BRL

TECHNICAL REPORT BRL-TR-2645

BLAF: A BLAST FIELD RECONSTRUCTION PROGRAM FROM PRESSURE HISTORIES

Aivars Celmins

March 1985

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TECHNICAL REPORT BRL-TR-2645	2. GOVT ACCESSION NO. AD-A155300	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) BLAF: A Blast Field Reconstruction Program from Pressure Histories		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Aivars Celmiņš		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: AMXBR-TBD Aberdeen Proving Ground, MD 21005-5066		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L162120AH25
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: AMXBR-OD-ST Aberdeen Proving Ground, MD 21005-5066		12. REPORT DATE March 1985 13. NUMBER OF PAGES 247
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Blast Field Spherical Blast in Ideal Gas Overpressure History Data Reconstructed Blast Field Components Dynamic Pressure		Model Fitting Accuracy Estimates Computer Programs Numerical Integration Flow Velocity/Flow Density

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The BLAF programs (BLAFS, BLAFOP, and BLAFHI) are designed to reconstruct parts of a spherical blast field from selected pressure history observations. The reconstruction method is based on a model fitting to the observed flow field and a subsequent numerical integration of the flow governing equations. This manual gives a short outline of the theoretical background, a description of the program and specifications for the input data and their formats. The flow field reconstruction is done in three steps, each step being realized by

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) an independent program: BLAFS for shock fitting, BLAFOP for blast field overpressure fitting and BLAFHI for blast field history calculations.

The input for BLAFS consists of observed shock arrival times t_s and overpressure p_s at a number of distances r. The program determines a shock model function by adjusting all three components of each data point, t_s, p_s and r. It also accepts incomplete observations where either t_s or p_s is missing. The output of BLAFS consists of a set of shock model parameters with corresponding error estimates.

The BLAFOP program uses as input the results from BLAFS and overpressure history observations at two or more positions. The output is a set of parameters (with error estimates) of an overpressure field function.

The program BLAFHI uses the results of BLAFS and BLAFOP and computes histories of all components of the blast field (velocity, pressure, density, and dynamic pressure) at distances specified by the user. These computations are done by numerical integration of the flow governing equations.

TABLE OF CONTENTS

	PAGE

LIST OF ILLUSTRATIONS

FIGURE NO. | PAGE

1. INTRODUCTION

For experimental studies of target response to high energy blast, one needs an accurate definition of the blast field which provides the load on the target. Direct measurements of the flow field usually are restricted, for technical reasons, to pressure history observations, and to shock arrival time and incident shock pressure measurements at various stations. Hence, one has to compute other flow variables, e.g., the density and the particle velocity, from the measured pressures. The problem can be formulated as a task to solve numerically the governing equations of the flow field with boundary conditions derived from pressure history and shock observations.

In this formulation, the task is a mathematically ill-posed problem because the boundary conditions overdetermine the solution in some parts of the flow field, and at the same time may not be sufficient to compute the complete flow history for the full duration of a pressure history observation at some other station.

A possible regularization of the problem is described in Reference 1. It consists of deleting one of the flow governing equations, solving the ensuing well-posed problem numerically, and using the deleted equation later for control calculations. The calculation starts by first determining a pressure field function $p_f(r,t)$ within a region of interest. The function is found by a least squares model fitting, and substituted into the governing equations which in turn determine the other flow variables. Problems of this type were considered by Makino[2] who observed that one does not need the continuity equation for the flow calculation if $p_f(r,t)$ is known. Following Makino's theoretical ideas, we have established computer programs that compute the flow in the aforementioned manner using the continuity equation at the end of the calculations to check the accuracy of the results. Reference 1 also contains an analysis of the sensitivity of the results to observational inaccuracies. The calculation of corresponding accuracy estimates of the results is included in the computer programs.

The present manual describes the structure of the programs and specifies the input requirements. The basic theory is described in Section 2, and Sections 3 and 4 provide an outline of the solution method. A more detailed description of the method is given in Reference 1. The computer program for the solution consists of three independent parts, BLAFS, BLAFOP AND BLAFHI, which are described in Sections 5, 6 and 7, respectively. Section 8 contains descriptions of all subroutines that are included in the three programs in alphabetical order. The programs are listed in Appendices A, B and C.

Users at the Ballistic Research Laboratory may contact the author about access for the latest versions of the programs.

