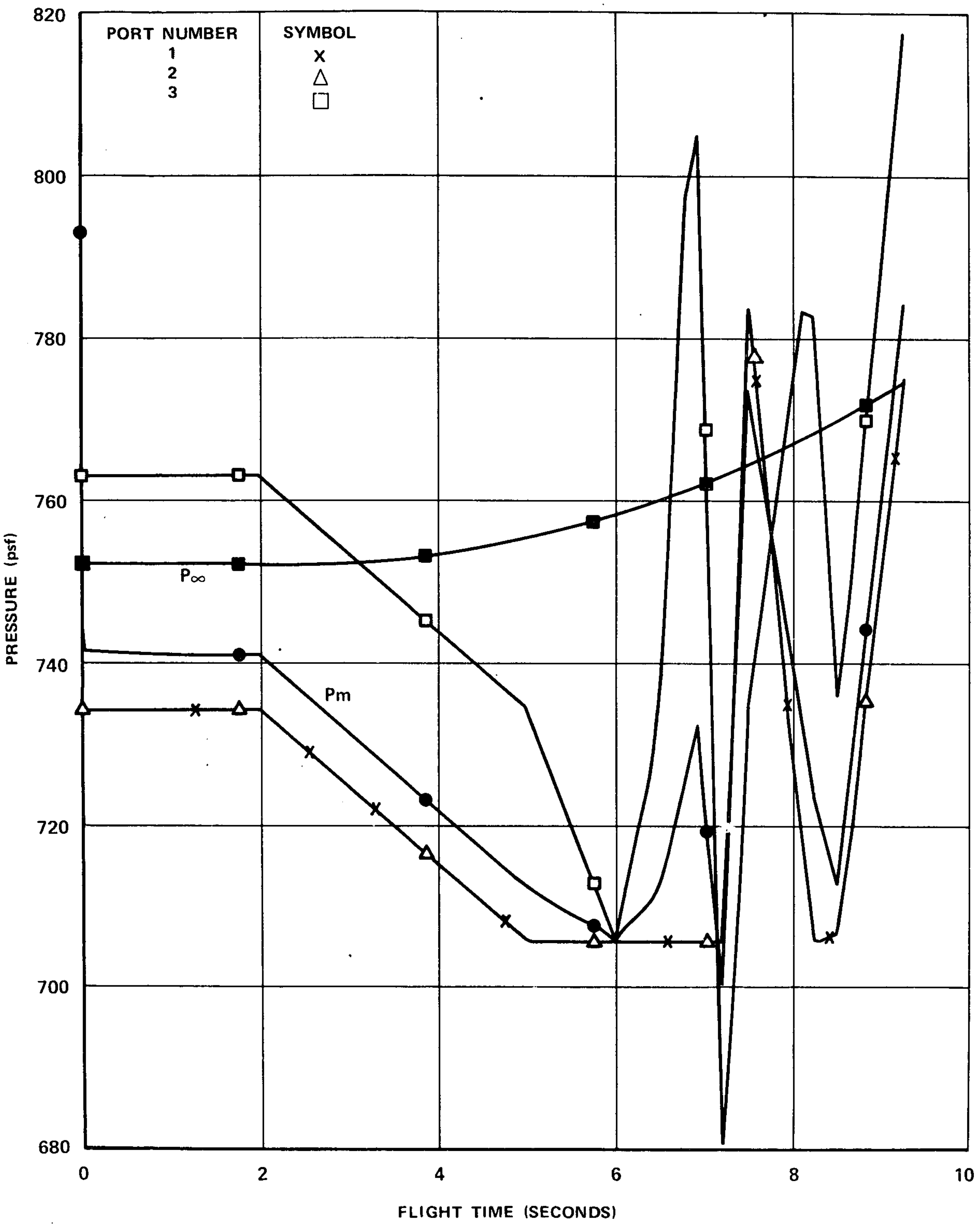


Figure 2a. Manifold Pressures Aerobee 350 Drop Test 11/15/71

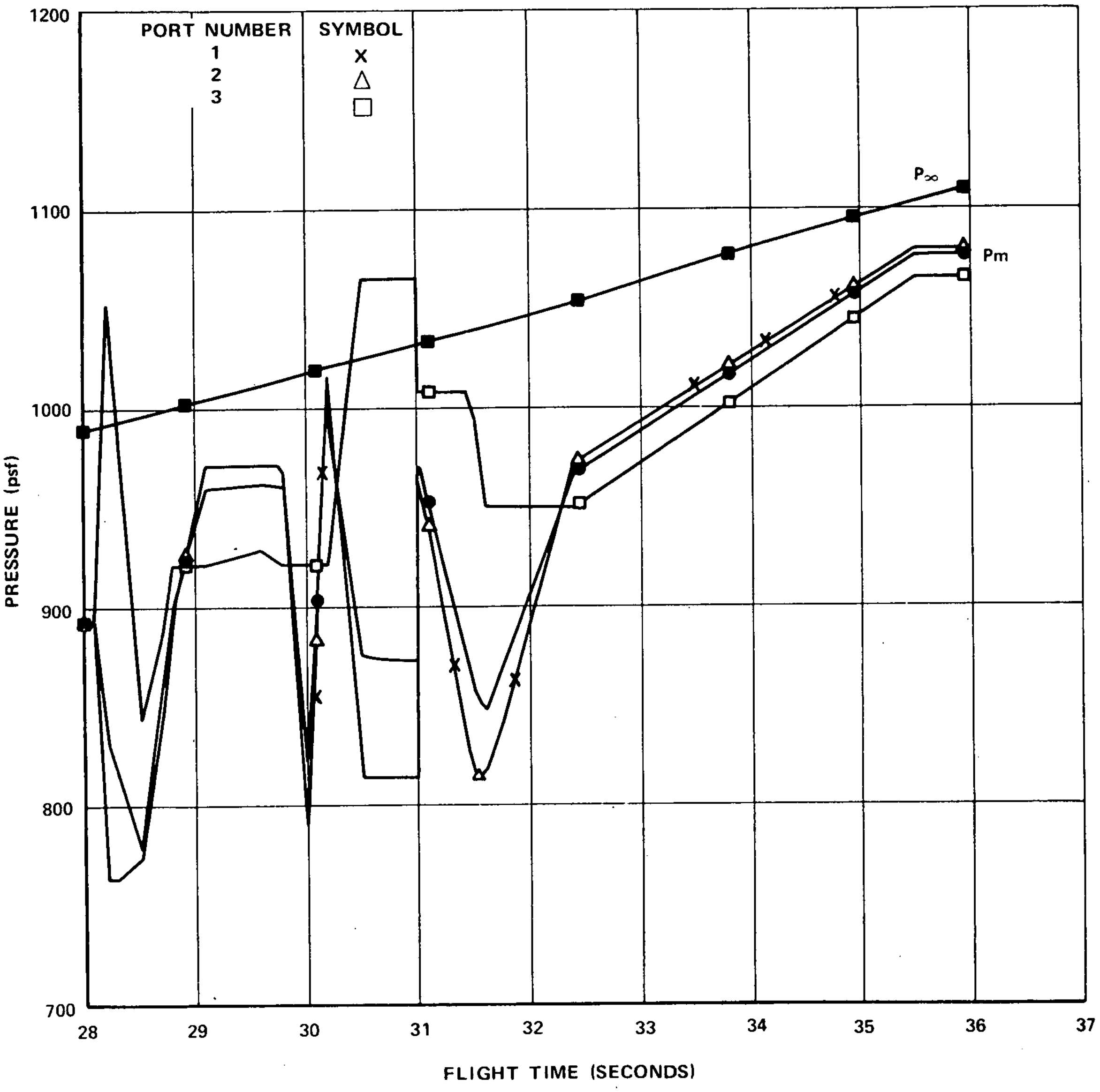


Figure 2b. Manifold Pressures Aerobee 350 Drop Test 11/15/71

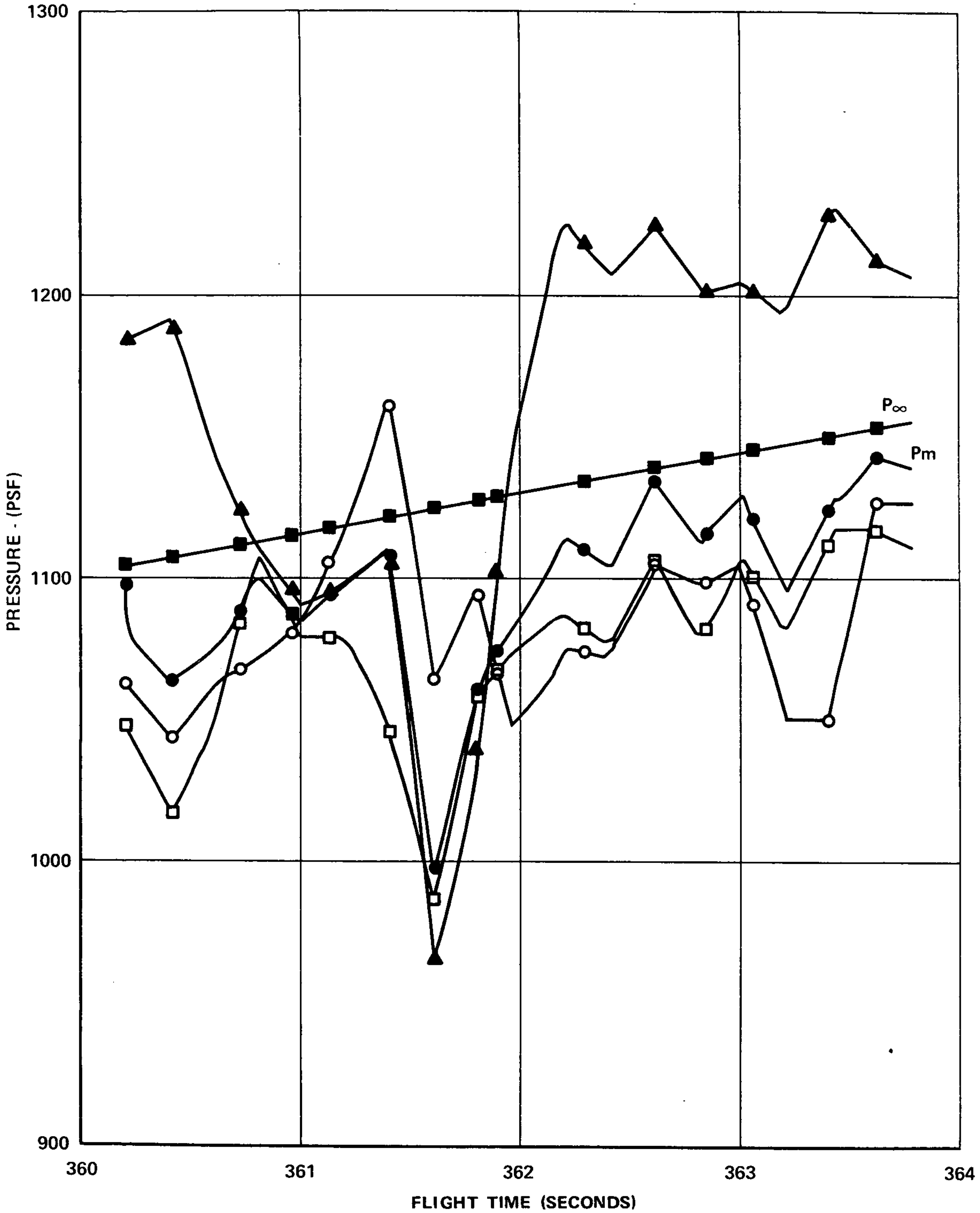


Figure 3a. Manifold Prog Calc. Aerobee Flt 17.05

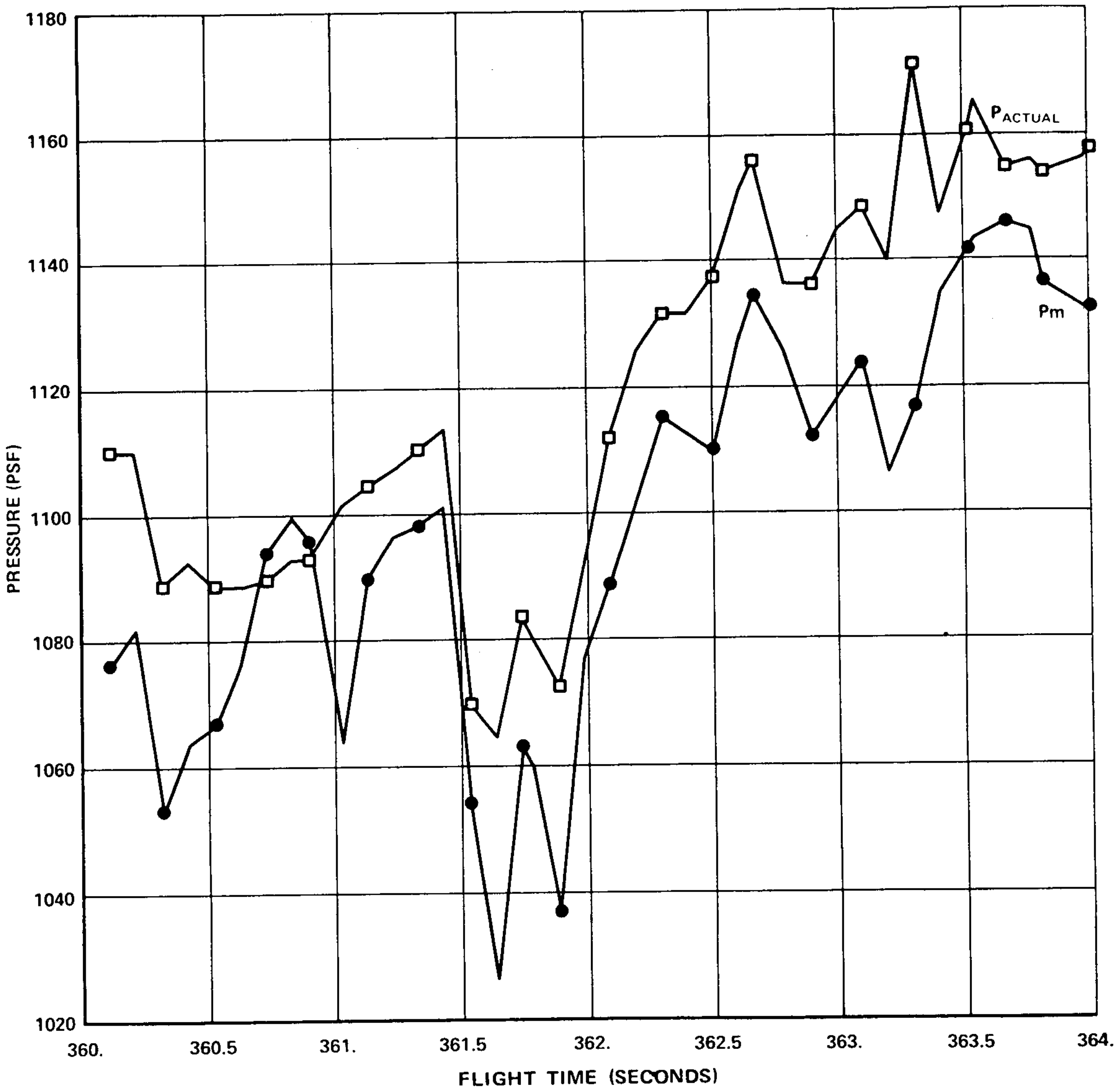


Figure 3b. Measured & Predicted Pressure for 17.05 Recovery

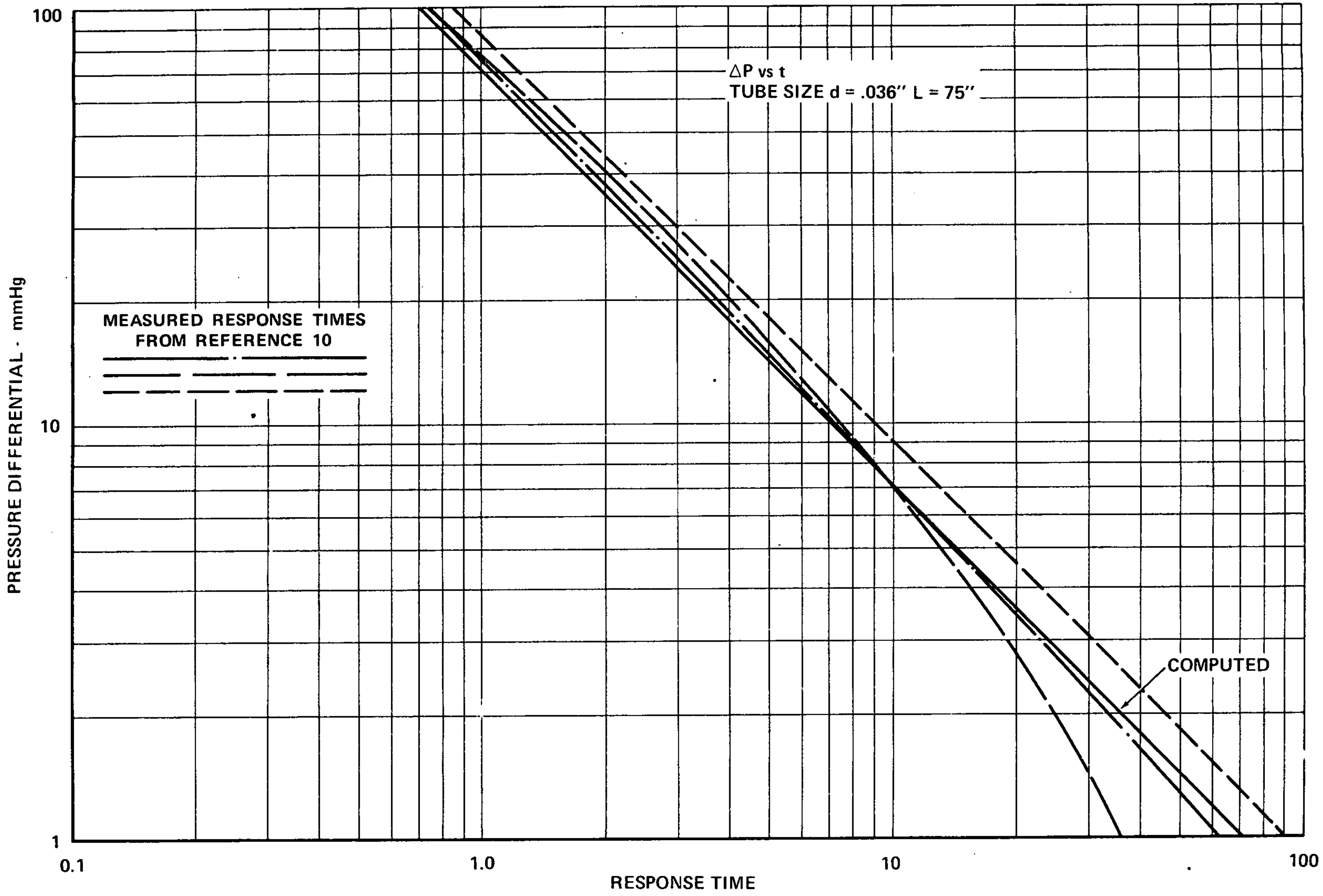


Figure 4. Pressure Response Through a Tube

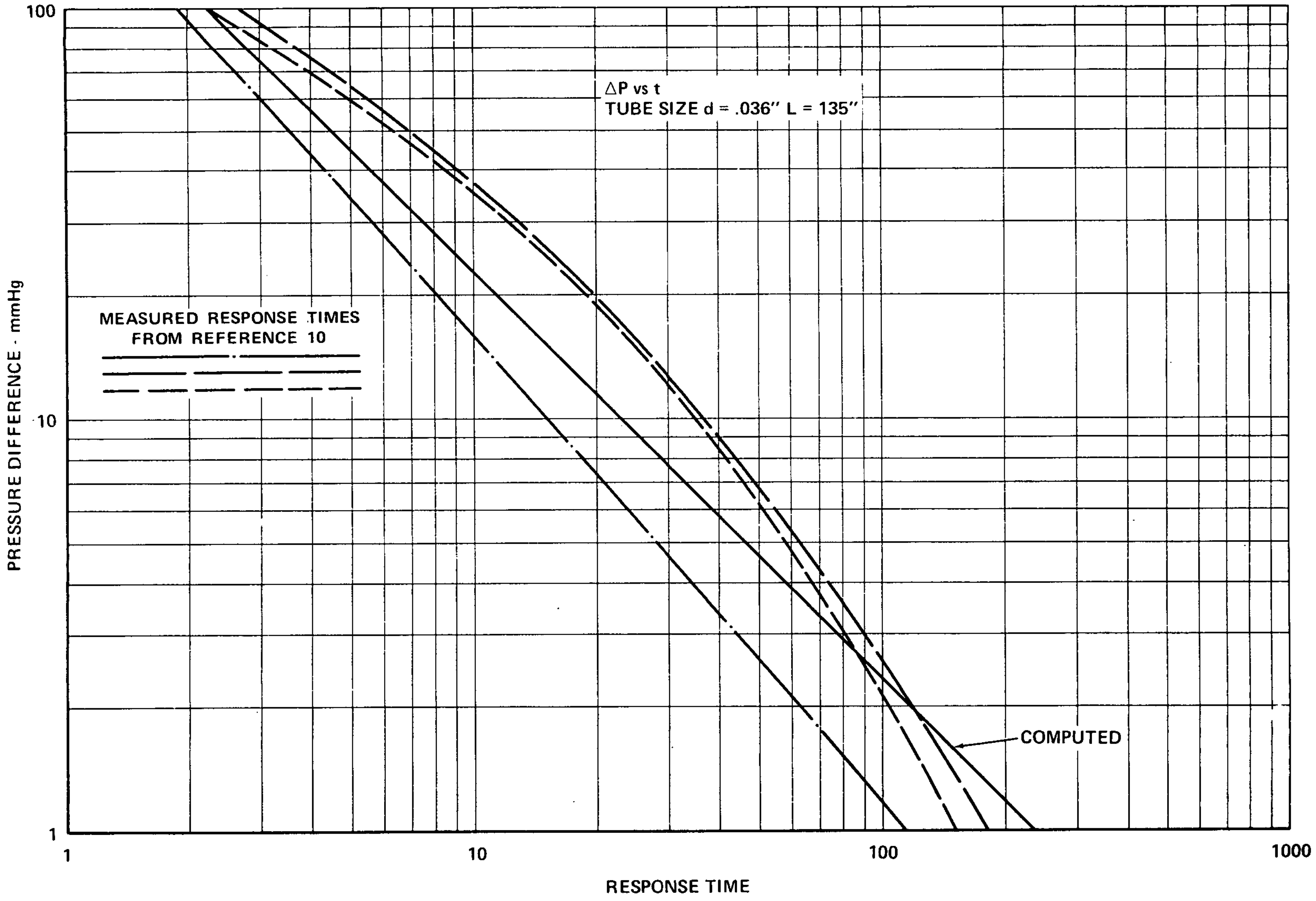


Figure 5. Pressure Response Through a Tube

REFERENCES

1. Thermodynamics by Van Wylen, Wiley, January 1962

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

3. An Introduction to Fluid Mechanics and Heat Transfer, Second Edition, by J.M. Kay, Cambridge University Press, 1963

4. Elements of Gas Dynamics by Liepmann and Roschko, Wiley, 1962

5. Numerical Methods for Scientists and Engineers by Stanton, Prentice Hall, 1963

6. The Numerical Treatment of Differential Equations by Collatz, Springer-Verlag Ohg. Berlin Gottingen Heildeberg, 1960

7. Principles of Fluid Mechanics by Eskinazi, Allyn & Bacon, Inc., 1963

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

9. Heat Transfer by Holman, McGraw Hill, 1963

10. The Response Times of Typical Transducer - Tube Configurations for the Measurement of Pressures in High Speed Wind Tunnels by M.J. Larcombe and J.W. Peto, NASA N67-21204

11. A Method for the Determination of the Time Lag in Pressure Measuring Systems Incorporating Capillaries by Archibald R. Sinclair and A. Warner Robins, NACA RM A50L07

12. Results of A Subsonic Pressure Test Conducted on A 27.25 Percent Scale Model of the Aerobee 350 Recovery Package by C.G. Thomas, June 1968, Fairchild Hiller TSD Report No. ER-115

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.