---

1. Aivars Celmiņš, "Reconstruction of a Blast Field from Pressure History Observations," ARBRL-TR-02367, September 1981 (AD-A106141).
2. Ray C. Makino, "An Approximation Method in Blast Calculations," BRL-MR-1023, February 1956 (AD-114 875).

2. BASIC ASSUMPTIONS AND THEORY

We seek to determine certain parts of the flow field within a blast bubble in air. The area of interest is a relatively narrow strip in the r,t-plane behind the initial shock trajectory at a distance where the shock strength is only moderate. We shall assume that the following conditions are satisfied within the area of interest:

(A) the flowing medium is an ideal gas with zero viscosity and no heat conduction, and

(B) the event is spherically symmetric and the flow has only a radial velocity component u.

The first assumption is satisfied in most applications because typically the maximum overpressure at the target is only of the order of one megapascal. Within this pressure regime air behaves like an ideal gas. The second condition is nearly satisfied in most experiments, because usually the explosion source and the targets are positioned on the same plane, and the blast bubble is a hemisphere. Deviations from spherical flow symmetry within the bubble may be caused by local surface disturbances, by wind, and by the presence of dust in the flow near the ground surface. The present technique cannot be applied to cases where such disturbances are not negligible.

The governing equations for a flow satisfying the conditions (A) are:[3]

$$\frac{d\rho}{dt} + \rho \operatorname{div} u = 0, \tag{2.1}$$

$$\rho \frac{du}{dt} + \operatorname{grad} p = 0 \tag{2.2}$$

and

$$\rho \frac{de}{dt} - \frac{p}{\rho} \frac{d\rho}{dt} = 0, \tag{2.3}$$

in which

$$\frac{d}{dt} = \frac{\partial}{\partial t} + (u \cdot \operatorname{grad}) \tag{2.4}$$

is the material derivative. The equation of state is

$$e = \frac{1}{\gamma-1} \frac{p}{\rho}, \tag{2.5}$$

where γ is the ratio of specific heats.

Eliminating the specific internal energy e between Equations 2.3 and 2.5 one obtains

$$\frac{1}{p} \frac{dp}{dt} - \gamma \frac{1}{\rho} \frac{d\rho}{dt} = 0. \tag{2.6}$$

---

3. Richard von Mises, "Mathematical Theory of Compressible Fluid Flow," Academic Press, NY, 1958.

Equation 2.6 can be integrated along a particle path line. The result is the well known formula for a particle in an adiabatic flow:

$$\frac{\rho}{\rho_A} = \left(\frac{p}{p_A}\right)^{1/\gamma}, \tag{2.7}$$

where the subscript A indicates reference values at a point A on the particle path.

The momentum Equation 2.2 can be reformulated by substituting in it the expression given in Equation 2.7. The result is

$$\frac{du}{dt} = - \frac{1}{\rho_A} \left(\frac{p_A}{p}\right)^{1/\gamma} \frac{\partial p}{\partial r} . \tag{2.8}$$

If the pressure function p(r,t) is given, e.g., by measurements, then Equation 2.8 can be numerically integrated together with the path line equation

$$\frac{dr}{dt} = u. \tag{2.9}$$

The integration provides the path line starting at a point A and the particle velocity along it. The density along the same path line is given by Equation 2.7. All other flow variables, such as, internal energy, dynamic pressure, and sound speed can be computed from p, u, and ρ.

The continuity Equation 2.1 is not needed for the described calculation of the flow corresponding to an observed pressure field p(r,t). Therefore, one can use the equation to test the calculated results, as suggested by Makino.[2] In fact, if the pressure p(r,t) is measured precisely then this test provides a check of the validity of the assumptions (A) and (B) about the flow field. In praxis, test calculations based on the continuity equation cannot provide exactly the same result as the integration along path lines because the pressure field function p(r,t) on the right-hand side of Equation 2.8 is an approximation containing observational and systematic errors. The effects of the former are estimated in our approach from input information about the data accuracy. Systematic errors may manifest themselves by differences between original and control calculations that are larger than predicted by the estimated propagation of the observational errors.

A control calculation based on the continuity equation can be carried out as follows. First, we use Equation 2.6 and reformulate the continuity Equation 2.1, obtaining

$$\text{div } u + \frac{1}{\gamma p} \frac{dp}{dt} = 0, \tag{2.10}$$

or

$$\frac{\partial}{\partial r} (r^2 u) + (r^2 u) \frac{1}{\gamma p} \frac{\partial p}{\partial r} + \frac{r^2}{\gamma p} \frac{\partial p}{\partial t} = 0. \tag{2.11}$$

Equation 2.11 expresses the dependence of the quantity r^2u on r for t = const. A formal integration of the equation along a line t = const. yields

$$u(r,t) = u_C \left(\frac{r_C}{r}\right)^2 \cdot \left(\frac{p_C}{p(r,t)}\right)^{1/\gamma} + \frac{1}{r^2 \gamma p(r,t)^{1/\gamma}} \int_r^{r_C} \xi^2 \cdot p(\xi,t)^{1/\gamma} \frac{\partial p(\xi,t)}{\partial t} \, d\xi \quad (2.12)$$

The subscript C in Equation 2.12 indicates function values at a point C with the coordinates (r_C,t). Using Equation 2.12 one can calculate the particle velocity u(r,t) by a numerical quadrature along t = const., if an initial value u_C and the pressure field function p(r,t) are known.

In summary, we proceed as follows for the calculation of the flow field. First, we establish a pressure field function p(r,t) by data fitting. Next, we integrate Equations 2.8 and 2.9 along a particle path A_1B_1, as shown in Figure 1. The integration produces the velocity u_B at B_1. The density ρ_B can be computed using Equation 2.7, once the path line is established. (The flow variables u_A and ρ_A on the shock are known from the pressure field function and shock relations.) Finally, the calculated velocity u_B is compared with another calculation using Equation 2.12, applied along the line CB_1. The velocity u_C at the point C is again obtained from shock relations.

The overpressure field function is determined within the indicated domain from pressure history measurements along the lines AA_3, BB_3 and CC_3, and from shock observations. The flow history at $r=r_B$ can be calculated between B and B_2, and test calculations by Equation 2.12 can be carried out between B and B_1.

3. NUMERICAL INTEGRATION AND ACCURACY ESTIMATES

In most applications, one needs the flow history at some fixed distance, say r_B. We obtain the history, i.e., the values of flow variables at a series of points along the line $r = r_B$ in Figure 1, by integrating Equations 2.8 and 2.9 along a number of path lines, each starting at a different point of the shock. The test calculation of the velocity is done by integration of Equation 2.12 along appropriate lines t = const. Figure 1 schematically shows the integration lines and the locations of the computed nodes in the r,t-plane. The values of the flow variables at the shock as well as the pressure field function behind the shock that are needed for these integrations, are obtained by model fitting of shock and pressure observations respectively.

The results of the shock model fitting are two functions of the radial distance r and of a model parameter vector θ describing the shock arrival time $t_s(r;\theta)$ and the shock overpressure $p_s(r;\theta)$ respectively. The shock density ρ_s