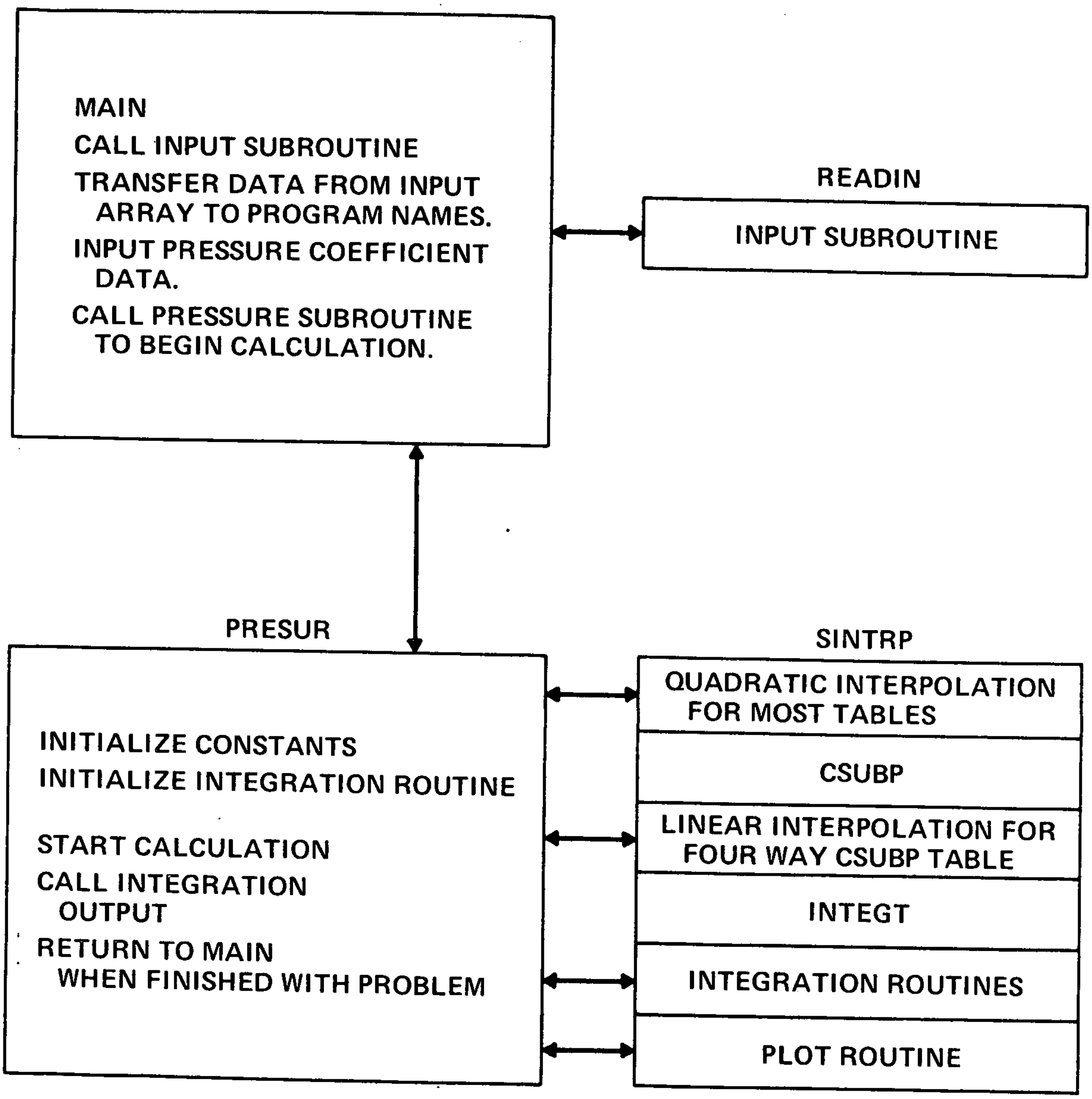
$$\dot{T}_m = \frac{C_p \; \Sigma \; {}_n\dot{m}_n T_n \; + \; Q \; - \; C_p T_m \dot{m}_m}{C_v m_m} \qquad \text{EQN 10}$$

$$\frac{d}{dt} = \frac{\dot{m}_m}{V_m} = \Sigma_n \frac{\dot{m}_n}{V_m} \qquad \text{EQN 17}$$

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
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

PROGRAM VARIABLE NAME AND INPUT NAME	DEFINITION
ALT	Altitude Table — Ft.
CQ	Discharge Coefficient table for orifice flow.
FRENO	The independent variable for the CQ table. The function may be either Reynolds Number, the square root of Reynolds Number or pressure ratio where $P/P_0 > 1$.
NCQ	The number of pairs of points in the CQ-FRENO table.
INDPVR	Indicates the type of independent variable FRENØ = 0 = $(\text{ReNo})^{1/2}$, - 1 = Re No, 1 = P/P_0 ($P/P_0 > 1$).
PRINTFQ	The time interval between printed output steps — should always be used, otherwise every step will be output.
BDUMP	BDUMP is input equal to 1 for all name load input to be printed, otherwise it is neglected.
TINIT	Initial time.
TEND	Time at which program is to stop.
NOPTS	Number of points in the Altitude-Velocity-Time Table, i.e., number of time points.
ALPHA	Input option to have the angle-of-attack vary with time. ALPHA is input only if this option is to be implemented, with ALPHA = Number of points in the α - T table (maximum of 10).
ALPHAA	Name of the angle-of-attack array.
TALPH	Name of time array associated with ALPHAA.
CASES	Number of cases to be run.

PROGRAM VARIABLE NAME AND INPUT NAME	DEFINITION
VR	Manifold Volume – input as inches^3, converted internally to feet^3.
NEWFMT	Used to trigger NAMELOAD subroutine to read in user input case label.
OMEGA	Roll rate in-CPS (optional input).
PM	Initial Pressure in the manifold – PSF.
RO	Initial air density in the manifold lbm/ft^3.
TM	Initial air temperature in the manifold ° R. Note: Only one of RO or TM need be input; the other is calculated in the program.
PLTCON	= 1 Enables user to continue plotting from one case to the next, i.e., could also be used when SAVE = 1 is used.
IPLOT	The array in which the variables to be plotted are specified.
PLOTFQ	Plot every PLOTFQ number of points (see A7 for explanation).
DELT	Initial time step, default = 0.0625.
DELMAX	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.
IERCRT	Error criteria for the numerical integrator.

ERCRIT(1)
ERCRIT(2)
	integration–preset.

PROGRAM VARIABLE NAME AND INPUT NAME	DEFINITION
JMAX	Number of equations to be solved—preset.
NØNLIN	Non-Linear Integration Option—preset.
ICP	The number of different CP tables to be input, i.e., succeeding cases may use either the same or a different CP table.
KT	The number of tubes or orifices connected to the manifold, default = 1, Max No. = 4.
NØQ	Option selector for tube or orifice flow default = tube, NØQ = 2 for orifice.
RT	Recovery Temperature factor for aero heating in boundary layer. (Usually around 0.9).
SAVE	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.
OPTSEL and CPCNTL	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.
IXMAX	Number of Mach numbers or times to be input.
IYMAX	Number of angles-of-attack to be input.
IZMAX	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.

PROGRAM VARIABLE NAME AND INPUT NAME	DEFINITION
PIPSIZ	Dummy array containing pipe or orifice dimensions (inches). PIPSIZ 1 = Radius (1) 2 = Length (1) 3 = Radius (2) 4 = Length (2) 5 = Radius (3) 6 = Length (3) 7 = Radius (4) 8 = Length (4)

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:

OPTION NUMBER/ LETTER	DESCRIPTION
0	Tube flow.
2	Orifice flow.
a	Trajectory input required, altitude, velocity, time.
b	Ambient temperature used at port.
c	Recovery temperature used at port. TR = TA (1+2*RT*M).
d	Pressure forcing function $C_p = \frac{\Delta P}{q} = f(M, \alpha, \phi)$.
e	Pressure forcing function $C_p = \frac{P_1}{P_o} = f(M, \alpha, \phi), \left(\frac{P_1}{P_o} > 1\right)$.
f	Pressure forcing function $C_p = P_1 = f(M, \alpha, \phi)$, lb/ft^2.
g	Pressure forcing function $C_p = \frac{\Delta P}{q} = f(t, \alpha, \phi)$.
h	Pressure forcing function $C_p = \frac{P_1}{P_o} = f(t, \alpha, \phi), \left(\frac{P_1}{P_o} > 1\right)$.
i	Pressure forcing function $C_p = P_1 = f(t, \alpha, \phi)$ (lb/ft^2).
j	Variable Discharge Coefficient CQ = f (Re$^{1/2}$).
k	Variable Discharge Coefficient CQ = f (Re).
l	Variable Discharge Coefficient $CQ = f\left(\frac{P_1}{P_2}\right), \left(\frac{P_1}{P_o} > 1\right)$.
n	Input constant roll rate for vehicle simulation, cps.
o	Output all input data except C_p table which is always output automatically.

OPTION NUMBER/ LETTER	DESCRIPTION
p	Plot up to 10 dependent variables as a function of any other variable.
q	Input angle-of-attack as a function of time to be used by the C_p table. This table is needed only when C_p is also a function of angle-of-attack.
r	Input non-symmetric C_p tables, i.e., $0 \leqslant \emptyset \leqslant 360$.
s	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.
t	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.
u	Save the manifold pressure and temperature and time to use in the next case, i.e., internal initialization of next case when multiple C_p tables are required because there is too much C_p data to input to one case.

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.

OPTION	INPUT NAME(S)	OPTION SELECTION OR INPUT VALUE
0	NØQ	0 (default)
2	NØQ	2
a	ALT, VEL, TF	Input data
b	RT	0 (default)

OPTION	INPUT NAME(S)	OPTION SELECTION OR INPUT VALUE
c	RT	Equal to the recovery temperature factor
d	OPTSEL, CPCNTL	0, 0 (default)
e	OPTSEL, CPCNTL	-1, 0
f	OPTSEL, CPCNTL	-2, 0
g	OPTSEL, CPCNTL	0, -1
h	OPTSEL, CPCNTL	-1, -1
i	OPTSEL, CPCNTL	-2, -1
j	INDPVR	0 (default)
k	INDPVR	-1
l	INDPVR	1
n	OMEGA	Roll rate cps
o	BDUMP	1
p	IPLOT, NCURVS, PLTCON	See A7
q	ALPHA	Number of pairs of points in table. Then the table is input under arrays.

Note: If an option is a default selection it need not (but may) be included in the input stream.

OPTION	INPUT NAME(S)	OPTION SELECTION OR INPUT VALUE
r and d	OPTSEL, CPCNTL	0, 1
r and e	OPTSEL, CPCNTL	-1, 1
r and f	OPTSEL, CPCNTL	-2, 1
r and g	OPTSEL, CPCNTL	0, 2

OPTION	INPUT NAME(S)	OPTION SELECTION OR INPUT VALUE
r and h	OPTSEL, CPCNTL	-1, 2
r and i	OPTSEL, CPCNTL	-2, 2
s	OPTSEL, CPCNTL	0, -3
t	TM	input -TM
u	SAVE	1

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. "NØPTS" 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

M_1 or t_1
a_1
ϕ_1, C_p
ϕ_2, C_p
a_2
ϕ_1, C_p
ϕ_2, C_p
M_2 or t_2
a_1
ϕ_1, C_p
ϕ_2, C_p
a_2
ϕ_1, C_p
ϕ_2, C_p
Etc.