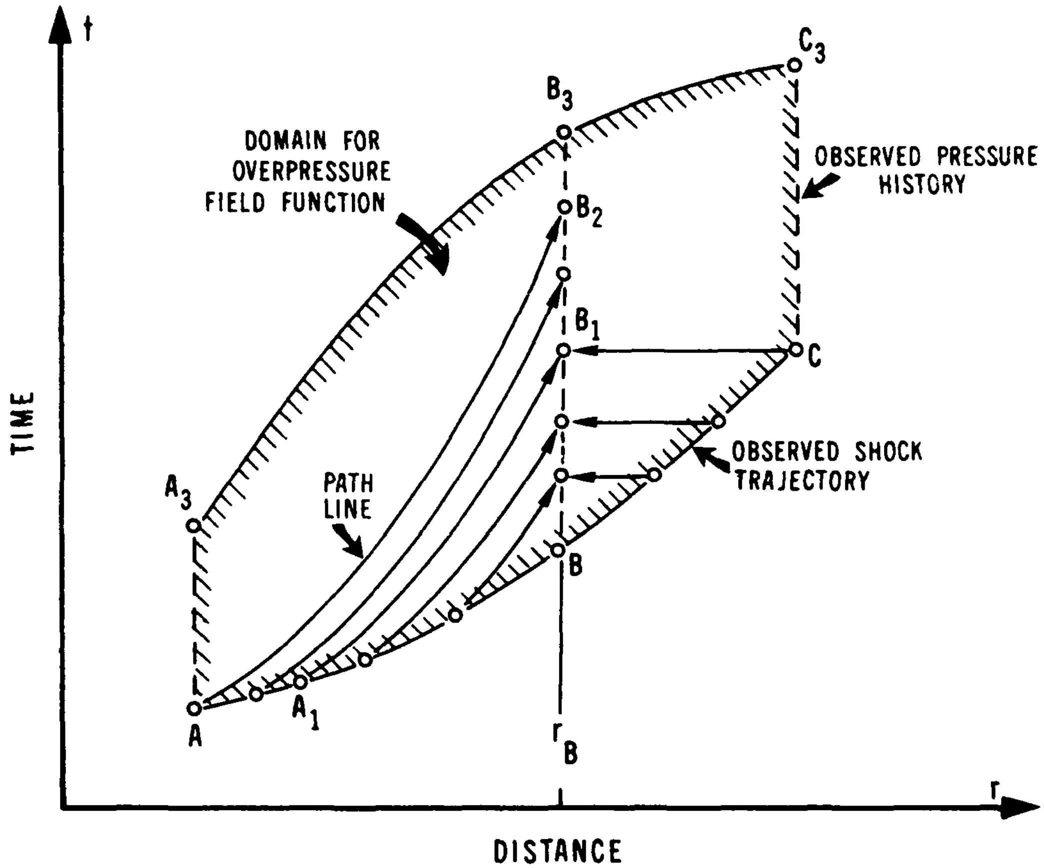


Figure 1. Computation of Flow History at a Given Distance.

The overpressure field function is determined within the indicated domain from pressure history measurements along the lines AA_3, BB_3, and CC_3, and from shock observations. For $r = r_B$, the flow history can be calculated between B and B_2, and test calculations can be carried out between B and B_1.

and particle velocity u_s behind the shock follow from these functions and shock relations. The model fitting of the observed pressure histories produces an overpressure field function $p_f(r,t;\theta)$. (See Section 4.)

The differential equations for the path line, Equations 2.8 and 2.9 are in terms of these functions:

$$\frac{dr}{dt} = u, \qquad \frac{du}{dt} = F(r,t;\theta) \tag{3.1}$$

where

$$F(r,t;\theta) = -\frac{1}{\rho_s(r_A;\theta)}\left(\frac{p_s(r_A;\theta)+p_o}{p_f(r,t;\theta)+p_o}\right)^{1/\gamma} \frac{\partial p_f(r,t;\theta)}{\partial r}, \tag{3.2}$$

and p_o is the ambient pressure. We integrate Equation 3.1 using a fourth order predictor-corrector algorithm.

The control calculation by Equation 2.12 is carried out by substituting p_s and p_f in it and then calculating the integral with a Romberg quadrature routine.

The accuracy of the computed results depends on the accuracies of the integration algorithms as well as on the accuracies of the data that are used to determine the pressure functions p_s and p_f. The pure integration errors can be reduced to desired levels by monitoring the integration step sizes. The errors due to data inaccuracies are estimated using the linearized law of variance propagation as described below.

The least squares data fitting programs[4] provide an estimate of the variance-covariance matrix V_θ of the parameter vector θ in terms of the estimated standard errors of the observations. An estimate of the standard error of a function of θ, e.g., of $p_f(r,t;\theta)$ is given by

$$\epsilon_p = \left[\frac{\partial p_f}{\partial \theta} V_\theta \left(\frac{\partial p_f}{\partial \theta}\right)^T\right]^{1/2}. \tag{3.3}$$

The standard error of $p_s(r;\theta)$ can be calculated by a corresponding formula, and the standard error of ρ can be calculated by using the relation between density and pressure given in Equation 2.7.

The standard error of the particle velocity can be calculated in the same manner provided that one knows the derivative vector $\partial u/\partial\theta$. Unlike $\partial p_f/\partial\theta$,

---

4. Aivars Celmiņš, "A Manual for General Least Squares Model Fitting," ARBRL-TR-02167, June 1979 (AD-B040229L).

that vector cannot be obtained by a formal differentiation because u is not given by a formula but obtained by solving numerically the equation system 3.1. Therefore, we differentiate that system with respect to the parameter and obtain another system of differential equations where the unknown functions are the derivatives $\partial u/\partial\theta$ and $\partial r/\partial\theta$. The new system is

$$\frac{d}{dt}\left(\frac{\partial u}{\partial \theta}\right) = \frac{\partial F}{\partial \theta} + \frac{\partial F}{\partial r}\,\frac{\partial r}{\partial \theta}$$

$$\frac{d}{dt}\left(\frac{\partial r}{\partial \theta}\right) = \frac{\partial u}{\partial \theta}. \tag{3.4}$$

The equations are integrated numerically concurrently with the path line Equations 3.1.