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 ϕ-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:

NAME LOAD OPTION NUMBER	OPTION
0	Read in variable names, and data. Return to calling program, i.e., manifold main program.
1	Option 0 plus read another option card.
2	Read in an array name, and data. Return to calling program.
3	Option 2 plus read another option card.
6	Read in an array name specific array locations, and data. Return to calling program.
7	Option 6 and read another option card.
5	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 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 <u>or</u> arrays.

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, α, Ø or t, α, Ø tables which are loaded in regular FORTRAN formats as described in A5.

CARD INPUT DETAILS

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

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 IPLOT — Name of the input array in which the plotted values are specified.

The following values may be plotted:

Index	Name
1	Time
2	Manifold Pressure
3	Port pressure 1
4	Port pressure 2
5	Port pressure 3
6	Port pressure 4

GODDARD SPACE FLIGHT CENTER

FORTRAN CODING RECORD

PROGRAM MANIFOLD PROGRAM		PUNCHING INSTRUCTIONS	GRAPHIC	PAGE OF
PROGRAMMER JOHN LAUDADIO	DATE		PUNCH	CARD ELECTRO NUMBER*

SAMPLE INPUT

```
    1   10    2
(' MANIFOLD ANALYSIS OF AEROBEE FLT 17.05 ,5/25/70')
VR      PM      TM      TINIT   TEND    BDUMP   PRNTFQ  CASES   KT      NØPTS
10.     1080.   520.    360.2   362.    1.      .01     2.      3.      21.
    3    6
IPLOT
1.      2.      3.      4.      5.      7.
   10    2    3
360.2
0.
0.             1048.6
120.           1062.9
240.           1183.7
5.                             Cp TABLE
0.             1048.6
120.           1062.9
240.           1183.7
360.4
  ↓
  etc.
```

Index	Name
7	Atmospheric pressure
8	Altitude
9	Velocity
11	Dynamic pressure
12	Mach Number
13	Angle-of-attack
14	Manifold temperature
15	Reynolds Number through port 1
16	Reynolds Number through port 2
17	Reynolds Number through port 3
18	Reynolds Number through port 4
19	Discharge coefficient for port 1
20	Discharge coefficient for port 2
21	Discharge coefficient for port 3
22	Discharge coefficient for port 4

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.

IPLOT (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 PRNTFQ may be any non-integer but only if PLOTFQ = PRNTFQ will every printed point be plotted.

A8 Program Output

OUTPUT NAME	UNITS
Altitude	Altitude of vehicle—ft
Velocity	Velocity of vehicle—ft/sec

OUTPUT NAME	UNITS
Time	Time
Manifold Press	lbf/ft^2
Pressure at each port	lbf/ft^2
Atmospheric Pressure	lbf/ft^2
QØ	Dynamic pressure—lbf/ft^2
Mach Number	Vehicle velocity/speed of sound at ALT
Alpha	Angle-of-attack—degrees
Manifold Temperature	Degrees R (Rankine)
Pressure coefficient at each port	Same as input
Roll rate	CPS
Mass flow rate (through each port)	lbm/sec
Reynolds Number (through each port)	Based on tube diameter
Internal and external ss	Speed of sound—ft/sec
Temperature (at each port)	Degrees R (T_{oo} or T_r) depending on input, i.e., if recovery factor is used
Phase angle at each port	In relation to free stream, degrees
Total mass change in manifold	lbm from t_o to $t_{present}$ (TINIT to TEND)
Velocity in each tube	Average gas velocity—ft/sec
Friction factor in each tube	Average friction factor in tube

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 —

Option Number	Description
0	Tube flow
a	Trajectory input
i	Pressure = f (t, α, Ø)
b	Atmospheric ambient pressure at each port
o	Output all input data
r (non-standard)	Input 360 degrees C_p Tables
p	Plot option for Port pressure 1, 2, 3, plus manifold and ambient pressure versus time
u (non-standard)	Save PM, TM and t to use in the next case

Option Implementation

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

The following data sheet is a listing of the program input for Example 1.

```
COLUMN 0        10        20        30        40        50        60        70        80
         1   10   2
('   MANIFOLD ANALYSIS OF AEROBEE FLT 17.05 ,5/25/70')
VR        PM        TM        TINIT     TEND      BDUMP     PRNTFQ    CASES     KT        NOPTS
10.       1080.     520.      360.2     362.      1.        .01       2.        3.        21.
         1    6
NCURVS    MODE      IPSKIP    SAVE      ICP       PLTCON
5.        0.        5.        1.        2.        1.
         1    2
OPTSEL    CPCNTL
-2.       2.
         3    6
IPLOT
1.        2.        3.        4.        5.        7.
         3   21
ALT
17008.9   16943.7   16878.9   16814.4   16749.9   16685.4   16621.2   16557.4   16492.9   16429.4
16365.9   16302.9   16240.2   16177.7   16115.1   16054.7   15993.4   15933.2   15873.2   15813.4
15754.7
         3   21
VEL
338.5     338.5     338.4     337.9     337.1     336.3     335.5     334.2     333.2     331.9
330.2     328.5     326.5     324.6     322.4     320.2     317.8     315.2     312.3     309.6
306.5
         3   21
TF
360.      360.2     360.4     360.6     360.8     361.      361.2     361.4     361.6     361.8
362.      362.2     362.4     362.6     362.8     368.      363.2     363.4     363.6     363.8
364.
         2    6
PIPSIZ
.0625     10.       .0625     10.       .0625     10.
        10    2    3
360.2
0.
0.                  1048.6
120.                1062.9
240.                1183.7
5.
0.                  1048.6
120.                1062.9
240.                1183.7
360.4
0.
0.                  1013.9
120.                1041.6
240.                1192.3
5.
0.                  1013.9
120.                1041.6
240.                1192.3
360.6
0.
0.                  1048.6
120.                1062.9
240.                1141.2
5.
0.                  1048.6
120.                1062.9
240.                1141.2
360.8
0.
```

```
0.                   1109.2
120.                 1071.4
240.                 1111.4
5.
0.                   1109.2
120.                 1071.4
240.                 1111.4
361.
0.
0.                   1079.
120.                 1084.2
240.                 1090.2
5.
0.                   1079.
120.                 1084.2
240.                 1090.2
361.2
0.
0.                   1078.8
120.                 1118.3
240.                 1098.7
5.
0.                   1078.6
120.                 1118.3
240.                 1098.7
361.4
0.
0.                   1048.6
120.                 1165.2
240.                 1111.4
5.
0.                   1048.6
120.                 1165.2
240.                 1111.4
361.6
0.
0.                   983.5
120.                 1062.9
240.                 962.8
5.
0.                   983.5
120.                 1062.9
240.                 962.8
361.8
0.
0.                   1057.2
120.                 1097.
240.                 1031.5
5.
0.                   1057.2
120.                 1097.
240.                 1031.5
362.
0.
0.                   1078.8
120.                 1032.9
240.                 1179.4
5.
0.                   1078.8
120.                 1032.9
240.                 1179.4
```

```
     0     1
TEND
364.
    10     2     3
362.2
0.
0.                 1087.6
120.               1075.7
240.               1226.2
5.
0.                 1087.6
120.               1075.7
240.               1226.2
362.4
0.
0.                 1074.5
120.               1071.4
240.               1204.8
5.
0.                 1074.5
120.               1071.4
240.               1204.8
362.6
0.
0.                 1109.2
120.               1105.5
240.               1226.2
5.
0.                 1109.2
120.               1105.5
240.               1226.2
362.8
0.
0.                 1074.5
120.               1097.
240.               1200.7
5.
0.                 1074.5
120.               1097.
240.               1200.7
363.
0.
0.                 1109.2
120.               1105.5
240.               1204.8
5.
0.                 1109.2
120.               1105.5
240.               1204.8
363.2
0.
0.                 1078.8
120.               1050.
240.               1192.2
5.
0.                 1078.8
120.               1050.
240.               1192.2
363.4
0.
0.                 1117.9
```

```
120.                 589.5
240.                 1234.7
5.
0.                   1117.9
120.                 589.5
240.                 1234.7
363.6
0.
0.                   1117.9
120.                 1126.8
240.                 1213.3
5.
0.                   1117.9
120.                 1126.8
240.                 1213.3
363.8
0.
0.                   1109.2
120.                 1126.8
240.                 1204.8
5.
0.                   1109.2
120.                 1126.8
240.                 1204.8
364.
0.
0.                   1100.6
120.                 1118.3
240.                 1204.8
5.
0.                   1100.6
120.                 1118.3
240.                 1204.8
```

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:

$$Yn + 1 = P = Yn + \frac{\Delta t}{24}[55\,\dot{Y}n - 59\,\dot{Y}(n-1) + 37\,\dot{Y}(n-2)$$

$$-9\,\dot{Y}(n-3)] + \frac{251}{720}\Delta t^5\,\dot{Y}^{IV}$$

$$Yn + 1 = C = Yn + \frac{\Delta t}{24}[9\,\dot{Y}(n+1) + 19\,\dot{Y}(n) - 5\,\dot{Y}(n-1)$$

$$+\dot{Y}(n-2)] - \frac{19}{720}\Delta t^5\,\dot{Y}^{IV}$$

where $\dot{Y}^{IV}$ is the fifth derivative of Y.