The end point of each path line has an uncertainty in the t-direction which again can be computed by the variance propagation formula using the derivatives

$$\frac{\partial t_B}{\partial \theta} = \frac{2t_s(r_A;\theta)}{\partial \theta} + \frac{1}{u_B} \cdot \left[\frac{\partial r}{\partial \theta}\right]_{r=r_B}. \tag{3.5}$$

For the computation of the standard error of the dynamic pressure $\rho u^2/2$ one needs to know the variances as well as the covariance of ρ and u. The full variance-covariance matrix of the flow field at an end point of a path line is calculated with the formula

$$V_H = \frac{\partial H}{\partial \theta}\, V_\theta \left(\frac{\partial H}{\partial \theta}\right)^T \tag{3.6}$$

where

$$H = (t_B, p_B, u_B, \rho_B)^T \tag{3.7}$$

is a vector that characterizes the flow field. V_H contains the covariance between velocity and density that is needed for the dynamic pressure error estimate.

4. OVERPRESSURE MODEL FITTING

The shock overpressure is modeled by the following three-parameter function

$$p_s(r;a,b,c) = a/r + b/r^2 + c/r^3, \tag{4.1}$$

and the shock arrival time is modeled by the four parameter function

$$t_s(r;a,b,c,d) = d + \int_{r_o}^{r} \frac{dx}{c_o\sqrt{1 + \frac{\gamma + 1}{2\gamma P_o}(a/x + b/x^2 + c/x^3)}}, \tag{4.2}$$

where c_o is the ambient sound speed and r_o is an arbitrary reference distance.

The overpressure field function is modeled by the five parameter model

$$p_f(r,t;\ A_1,A_2,B_1,B_2,C_1;p_s,t_s) = \left[p_s - C_1/r^{n_C}\right] e^{Q} + C_1/r^{n_C} \quad , \qquad (4.3)$$

where

$$Q = \left[t-t_s\right] (A_1+A_2 r)/r^{n_A} + \left[t-t_s\right]^2 (B_1+B_2 r)/r^{n_B} \quad . \qquad (4.4)$$

In these equations, the exponents n_A, n_B, and n_C are determined by an analysis of the trends of the observed pressure histories. Therefore, the total number of free parameters for both model fittings is nine, the four shock parameters, a through d, and the five parameters, A_1 through C_1.

The model fitting is done in two stages using utility programs from Reference 4. In the first stage, one determines the shock functions p_s and t_s. The second stage provides the overpressure field function p_f. The data for the model fittings are measurements of overpressures, times, and distances with corresponding accuracy estimates. In the second stage one also uses as input the results of the first stage, namely, the shock parameters a,b,c,d and their accuracy estimates.

The two adjustment stages are programmed as two independent program packages, BLAFS and BLAFOP. A third package, BLAFHI, uses the results of the first two (essentially, the nine pressure field parameters with accuracy estimates) and carries out the integrations described in Section 3. Instructions for the use of the three program packages are given in Sections 5, 6, and 7, respectively.

5. SHOCK FITTING PROGRAM BLAFS

5.1 Purpose of the Program.

The purpose of the program is to determine from measurements of shock arrival times, distances and overpressures a shock overpressure model function

$$p_s\ (r;\ a,b,c) = a/r + b/r^2 + c/r^3 \qquad (5.1)$$

and a shock arrival time model function

$$t_s(r;\ a,b,c,d) = d + \int_{r_o}^{r} \frac{dx}{c_o\left[1 + \frac{\gamma+1}{2\gamma p_o}(a/x + b/x^2 + c/x^3)\right]^{1/2}} \quad . \qquad (5.2)$$

In these equations r_o is an arbitrary reference distance, c_o is the ambient sound speed, p_o is the ambient pressure, and γ is the ratio of specific heats of the ambient air. These four quantities are part of the input for the program, in addition to the shock measurements. The program calculates least squares values of the four shock model parameters a, b, c and d, and provides estimates of their variances and covariances. A program listing with comments is given in Appendix A, and the subroutines of the program are described in Section 8.

5.2 Input for the Shock Fitting Program

The input consists of two parts: general data describing the ambient air and the charge, and shock observations.

The general data are provided by three mandatory and three optional cards. The end of the general data batch is indicated by a blank card. The first two mandatory cards have the format (8A10) and the third card has the format (2A10, 6E10.3). The contents of the mandatory cards are as follows:

1 TITLE	11 30 character title
1 PLOTLABEL	11 40 character plotting label
1 CHARGE	21 V, E, H, e_H.

The TITLE card contains the identification of the computer run. The identification will appear on all printed and plotted output.

The PLOTLABEL card contains the identification for the Calcomp plotter output. It will not appear on individual plots.

The CHARGE card contains a description of the charge by the following parameters:

- V = volume of the fire ball, m^3,
- E = released energy, J,
- H = height of burst, m,
- e_H = standard error of H, m.

The values of V and E are only needed to scale the event, and they do not affect any other results of the calculations. If scaling is not of interest, then arbitrary or nominal values of V and E may be entered. However, V must be positive. The height H corresponds to the center of the fire ball. It should be small compared to the distance between the center of the explosion and the locations of the pressure gages in order not to violate the assumption of a spherical symmetry of the flow field.

The three optional cards have the same format as the CHARGE card, namely (2A1Ø, 6E1Ø.3), and they may be entered in arbitrary sequence after the first two or three mandatory cards. The cards have the following contents:

1	21
AMBIENT	p_o, t_o, γ, M
1	21
SCALES	s_r, s_t, s_p
1	21
PLOTTING DATA	f_p, f.

The AMBIENT card specifies the ambient air as follows:

p_o = ambient pressure, Pa (1Ø1325.Ø)

T_o = ambient temperature, K (293.Ø)

γ = ratio of specific heats (1.4)

M = molar mass, kg/mol (Ø.Ø2896) .

If this card is missing, or if an input value is not positive, then the missing or faulty value is replaced by the corresponding default value shown in parentheses. The input must be expressed in base SI units, as indicated.

The SCALE card allows one to carry out the calculations in arbitrary scales. The specified scales are:

s_r = distance scale, m

s_p = pressure scale, Pa

s_t = time scale, s.

If the SCALE card is missing or if any of the scales is not positive then the following default scales will be used:

$s_r = V^{1/3}$,

$s_p = p_o$,

$s_t = s_r/c_o$,

where c_o is the ambient sound speed, computed with the formula

$$c_o = (T_o R/M)^{1/2}$$

with the universal gas constant R = 8.3143 J/(K x mol). The scales s_r, s_p, and s_t are also used for the output. Therefore the SCALE card permits one to obtain the output in non-standard scales, if desired. If the output is to be in base SI units then unit scales $s_r = s_p = s_t = 1$ must be specified. The numerical performance of the program is little influenced by the scaling.

The PLOTTING DATA card contains error factors for the plotting of confidence limits:

f_p = error factor for confidence limits in pressure plots,

f = error factor for confidence limits in all other plots.

The plotted confidence limits will correspond to f_p and f standard errors, respectively. If the card is missing then the default values $f_p = f = 2.0$ are used. If a factor is zero then corresponding confidence limits will not be plotted.

The end of the general data is indicated by a blank card. It is followed by cards containing snock data. All shock data cards have the format (2A10, 6E10.3) and their sequence is arbitrary. Each shock point is represented by two cards with identical labels. The two cards contain the following data:

1 10	11 20	21
Label	SHOCKbbbbb	t, e_t, p, e_p

1 10	11 20	21
Label	RANGEbbbbb	x, e_x, h, e_h

where

t = shock arrival time, s,

e_t = standard error of t, s,

p = shock overpressure, Pa,

e_p = standard error of p, Pa,

x = range (ground distance) of observation station, m,

e_x = standard error of x, m,

h = elevation of observation station, m,

e_h = standard error of h, m.

The "Label" is a ten character alphanumeric identification of the observation. Missing t- or p- observations are indicated by a zero or a blank field. ($t = 0$ or $e_t = 0$ indicate a missing time observation; $p = 0$ or $e_p = 0$ indicate a missing pressure observation.)