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

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

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

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

$$K_4 = t\,f(x + \Delta t, y + K_3)$$

$$Y = \frac{1}{6}(K_1 + 2K_2 + 2K_3 \ K_4)$$

$$Y_2 = Y_1 + \Delta Y$$

APPENDIX D

PROGRAM LISTING

```
//NQJFLMFD JOB (NQ0011879K,C,A00027,H00001),122,MSGLEVEL=(1,1)
// EXEC PGM=IEFBR14,REGION=6K
//SCRATCH DD UNIT=DISK,VOL=SER=M2SCR6,DSN=NQJFLMFD,DISP=(OLD,DELETE)
// EXEC FORTRANG
//SOURCE.SYSIN DD *
C         JOHN F LAUDADIO 3/18/70
      REAL*8DICT(300)
C    ZILCHI IS COMMON FOR THE MANIFOLD SUBROUTINES
      COMMON/ZILCH1/YX(15),YPRIME(15),ERCRIT(2),P(4),RAD(4),AL(4),
     1ALT(50),VEL(50),TF(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,
     4THRCON,PLTCON ,FAZANG,ALPHA,CP,AA,TM,  PRNTFQ,TEND,PHI,RT,INDPVR,
     5JMAX,NONLIN,IERCRT,IXMAX,IYMAX,IZMAX,KT,NOQ,NOPTS,NCQ,NALPHA,
     6NCURVS,PLOTFQ,IPLOT(15),APLOT(15)
C         THE  'B' ARRAY CONTAINS ALL INPUT QUANTITIES FROM READIN
      DIMENSION B(300)
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','NOQ','RO','TM','INDPVR',
     3'BLANKS','NCURVS','BLANKS','PLOTFQ',7*'BLANKS',4*'PRESSF','OPTSEL'
     4,'CPCNTL','TCON','TTUBE','THICK','CPTUBE','ROTUBE',5*'BLANKS',
     5 'DELT','DELMAX',2*'ERCRIT','JMAX','NONLIN','IERCRT','ICP','KT',
     6 'SAVE',10*'ALPHAA',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)
55551 CONTINUE
      NCQ=B(71)
      PRNTFQ=B(72)
      BDUMP=B(73)
C    T IS THE INITIALIZATION TIME FOR THE INTEGRATION
C         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 SET =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 PRESUR UNLESS DENSITY IS INPUT ,IE DENSITY IS
```

```
C         NONSTANDARD AND TAKES PRECIDENCE IF BOTH TEMP AND DENSITY ARE
C         INPUT BY MISTAKE. PRESS MUST ALWAYS BE INPUT
C         YX(2) IS THE INITIAL TEMPERATURE
      YX(1)=B(86)
      YX(2)=B(87)
      IF( YX(2) .LT. 0.) WRITE(6,1000)
1000  FORMAT('0 THE ISOTHERMAL SOLUTION TO THE EQNS HAS BEEN CHOSEN')
      INDPVR=B(88)
C         INDPVR IS A TRIGGER INDICATING THE TYPE OF INDEPENDENT VARIABLE
C         BEING USED FOR THE CQ TABLE,IE, -1,RENO, O(DEFAULT ) SQRT(RE),
C         1 PRESSURF RATIO .GT. 1.0
      IF(INDPVR .NE. 0) WRITE(6,100)INDPVR
100   FORMAT('0 A NONSTANDARD INDEPENDENT TABLE INPUT HAS BEEN USED FOR
     1CQ, INDPVR=',I2)
C         SET UP VARIABLES FOR PLOTTING
      NPPTS=B(89)
      NCURVS=B(90)
C         DEFAULT NPPTS
      IF(NPPTS .EQ. 0 .AND. NCURVS .GT. C) NPPTS=100
      MCDE=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(5&6) ARE INPUT AS VARIABLE NAMES (OPTSEL & CPCNTL) IN
C READIN BUT IS HANDLED EVERYWHERE ELSE AS PRESSF(5&6)
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 OPTICN HAS BEEN CHOSEN,IE, P(K)=PO*CP')
      IF(PRESSF(5)  .EQ. -2.) WRITE(6,113)
113   FORMAT('0 THE SPFCIAL USE OPTION HAS BEEN CHOSEN, IE, P(K)=CP ')
      IF(PRESSF(5) .EQ. -3.) WRITE(6,111)
111   FORMAT('0 THE SPECIAL USE OPTICN HAS BEEN CHOSEN, IE, P(K)=F(TF)')
      IF(PRESSF(6) .EQ. -1. .OR. PRESSF(6) .EC. 2.)WRITE(6,112)
112   FCRMAT ('OTHE SPECIAL USE OPTICN OF CP AS A FUNCTION OF TIME HAS
     1BEEN CHOSEN,IE, CP=F(T,ALPHA,PHI)')
C         FOR THE SPECIAL USE OPTICNS THE ATMOSPHERIC VARIABLES ARE INPUT
C         AS CONSTANTS IN THE PRESSF ARRAY AS FOLLOWS
C         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)
      ROTUBE=B(110)
C         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
C         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
C        SAVE=1 SAVES PM,PM,T FROM PRESENT CASE TO BE USED IN NEXT CASE
C        SAVE MUST BE RESET EACH TIME IT IS TO BE USED
      SAVE=B(125)
      IF(NOQ .LE.  1) WRITE (6,105) NOQ
      IF(NOQ .GE.  2) WRITE (6,106) NCQ
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(1) .NE. 0. .OR. VEL(1) .NE. 0. .OR. TF(1) .NE. 0.).AND.
     1NOPTS .EQ. 0.) WRITE(6,225)
225   FORMAT('0 THE TRAJECTORY HAS BEEN INPUT BUT NOPTS IS STILL REQD')
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)
      ALPHAA(I)=B(I+125)
5     TALPH(I)=B(I+135)
      IF(ALPHAA(1) .NE. 0. .OR. TALPH(1) .NE. 0. .AND. ALPHA .EQ. 0.)
     1WRITE(6,226)
226   FORMAT('0 ALPHAA OR TALPH HAS BEEN INPUT BUT ALPHA IS STILL REQD')
      IF(CQ(1) .NE. 0. .OR. FRENO(1) .NE. 0. .AND. NCQ .EC. 0.)WRITE(6,
     1107)
107   FORMAT('0 ORIFICE COEFF HAVE BEEN LOADED BUT NCQ (COUNT OF NCQ''S)
     1HAS NOT BEEN LOADED')
      IF(BDUMP .EQ. 1.) WRITE(6,10001)(B(I),I=1,300)
10001 FORMAT(1H0,10G13.6)
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(1H0,'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(1H0,'INITIAL CONDITIONS ','VR=',G13.6,'OMEGA=',G13.6,'PM=',
     1 G13.6,'TM=',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(1H0,8G10.5)
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   CCNTINUE
921   CCNTINUE
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   CCNTINUE
      VR=VR/1728.
      DO 922 II=1,4
      RAD(II)=RAD(II)/12.
922   AL(II)=AL(II)/12.
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    CCNTINUE
C        INITIALIZE MANIFULD SUBROUTINE HERE
C        ENTER CALCULATION PHASE OF MANIFCLD PROGRAM
      CALL PRESUR
      IF(SAVE .NE. 1.) GC TO 55552
      B(81)=PM
      B(87)=TM
      B(74)=T
      SAVE=0.
55552 CCNTINUE
      ICASES=ICASES-1
      B(78)=ICASES
      IF(ICASES .GT. 0) GO TO 200
      IF(ICASES .LE. 0 .AND. ICP.GF.1) GO TO 55551
      STCP
      ENC
      SUBROUTINE PRESUR
      COMMON/ZILCH1/YX(15),YPRIME(15),ERCRIT(2),P(4),RAD(4),AL(4),
     1ALT(50),VEL(50),TF(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,
     4THRCON,PLTCON ,FAZANG,ALPHA,CP,AA,TM,  PRNTFQ,TEND,PHI,RT,INDPVR,
     5JMAX,NONLIN,IERCRT,IXMAX,IYMAX,IZMAX,KT,NOQ,NOPTS,NCQ,NALPHA,
     6NCURVS,PLOTFQ,IPLOT(15),APLOT(15)
      INTEGER ONE/1/,TWO/2/,I/58/
C     DP/DT IN THIS PROG IS BASED ON ENERGY AND HEAT TRANS EQNS
      DIMENSION ANS(4),DM(4),TA(4),RHO(4),C(15),CQW(4),CPW(4),REW(4)
      DIMENSION PHII(4),PLOT(22),VVOUT(4),QAIR(4)
      DIMENSION TUBE(4),TCONV(4),AVOL(4),ALAT(4),TVOL(4)
      DATA ISKIP/0/
C        INITIALIZE CONSTANTS FOR CALCULATION
C        DEFAULT FOR ORIFICE EQNS IF TUBE DIMENSIONS ARE ZERO
      IF( AL(1) .EQ. 0. .AND. NOQ .LT. 2) NOQ=2
C        INITIALIZE V2 TO DETERMINE CPTICN SELECTION
      V2=0.
C        INITIALIZE PLOT ROUTINE HERE
      IF(NCURVS .GT. 0 .AND. ISKIP .EQ. 0) CALL RJPLOT(APLOT,NCURVS)
C        IF PLTCON IS USED FOR THIS CASE SKIP PLOT REINITIALIZING
      IF(PLTCON .EQ. 1.) ISKIP=1
      IF(PLTCON .EQ. 0.) ISKIP=0
      IIPPSS=0
      PI=3.14159
      GAMMA=1.4
      VV=1.
      AJ=778.26
C        OMEG=ROLL RATE IN CPS
C        SAVE OMEGA IN CPS THEN CONVERT TO RAD/SEC
      OMEG=OMEGA
      OMEGA=OMEGA*6.28318
      GC=32.174048
C        R IN (FT-LBF)/(LBM- DEG R)
      R=53.36
      CCP=.240
      CV=.1710
C       INITIALIZE TOTMAS WHICH IS THE TOTAL MASS CHANGE IN THE MANIFOLD
      TOTMAS=0.
      RR=((.75*VR)/PI)**(.3333)
      TM=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 0 USE DENSITY INPUT OPTION
      IF(YX(1) .EQ. 0.) GO TO 70
      TM=PM/(R*RO)
      YX(2)=TM
      GO TO 75
   70 RO=PM/(R*TM)
      YX(1)=RO
   75 CONTINUE
C        ZERO OUT PORT VARIABLES
      DO 11111 K=1,4
      P(K)=0.
      TA(K)=0.
      CPW(K)=0.
      DM(K)=0.
      REW(K)=0.
      PHII(K)=0.
      VVOUT(K)=0.
      CQW(K)=0.
C        INITIALIZE INITIAL HEAT TRANSFER RATE AND CALCULATE CONSTANTS
      QAIR(K)=0.
      TUBE(K)=TTUBE
      TCONV(K)=TCON
      AVOL(K)=PI*RAD(K)**2*AL(K)
      ALAT(K)=2.*PI*RAD(K)*AL(K)
      TVOL(K)=PI*(RAD(K)+THICK)**2*AL(K)-AVOL(K)
      WRITE(6,10109)AVOL(K),ALAT(K),TVOL(K)
11111 RHO(K)=0.
C        SET V2=PRESSF(5) IF PRESSF(5) LT 0
      IF(PRESSF(5) .LT. 0.) V2=PRESSF(5)
C        INPUT ATMOSPHERIC VARIABLES HERE IF V2 IS NEGATIVE
      IF(V2 .EQ. 0.) GO TO 81
      RHOO=PRESSF(1)
      PO=PRESSF(2)
      TO=PRESSF(3)
      AB=PRESSF(4)
      AA=0.
      HO=0.
      V1=C.
      QO=0.
      CP=0.
81    CCNTINUE
C        FAZANG=ANGLE BETWEEN KT EVENLY SPACED PORTS
      FAZANG=(2.*PI)/KT
C     DISCHARGE COEFFICIENT FOR AN ORIFICE -CONSTANT
      CQD=.611
C        INITIALIZE INTEGRATION SUBRCUTINE HERE
C        LIMIT INTEGRATION STEP SIZE TO OUTPUT FREQUENCY
      IF(DELMAX .GT. PRNTFQ) DELMAX=PRNTFQ
      CALL START1(YX,YPRIME,T,DELT,DELMAX,ERCRIT,JMAX,NONLIN,IERCRT)
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 ROD AND TMD ARE CALCULATED IN PCALC
      IS=0
      III=1
      IORDER=3
C        PLOTF IS INITIALIZED HERE
      PLCTF=0.
35    CCNTINUE
C        TPRNT IS THE TIME AT THE LAST OUTPUT STEP AND IS SAVED TO
C        COMPARE WITH PRNTFQ ( PRINT FRECUENCY)
      TPRNT =T
21    CCNTINUE
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    CCNTINUE
      NOPRNT=3
      IF(NOPTS .EQ. 0 ) GO TO 80
      CALL SINTRP(T,ALT,TF,NOPTS,IS,III,ICRDER,HO,NCPRNT)
      NOPRNT=3
      IF(PRESSF(5) .EQ. -3.) GC TO 80
C        SKIP AT62 AND VEL INTERPOLATION WHEN 2ND SPECIAL USE OPTION
C        IS USED
      CALL SINTRP(T,VEL,TF,NOPTS,IS,III,IORDER,V1,NOPRNT)
      CALL AT62(HO ,ANS)
      RHOO=ANS(1)*GC
      PO=ANS(2)
      TO=ANS(3)*1.80007
      AB=ANS(4)
      AA=ABS(V1)/AB
      QO=(RHOO*V1*V1*.5)/GC
80    CCNTINUE
30    CCNTINUE
C        BEGIN CALCULATION OF DERIVATIVES ROD(RO DOT) & TMD(TM DOT)
      THETA=C.
      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 ,PO=F(TF) WITH
      ICONS=2
      COUNT=0.
      IF(V2 .GE.-2.) CALL CSUBP
      P(K)=PO+CP*QO
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 TC 0. THE DENSITY ECN WILL BE DIVICING BY 0. ALSO
      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 TE
      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 ASSUMMED 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 DIRECTICN CF FLCW
      SS=SQRT(GAMMA*GC*R*TE)
C        CALCULATE VISCOSCITY BASED ON TE
      AMUU=  2.22E-8*(TE**.5/(1.+(180./TE)))
C  IF NOC= 2 OR  3 USE ORIFICE FLOW EQNS  2 NO HEAT ,
      IF(NOQ .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 LBM/FT**3   DM(K) IN LBM/SEC
```

```
      ROAVGS=ROAVG/GC
C        ORIFICE FLOW EQUATIONS
40    CCNTINUE
      DM(K)=CQD*PI*RAD(K)**2*ROAVG*SCRT(2.*(ABS(P(K)-PM))/ROAVGS)
      IF(P(K) .LT. PM) DM(K)=-DM(K)
      VTRIL=ABS(VV)
      VV=DM(K)/(ROAVG*PI*RAD(K)**2)
      RE=(ROAVG*ABS(VV)*2.*RAD(K))/(AMUU*GC)
      IF(VV .GT. SS .AND. NCQ .EQ. 0) GO TO 600
      IF(NCQ .EQ. 0) GO TO 1060
      IF(VV .NE. 0.) GO TO 50
      VV=1.
      DM(K)=0.
      GO TO 1060
50    CONTINUE
C        ITERATE TO FIND DM(K) IF CQD IS VARIABLE
      VTEST=ABS(VV/VTRIL)
      IF(ABS(VTEST-1.) .LE. .01 .AND. VV.GT. SS) GO TO 600
      IF(ABS(VTEST-1.) .LE. .01) GO TC 1060
C        THE DISCHARGE COEFF TABLE IS CQD VS (RE)**.5
C        CHECK TO FIND WHICH INDEP VAR IS USED FOR CQ
      IF(INDPVR)76,77,78
76    RS=RE
      GC TO 79
78    RS=P(K)/PM
      IF(PM .GT.P(K)) RS=PM/P(K)
      GO TO 79
77    RS=SQRT(RE)
79    CCNTINUE
      NOPRNT=3
      CALL SINTRP(RS,CQ,FRENO,NCQ,IS,III,IORDER,CQD,NOPRNT)
      DMSAVE=DM(K)
      GO TO 40
601   CCNTINUE
      ROAVG=(RO+RHO(K))*.5
      VTRIL=ABS(VV)
      RE=(ROAVG*VTRIL*2.*RAD(K))/(AMUU*GC)
C        FRICTION COEFFICIENT FOR $MCCTH PIPES
C        ALLOW CHANGE OF RE RANGE UP TO 10 STEPS
      IF (COUNT .LE. 10) GO TO 704
C        FORCE CALC TO REMAIN IN ONE RE RANGE
      IF(ICONS -2)702,700,701
704   IF(RE .LT. 100000. .AND. RE .GT. 1185.) GO TO 700
      IF(RE .LE. 1185.) GO TO 701
C        NIKURADSE EQN
702   FF=(.0082+.05525/RE**(.237))
      ICCNS=1
      GO TO 705
C        EQN FOR BLASIUS RANGE
700   FF=.0791/RE**.25
      ICCNS=2
      GO TO 705
701   FF=16./RE
      ICCNS=3
705   CCNTINUE
      LOVRD=AL(K)/RAD(K)
C        CALCULATE (1/RO-1/RHO) FOR CCMPRESSIBLE FLCW TUBE FLOW EQN
      ROCOMP=ABS((1./RHO(K))-(1./RO))
C        COMPRESSIBLE FRICTION FLOW EQN INVERTED TO SOLVE FOR A
C        VELOCITY DENSITY-COUPLE USED TO CALC MASS FLOW RATE
```

```
      VV=SQRT((ABS(P(K)-PM)  /(FF*RCAVG*LOVRD+ROAVG**2*ROCOMP))*GC)
      COUNT=COUNT+1.
C        CQD REPRESENTS THE FRICTION FACTOR FF WHEN TURE FLOW IS USED
      CQC=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 APPROXIMATEC BY .611 IT SHOULD BE
C        INVESTIGATED IF ANY AMOUNT OF CHOKED ORIFICE FLOW IS ANTICIPATEC
      IF(NOQ .EQ.  2 .OR. NOQ .EQ.  3) VV=AB*CQD
630   IF(P(K)-PM)602,603,6C3
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 APPROXIMATICN 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 TEMPAVOL=VOL CF AIR IN EACH TUBE,TVOL=
C        VOL OF TUBE MATERIAL
      IF(NOQ .EQ. 0) GO TO 305
      IF(TCON .LE. 0. .OR. NOQ .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   HA=ABS(VV)*ROAVG*CCP*CQD*.5
300   CCNTINUE
      DTAIR=(QAIR(K)*DELT)/(CCP*ROAVG*AVOL(K))
      DTUBE=-(QAIR(K)*DELT)/(CPTUBE*RCTUBE*TVCL(K))
      TMEAN=(TE+TCONV(K))*.5
      TCCNV(K)=TMEAN+DTAIR
      TUBE(K)=TUBE(K)+DTUBE
C        QAIR IS NEGATIVE CUT OF THE AIR
      QAIR(K)=HA*ALAT(K)*(TUBE(K)-TMEAN)
C        CONVECTION HEAT TRANSFER FROM-TO AIR- TUBE
      IF(P(K) .GT. PM) QTOT=QTOT+QAIR(K)
305   CCNTINUE
500   CONTINUE
C        INTEGRATE EQNS HERE ,SEND VALUES FOR CALC DERIVATIVES
C        INTO FUNCTION SUBROUTINE
C     ROD=DMTOT/VR
C     TMD    =(SUM-CV*TM*DMTOT+QTOT )/(CV*RO   *VR)
      C(1)=DMTOT
      C(2)=VR
      C(3)=SUM
      C(4)=CV
      C(5)=QW
      C(5)=QTOT
      CALL CONSTS(C)
20    CCNTINUE
```

```
      J=ONE
      CALL INTEGT(J,RO,ROD,T,ISAVE,DELT)
      J=TWO
      CALL INTEGT(J,TM,TMD,T,ISAVE,DELT)
      IF(ISAVE .EQ. 1) GO TO 20
22222 CCNTINUE
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.
     1 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 FCRMAT('1  ALTITUDE',2X,'VELOCITY',7X,'TIME',3X,'MANIFOLD PRESS',
     14X,'PRESSURE (PSF) AT EACH PORT',24X,'ATMOS PRESS',4X,'QTOT')
10101 FORMAT('0',2X,'QO',8X,'MACH NO',8X,'ALPHA',2X,'MANIFOLC TEMP',5X,'
     1PRESSURE COEFFICIENT AT EACH PCRT',18X,'ROLL RATE')
10106 FORMAT('0  MASS FLOW RATE IN EACH TUBE',23X,'REYNOLDS NO IN EACH T
     1UBE',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 MANIFCLD')
10103 FORMAT('0  VELOCITY IN EACH TUBE',29X,'DISCHARGE CCEFFICIENT AT EA
     1CH PORT')
10107 FORMAT('0  VELOCITY IN EACH TUBE',29X,'FRICTICN FACTOR IN EACH TUB
     1E',23X,'QDOT 1',8X,'GDOT 2')
10108 FORMAT('0  TCONV, MEAN GAS TEMPERATURE IN EACH TUBE',10X,'MEAN TUB
     1E TEMPERATURE OF EACH TUBE',17X,'QCCT 3',8X,'CDOT 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. NOC .EQ.  1) WRITE(6,10108)
      IF(TCON .NE. 0. .AND. NOQ .EQ.  1) I=I+1
505   CCNTINUE
      WRITE(6,10110)HO,V1,T,PM,P(1),P(2),P(3),P(4),PO,QTCT
      WRITE(6,10104)QO,AA,ALPHA,TM    ,(CPW(IZ),IZ=1,4 ),OMEG
      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),(CGW(IZ),IZ=1,4)
      IF(TCON .NE. 0. .AND. NOQ .EQ.  1) WRITE(6,10102)QAIR(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 FCRMAT('+',104X,2G13.6)
10104 FORMAT(1X,10G13.6)
10110 FORMAT(1HO,10G13.6)
      IF(NCURVS .EQ. 0) GO TO 90
C         VARIABLE PLOT POINT CAPABILITY IS CHECKED HERE
10201 IF(PLOTFQ .EQ. PRNTFC) GO TO 10202
      PLCTF=PLOTF+1.
      IF(PLOTF .LT. PLOTFQ) GO TO 10203
      PLOTF=C.
10202 CCNTINUE
      PLCT(1)=T
      PLOT(2)=PM
      PLCT(3)=P(1)
      PLOT(4)=P(2)
      PLOT(5)=P(3)
      PLOT(6)=P(4)
      PLOT(7)=PO
      PLCT(8)=HO
      PLOT(9)=V1
      PLOT(10)=QTOT
      PLOT(11)=QO
      PLOT(12)=AA
      PLOT(13)=ALPHA
      PLOT(14)=TM
      PLCT(15)=REW(1)
      PLCT(16)=REW(2)
      PLCT(17)=REW(3)
      PLCT(18)=REW(4)
      PLCT(19)=CQW(1)
      PLCT(20)=CQW(2)
      PLCT(21)=CQW(3)
      PLCT(22)=CQW(4)
      DO 10200 IPL= 1,NCURVS
10200 APLOT(IPL)=PLOT(IPLOT(IPL))
      CALL RJPLOT(APLOT,0)
10203 IF((T-TPRNT) .LT. PRNTFQ) GO TC 21
90    CCNTINUE
      IF(T .LT. TEND) GO TO 35
      I=58
      IF(PLTCON .EQ. 1.) GO TO 10209
      CALL RJPLOT(APLOT,-1)
10209 RETURN
      ENC
/*
// EXEC 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 NQJFLMFD(R)
/*
.. DATA DMS,N,GCA19
```

E

```
      SUBROUTINE CSUBP
      COMMON/ZILCH1/YX(15),YPRIME(15),ERCRIT(2),P(4),RAD(4),AL(4),
     1ALT(50),VEL(50),TF(50),CQ(10),FRENQ(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,PM,VR,THETA,OMEGA,
     4THRCON,PLTCON ,FAZANG,ALPHA,CP,AA,TM,  PRNTFQ,TEND,PHI,RT,INDPVR,
     5JMAX,NONLIN,IERCRT,IXMAX,IYMAX,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 INTERPOLATICN USEC 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 TO512
      IY=IY+1
      GO TO 511
512   CCNTINUE
C     PHI= VEHICLE ROLL ANGLE
      PHI=OMEGA*T+THETA
      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. PRESSF(6) .LT. 1.) PHI=2.*PI-PHI
C        PHI IN DEG = PHI(RAD)*57.296(DEG/RAD)
      PHI=PHI*57.296
540   IF(PHI .GE. FI(IX,IY,IZ) .AND. PHI .LE. FI(IX,IY,IZ+1)) GO TC 550
      IZ=IZ+1
      GO TO 540
550   CCNTINUE
C     TEMPORARY PROCEEDURE 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   CCNTINUE
      DIV1=FI(IX,IY+1,IZ+1)-FI(IX,IY+1,IZ)
      DIV2=FI(IX,IY  ,IZ+1)-FI(IX,IY  ,IZ)
      SLCPE1=(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,IZ))+CPP(IX,IY+1,IZ)
      CP2= SLOPF2*(PHI-FI(IX,IY,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
      CPI=CP
      GO TO 405
400   IF(MINTRP .EQ. 2) GO TO 410
C        CP VALUE BETWEEN MACH NUMBERS
      CP=((CP-CPI)*(AT-CMACH(IX-1)))/(CMACH(IX)-CMACH(IX-1)) +CPI
410   CCNTINUE
C     FAZANG IN RAD =2.*PI/KT) WHERE KT = NO CF PCRTS
      THETA=THETA+FAZANG
      RETURN
      END
      SUBROUTINE READIN(DICT,B,NDICT)
      REAL*8 DICT(NDICT),A(100),ARRAY
      INTEGER OPTION
      DIMENSION B(NDICT),             INPUT(100),FMT(20),FMTA(20),
     1FMTW(20),FMTHOL(20)
C         INITIALIZF READ AND WRITE FORMATS
      DATA FMT/'(10G','8.5)',18*'    '/
      DATA FMTA/'(10G','8.5)',18*'    '/
      DATA FMTW/'(10G','13.6',')   ',17*'    '/
C         NDICT IS THE NUMBER OF DICTICNARY ENTRIES,IE HIGHEST INDEX USED
      DO 1000 I=1,100
      A(I)=0.
      INPUT(I)=0
1000   CONTINUE
1     FORMAT(16I5)
2     FORMAT(10A8)
3     FORMAT(20A4)
7     FORMAT(1H0,10A8)
9     FORMAT(A8,I7,13I5,(/,16I5))
10    FORMAT('0 THE NAME  ',A8,'IS NOT IN THE DICTIONARY, IT IS MISSPELL
     1ED OR MISPLACED IN THE FIELD.PLEASE CORRECT AND RESUBMIT PROGRAM')
C         ISTOP IS USED TO STOP THIS PROGRAM WHEN  INPUT NAMES ARE
C         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   CCNTINUE
      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.53) READ(5,3) FMTHOL
      IF(NEWFMT .EQ. 2 .OR. NEWFMT .EQ.53) WRITE(6,FMTHOL)
      IF(NEWFMT .EQ.51 .OR. NEWFMT .EQ.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
C         OPTION,6=READ IN AN ARRAY NAME + SPECIFIC ARRAY LOCATIONS,7=
C         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 PREVIOLS OPTICN TO INPUT A NEW DATA FORMAT
C         IE, RE-DEFINE A PREVIOUSLY DFFINED DATA FORMAT, 50,56,540,543
C         NOTE OPTICN 4 MAY NOT PRECEDE  OPTION 5 IE, 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    CCNTINUE
      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
6     CCNTINUE
```

```
C          IF THIS WRITE IS IMPLEMENTED AN UNRECGONIZABLE NAME HAS
C          BEEN FOUND, NAME PROCESSING WILL CONTINUE UNTIL END OF NAMES
      WRITE(6,10) A(I)
      ISTOP=1
      INPUT(I)=1
5     CCNTINUE
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 TC BE READ IN STARTING AT
C  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)) GC TO 55
50    CCNTINUE
C          IF THIS WRITE IS IMPLEMENTED AN UNRECGONIZABLE NAME HAS
C          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   CCNTINUE
      READ(5, 9) ARRAY,(INPUT(I),I=1,NOVAR)
      DO 60 I=1,NDICT
      IF(ARRAY .EQ. DICT(I)) GO TO 61
60    CCNTINUE
C          IF THIS WRITE IS IMPLEMENTED AN UNRECGONIZABLE NAME HAS
C          BEEN FOUND, NAME PROCESSING WILL CONTINUE UNTIL END OF NAMES
      WRITE(6,10) ARRAY
      ISTOP=1
      I=1
61    CCNTINUE
      ISAVE=I-1
      DO 62 I=1,NOVAR
      INPUT(I)=INPUT(I)+ISAVE
62    CCNTINUE
      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),DENOM( 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 INTERPOLATION TOOK PLACE OUTSIDE THE TABLE ,LOWER
C     (-1),UPPEREND(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=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(IT),X(IT),Y(IMAX),X(IMAX),XI,IMAX
101   FORMAT(1H0,'AN ERRONEOUS VALUE HAS BEEN SUPPLIED TO SINTRP, IE XI
     1 IS OUTSIDE THE TABLE BEING USED,Y1=',D15.8,'X1=',D15.8,/,' YMAX=
     2 ',D15.8,'XMAX=',D15.8,'XI=',D15.8,'IMAX=',I5,/,' THE APPROPRIATE
     3END POINT IS USED FOR THE INTERPOLATED VALUE')
45    CONTINUE
      II=IMAX
      IF(XI .LT. X(IMAX))II=1
      UX=Y(II)
      IORDER=IORDRS
      NOPRNT=-1
      IF( II .EQ. IMAX) NOPRNT=1
      D=UX
      RETURN
21    IF(XI .LT. X(IT)) GO TO 46
C     SAVE THE ORIGINAL INDEX I TO RETURN TO CALLING PROGRAM
      IRETRN=I
```

```
C    THE INDEX I REPRESENTS THAT POINT IN THE TABLE WHERE THE
C    POLYNOMIAL WAS STARTED
      IF(IORDER .NE. IORDRS) I=I-(IORDRS-IORDER)
C    NTH ORDER INTERPOLATOR
50    CCNTINUE
C    RETURN IORDER TO ORIGINAL VALUE
      IORDER=IORDRS
      NOPRNT=C
      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(I)-X(L)
C    SET DIAGONAL =1 SO THAT DIVISICN BY 0 DCES NCT OCCUR
      IF(I .EQ. L) DENOM(II,LL)=1.
25    CCNTINUE
      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    CCNTINUE
      U(I)=TERM
26    CCNTINUE
      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    CCNTINUE
      I=IRETRN
      D=UX
22221 CCNTINUE
      RETURN
      END
```

```
           SUBROUTINE AT62 (ZFT,ANS)
          REAL MOLWT,LOGPR
          DIMENSION H(23),TBASE(22),TGRAD(22),PBASE(22),MOLWT(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
        3,26C.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.,7.,5.,4.,3.3,2.6,1.7,1.1/,PBASE/1.,2.23361E-01,5.40
        6328E-02,8.56663E-03,1.09455E-03,5.82289E-04,1.79718E-C4,1.0241E-05
        7,1.6223E-06,1.6223E-06,2.9681E-07,7.2582E-08,2.4887E-08,4.9955E-09
        8,3.6460E-09,2.7561E-09,1.6632E-C9,6.8694E-1C,1.8592E-10,3.9777E-11
        9,1.C814E-11,3.405E-12/,MOLWT/1C*28.9644,28.88,28.56,28.07,26.92,26
        A.66,26.40,25.85,24.70,22.66,19.94,17.94,16.84,16.17/,RE/6355.63/,C
        B/34.1628/,CRHO/.3236458E-03/,PSFA/2116.22/
          IF (ZFT.LT.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
1         CCNTINUE
2         CCNTINUE
          TKELV=TBASE(J)+TGRAD(J)*(Z-H(J))
          IF (ABS(TGRAD(J)).LT..5) GO TC 3
          PSF=PBASE(J)*PSEA*((TBASE(J)/TKELV)**(C/TGRAD(J)))
          GO TO 4
3         CONTINUE
          PSF=PBASE(J)*PSEA*EXP(-C*(Z-H(J))/TRASE(J))
4         CCNTINUE
          SLGFT3=CRHO*PSF/TKELV
          VSOUND=SQRT(4325.746*TKELV)
          GC TO 8
5         CCNTINUE
          DO 6 K=10,22
          IF (Z.GE.H(K).AND.Z.LE.H(K+1)) GC TC 7
6         CCNTINUE
7         CONTINUE
          TM=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./(A+X))-(1./A)+(1./(A-B))*ALOG
        1((A*(B+X))/(B*(A+X))))
          PSF=PBASE(K)*PSEA*EXP(LOGPR)
          SLGFT3=CRHO*PSF/TM
          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,DELTT,YX(15),XX,DELMX,ERCRIT(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),FDOTS(10,15),PCER   ,DELT,DELMAX,DEL,SAVE,SI,SAVE1, SII,
     2 ISKIP(15),K,N,J,JMAX,ISTEP,IDUBLE,ICHECK,NODBLE,NONLIN,ITERAT
C    W=DEP VAR,T= INDEP VAR, DELT=CHANGE IN INDEP VAR
C    JMAX= NO OF EQNS ,K,N,I, ARE FREE INDICIES
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   DELTT DELTA T ,STEP SIZE INITIAL VALUE
C        YDOT(I)=INITIAL VALUES OF THE DERIVATIVE
C   ERCRIT(1)= ERROR CRITERIA FOR STABILITY,ERCRIT(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
22223 CCNTINUE
      ENTRY START1(YX,YDOT  ,XX,DELTT,DELMX,ERCRIT,JAX,NCNLN,IERCRT)
      JMAX=JAX
      NCNLIN=NONLN
      DELT=DELTT
      DELMAX=DELMX
      T(1)=XX
C   DEFAULT OPTION FOR ERRCR CRITERIA
      SII=.01
      PCER=.01
      IF(IERCRT .EQ. 0) GO TO 5
      IF(IERCRT .EQ. 1) SII=ERCRIT(1)
      IF(IERCRT .EQ. 2)PCER=ERCRIT(2)
      IF(IERCRT .NE. 3) GO TO 5
      SII=ERCRIT(1)
      PCER=ERCRIT(2)
5     CCNTINUE
      DO 20 I=1,JMAX
      W(1,I)=YX(I)
      IF(NONLIN .GT. 0) FDCT(1,I)=YDOT(I)
      ISKIP(I)=1
      ZIP(I)=0.
20    CCNTINUE
      DEL=DELT
      SAVE1=0.
      K=1
      IDUBLE=1
      NCDBLE=0
      ISTEP=1
      ISAVF=SAVE1
      ICHECK=0
      RELTIM=T(1)
      RETURN
      ENTRY CONSTS(C)
C  CST TRANSFERS CONSTS OR VAR TO THE DIFF EQN WHILE INTEG IS IN PROCESS
      DO 30 IC=1,15
30    CST(IC)=C(IC)
      RETURN
      ENTRY INTEGT(JJ,WW,FDT,TT,ISAVE,     DELTT)
      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)
      DELTT=DELT
22221 CCNTINUE
      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,SI,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(1,I)
      YDCT(I)=FDOT(K,I)
      IF(K .LT. 8)  YDOT(I)=FDOT(1,I)
50    CCNTINUE
100   GO TO (1,2,3,4,5,6,7,8,9,10),J
1     CCNTINUE
C    PLACE THE 1ST  EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      YDOT(1)=CST(1)/CST(2)
      FDCT(K,J)=YDOT(J)
      RETURN
2     CCNTINUE
C    PLACE THE 2ND  EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      YDOT(2)=(CST(3)-CST(4)*Y(2)*CST(1)+ CST(5))/(CST(4)*Y(1)*CST(2))
      FDOT(K,J)=YDOT(J)
      RETURN
3     CCNTINUE
C    PLACE THE 3RD  EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      FDOT(K,J)=YDOT(J)
      RETURN
4     CCNTINUE
C    PLACE THE 4TH  EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
      FDOT(K,J)=YDOT(J)
      RETURN
5     CCNTINUE
C    PLACE THE 5TH  EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
      FDOT(K,J)=YDOT(J)
      RETURN
6     CCNTINUE
C    PLACE THE 6TH  EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
      FDOT(K,J)=YDOT(J)
      RETURN
7     CCNTINUE
C    PLACE THE 7TH  EQN ,YDOT=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
      FDOT(K,J)=YDOT(J)
      RETURN
8     CCNTINUE
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    CCNTINUE
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/ 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,SI,SAVE1, SII,
     2 ISKIP(15),K,N,J,JMAX,ISTEP,IDUBLE,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
      X=T(N+1)
      DO 50 I=1,JMAX
      Y(I)=W(N+1,I)
50    YDOT  (I)=FDOT(N+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     CCNTINUE
C    PLACE THE 1ST  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      DFDY=0.
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
2     CONTINUE
C    PLACE THE 2ND  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      DFDY=0.
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
3     CCNTINUE
C    PLACE THE 3RD  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      DFDY=(CST(1)+CST(2)*DCOS(4*Y(4)))**2*Y(3)-(CST(1)+CST(2)*DCOS(4.*
     1 Y(4)))*(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 FOLLOWIN THIS CARD
      DFDY=0.
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
5     CCNTINUE
C    PLACE THE 5TH  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
6     CCNTINUE
C    PLACE THE 6TH  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWIN THIS CARD
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
7     CCNTINUE
C    PLACE THE 7TH  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
8     CCNTINUE
C    PLACE THE 8TH  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
9     CCNTINUE
C    PLACE THE 9TH  EQN ,DFDY=F(X,Y) IMMEDIATELY FOLLOWING THIS CARD
      AK(J)=DABS(DFDY*DELT)
      GO TO 100
10    CCNTINUE
```

```
C   PLACE THE 10TH  EQN ,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
40    SAVE=B(L+1)
36    IF(SI-AK(L+1))41,35,35
41    SI=AK(L+1)
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(QQQ)=DEXP(QQQ)
      ALOG(QQQ)=DLOG(QQQ)
      ABS(QQQ)=DABS(QQQ)
      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)
      DENOM=(T(N)-T(N-1))*(T(N)-T(N-2))*(T(N)-T(N-3))*(T(N)-T(N-4))
      DENOM1=(T(N-1)-T(N))*(T(N-1)-T(N-2))*(T(N-1)-T(N-3))*(T(N-1)-
     1 T(N-4))
      DENOM2=(T(N-2)-T(N))*(T(N-2)-T(N-1))*(T(N-2)-T(N-3))*(T(N-2)-
     1 T(N-4))
      DENOM3=(T(N-3)-T(N))*(T(N-3)-T(N-1))*(T(N-3)-T(N-2))*(T(N-3)-
     1 T(N-4))
      DENOM4=(T(N-4)-T(N))*(T(N-4)-T(N-1))*(T(N-4)-T(N-2))*(T(N-4)-
     1 T(N-3))
      IF(DENOM .NE. 0. .AND.DENOM1 .NE. 0. .AND. DENOM2 .NE. 0. .AND.
     1DENOM3 .NE. 0. .AND. DENCM4 .NE. 0.) GO TO 30
      ZPC=  ((W(N,J)+W(N-5,J))/120. +(W(N-1,J)+W(N-4,J))/24. +(W(N-2,J)
     1 +W(N-3,J))/12.)
      RETURN
30    CCNTINUE
      DO 20 I=1,15
20    Y(I)=W(I,J)
C        ZPC REPRESENTS THE 5TH DERIVATIVE OF THE FUNCTION
C        THE USER MAY DEFINF ZPC ANALYTICALLY OR USE THE STATEMENT
C        FUNCTION DERIV WHICH FINDS ZPC VIA THE 4TH DIVICEC CIFFERENCE
C        OF THE FUNCTION'S DERIVATIVE,FDCT
C     CONTINUE STATEMENT  REPRESENTING THE EQN NUMBER
      GO TO (1,2,3,4,5,6,7,8,9,10),J
1     CCNTINUE
      ZPC=DERIV (FN,FN1,FN2,FN3,FN4,DENOM,DENOM1,DENOM2,DENOM3,DENOM4)
      RETURN
2     CCNTINUE
      ZPC=DERIV (FN,FN1,FN2,FN3,FN4,CENOM,DENOM1,DENOM2,DENOM3,DENOM4)
      RETURN
3     CCNTINUE
      ZPC=DERIV (FN,FN1,FN2,FN3,FN4,DENOM,DENOM1,DENOM2,DENOM3,DENOM4)
      RETURN
4     CCNTINUE
      ZPC=DERIV (FN,FN1,FN2,FN3,FN4,DENCM,DENCM1,DENOM2,CENOM3,DENCM4)
      RETURN
5     CONTINUE
      RETURN
6     CCNTINUE
      RETURN
7     CCNTINUE
      RETURN
8     CCNTINUE

      RETURN
9     CCNTINUE
      RETURN
10    CCNTINUE
      RETURN
      END
      SUBROUTINE RKUTTA
      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,SI,SAVE1, SII,
     2 ISKIP(15),K,N,J,JMAX,ISTEP,IDUBLE,ICHECK,NODBLE,NONLIN,ITERAT
      DIMENSION A(4)
      DO 10 I=1,4
      CALL FUNCT
      A(I) = DELT  * FDOT(K,J)
      GO TO (1,1,3,10),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    CCNTINUE
      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 .GE. 1) GO TO 15
      CALL STPCHK(W,FDOT,T,DELT,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 ICHECK=3 DELT WAS TOO LARGE ,CALC WILL BE REDONE WITH DELT/2
      IF (ICHECK .NE. 3) GO TO 30
      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    CCNTINUE
      K=1
      T(1)=T(N)
      J=JMAX
      RETURN
      END
      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(1,J)
      E(J)=FDOT(1,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(1,J)=W(N,J)
      FDOT(1,J)=FDOT(N,J)
      IF(J .LT. JMAX) RETURN
      ICHECK=2
      T(1)=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 2I=1,JMAX
      W(1,I)=D(I)
      FDOT(1,I)=E(I)
2     CCNTINUE
      T(1)=F
      DEL=DELT
      ICHECK=3
      SAVE1=1.
      RETURN
      END
```

F

```
      SUBROUTINE INTGRT
      IMPLICIT REAL*8 (A-H,O-Z)
      EXP(QQQ)=DEXP(QQQ)
      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),
     1 ZIP(15),FDOTS(10,15),PCER   ,DELT,DELMAX,DEL,SAVE,SI,SAVE1, SII,
     2 ISKIP(15),K,N,J,JMAX,ISTEP,IDUBLE,ICHECK,NODBLE,NONLIN,ITERAT
      REAL MODIFR
      DIMENSION PRED(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=9
      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
     1-1,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(6,11000) PRED(J),CORR(J),FDOT(9,J),
     1TTEMP,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(1H0,4D16.8,I5)
      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)
```

```
305    CCNTINUE
       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 EQN
       IF(NONLIN .EQ. 1) CALL SYSIT
135    CCNTINUE
       IF(NONLIN .GE. 2) CALL SYSIT1(&600,&125,&135)
125    CCNTINUE
C     STEP SIZER CHFCKS ERROR TERMS HERE
C            ERROR CRITERIA  DETERMINES WHETHER TO DOUBLE OR HALF
C     SI AND SAVE CHECK  STABILITY AND TRUNCATION ERROR RESP
       IF(SI .GT. SII) GC TC 211
C     PCER IS THE ERROR CRITERIA DEPENDING ON THE PRED AND CORR.
C     PCER IS MADF STRONGER IF SI=0. (DFDY=0) SINCE IT IS THEN THE ONLY
C     ERRCR CRITERIA FOR CHECKING
       IF(SI .EQ. 0. .AND. PCER .GT. .0001) PCER=.0001
       IF(ABS(SAVE) .LE. PCER) GO TO 50
211    CALL STPSIZ
       IDUBLE=1
       NODBLE=0
       GC TO 106
50     IF(NODBLE .EQ. 1) GO TO 105
       IF(ISTEP .LE.  5)GC TO 1C5
       IF(SI .GT..5*SII.AND. SAVE .GT. .5*PCER) GO TO 105
       CALL DOUBLE
106    IF(DEL .EQ. DFLT) 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    CCNTINUE
       ISTEP=ISTEP+1
       J=JMAX
116    CCNTINUE
       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    CCNTINUE
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,CELMAX,DEL,SAVE,SI,SAVE1, SII,
     2 ISKIP(15),K,N,J,JMAX,ISTEP,IDUBLE,ICHECK,NODBLE,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
      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(FDCT(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
      GC TO 2
6     CCNTINUE
      NCNLIN=1
      J=JMAX
      RETURN 2
      END
      SUBROUTINE STPSIZ
      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,SI,SAVE1, SII,
     2 ISKIP(15),K,N,J,JMAX,ISTEP,IDUBLE,ICHECK,NODBLE,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) GU 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)
      FDCT(2,J)=FDOTS(2,J)
      FDCT(3,J)=FDOTS(3,J)
      FDCT(4,J)=FDOTS(4,J)
      FDCT(5,J)=FDOTS(5,J)
      FDCT(6,J)=FDOTS(6,J)
      FDCT(7,J)=FDOTS(7,J)
      FDCT(9,J)=FDOTS(9,J)
 25   CCNTINUE
      GO TO 30
 20   CCNTINUE
      DO 1 J=1,JMAX
 100  FORMAT(1X,8D15.8)
      DO 11 I=5,8
      WS(I-4)=W(I,J)
      FS(I-4)=FDOT(I,J)
      TX(I-4)=T(I)
 11   CCNTINUE
      DC 2 L=1,3
      XI=TX(L)+DEL
      CALL SINTP   (XI,WS,TX,IMAX,IS,L,ICRDER,UX,NOPRNT)
      A(L)=UX
      CALL SINTP   (XI,FS,TX,IMAX,IS,L,IORDER,UX,NOPRNT)
      B(L)=UX
 2    CCNTINUE
      MM=4
      DC 3 M=2,6,2
      MM=MM+1
      W(M,J)=W(MM,J)
      FDOT(M,J)=FDOT(MM,J)
 3    CCNTINUE
      MM=1
      DO 4 M=3,7,2
      W(M,J)=A(MM)
      FDOT(M,J)=B(MM)
      MM=MM+1
 4    CCNTINUE
 1    CONTINUE
 30   CCNTINUE
      IORDER=2
      IMAX=8
      DO 5 K=5,7
      XI=T(K)+DEL
      CALL SINTP (XI,T ,T,IMAX,IS,K,IORDER,UX,NOPRNT)
      TX(K)=UX
5     CONTINUE
      MM=5
      I=4
      DO 6 K=1,7
      GO TO (15,16,15,16,15,16,15),K
15    T(K)=TX(I)
      I=I+1
      GO TO 6
16    T(K)=T(MM)
      MM=MM+1
6     CONTINUE
      J=JMAX
      RETURN
      ENTRY DOUBLE
C     STEP SIZE =DELT*2
      DEL=DELT*2.
      DO 55 J=1,JMAX
      IDUBLE=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)
      FDOTS(2,J)=FDOT(2,J)
      FDOTS(3,J)=FDOT(3,J)
      FDOTS(4,J)=FDOT(4,J)
      FDOTS(5,J)=FDOT(5,J)
      FDOTS(6,J)=FDOT(6,J)
      FDOTS(7,J)=FDOT(7,J)
      FDOTS(9,J)=FDOT(9,J)
      DO 55 I=1,3
      GO TO (65,70,75),I
65    N=7
      NN=6
      GO TO 80
70    N=6
      NN=4
      GO TO 80
75    N=5
      NN=2
80    CONTINUE
      W(N,J)=W(NN,J)
      IF(J .EQ.1)  T(N)= T(NN)
      FDOT(N,J)=FDOT(NN,J)
55    CONTINUE
      ISTEP=1
      J=JMAX
      RETURN
      END
      SUBROUTINE SINTRP (A,B,C ,IMAX,IS,I,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
      XI=A
      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 INTERPOLATION TOOK PLACE OUTSIDE THE TABLE ,LOWER
C     (-1),UPPEREND(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=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(IT),X(IT),Y(IMAX),X(IMAX),XI,IMAX
101   FORMAT(1H0,'AN ERRONEOUS VALUE HAS BEEN SUPPLIED TO SINTRP, IE XI
     1 IS OUTSIDE THE TABLE BEING USED,Y1=',D15.8,'X1=',D15.8,/,' YMAX=
     2 ',D15.8,'XMAX=',D15.8,'XI=',D15.8,'IMAX=',I5,/,' THE APPROPRIATE
     3END POINT IS USED FOR THE INTERPOLATED VALUE')
45    CONTINUE
      II=IMAX
      IF(XI .LT. X(IMAX))II=1
      UX=Y(II)
      IORDER=IORDRS
      NOPRNT=-1
      IF( II .EQ. IMAX) NOPRNT=1
      D=UX
      RETURN
21    IF(XI .LT. X(IT)) GO TO 46
C     SAVE THE ORIGINAL INDEX I TO RETURN TO CALLING PROGRAM
      IRETRN=I
C     THE INDEX I REPRESENTS THAT POINT IN THE TABLE WHERE THE
C     POLYNOMIAL WAS STARTED
```

```
      IF(IORDER .NF. IORDRS) I=I-(IORDRS-IORDER)
C    NTH ORDER INTERPOLATOR
50    CCNTINUE
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,IORDER
      L=LL+ISAVE-1
      DENOM(II,LL)=X(I)-X(L)
C    SET DIAGONAL =1 SO THAT DIVISICN BY 0 DCES NOT OCCUR
      IF(I .EQ. L) DENOM(II,LL)=1.
25    CCNTINUE
      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    CCNTINUE
      U(I)=TERM
26    CCNTINUE
      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    CCNTINUE
      I=IRETRN
      D=UX
      RETURN
      END
.. DATA DMS,N,GCB19
/*
```