The maximum number of shock observations that will be read by the program is 50. If the number is less than 50, then the end of the shock data should be indicated by another blank card. The minimum number of shock points for the model fitting is four because the model function contains four free parameters.

After the data have been processed and the shock model parameters determined, the program will try to read the next shock fitting case, starting with the general input. The execution will come to a programmed stop if the input is not a TITLE or PLOTLABEL card, for instance, if it is a blank card.

The computing time for a typical shock fitting problem is less than 20 seconds on the CDC 7600.

5.3 Shock Fitting Process and Output

The shock fitting is done by a least squares process with constraint equations derived from the model functions p_s and t_s, defined by Equations 5.1 and 5.2. Let p_i, r_i and t_i be the observed shock overpressures, distances from the center of explosion and shock arrival times, c_{pi}, c_{ri} and c_{ti} be the corresponding residuals, and let s be the number of observed shock points. Then the constraints are formulated as follows:

$$F_{1i} = (p_i + c_{pi})(r_i + c_{ri})^3 - (r_i + c_{ri})^3\, p_s(r_i + c_{ri};\ a,b,c) = 0,$$
$$F_{2i} = c_o t_s(r_i + c_{ri};\ a,b,c,d) - (t_i + c_{ti})\, c_o = 0\ ,\ i = 1, \ldots, s. \qquad (5.3)$$

The distance r_i is calculated from the range (ground distance) x_i and elevation h_i by

$$r_i = (x_i^2 + (h_i - H)^2)^{1/2} \qquad (5.4)$$

with the estimated standard error

$$e_{ri} = [(x_i e_{xi})^2/r_i^2 + ((h_i - H)/r_i)^2\ (e_{hi}^2 + e_H^2)]^{1/2}. \qquad (5.5)$$

The arbitrary constant r_o in the function t_s, Equation 5.2, is set equal to the smallest observed distance r_i.

The least squares objective function is

$$W = \sum_{i=1}^{s} [(e_{pi}/e_{pi})^2 + (c_{ri}/e_{ri})^2 + (c_{ti}/e_{ti})^2]\ . \qquad (5.6)$$

It is minimized subject to the constraints 5.3. The minimization is done by a version of the least squares utility routine COLSMU (Reference 4) for problems with multi-component constraints. The flexibility of the routine permits one to use also such data sets from which either the overpressure observation p_i

or the time observation t_i is missing. (The constraint for such an incomplete data set is only one of the two Equations 5.3.)

The data fitting is done in four steps:

Step 1. Only pressure is adjusted. This renders the problem linear in the parameters (only the first equation of Equation 5.3 is used) and provides a convenient method to obtain initial approximations of the parameters a, b and c.

Step 2. Only pressures and distances are adjusted. This provides better initial approximations of the three parameters a, b and c for the next step.

Step 3. Simultaneous adjustment of all observations: pressure, distance and time. This provides the final values of all four parameters, a, b, c and d.

Step 4. Only pressures and times are adjusted. This is merely a test for the effect of distance measurement inaccuracies. The result of this step corresponds to the assumption that distances are measured without errors. We notice, however, that the "distances" are measured from an imaginary and ill defined "center of explosion." Therefore, very small distance errors are probably not a realistic assumption and the range standard errors e_x, to be specified by input, probably should be larger than the range survey errors.

The output of the shock fitting program consists of printed summaries of the general data and shock data in self-explaining formats, and of printed and plotted results of the four adjustment steps. The printed output of the adjustment steps also includes standard output generated by the least squares subroutine COLSMU, which may be useful in case of algorithmic difficulties. Normally, the only relevant output is the self-explaining summary of the adjustment results in Step 3. Corresponding plots of $p_s(r)$, $p_s(t)$ and $r_s(t)$ curves serve as illustrations and provide a visual check of the adjustment quality in all four steps. Examples of output plots are reproduced in Reference 1.

5.4. Structure of the Shock Fitting Program

The shock fitting program consists of a main program and 15 subroutines. Figure 2 shows a flowchart of the main program. The hierarchy of the various subroutines is shown in Figure 3 and the communications between the subroutines through COMMON blocks is displayed in Figure 4. A listing of the programs is given in Appendix A. The contents of the six COMMON blocks that are used in the shock fitting programs are as follows:

COMMON/AMBCHA/p_o, T_o, γ ,M, V, E, H, e_H.

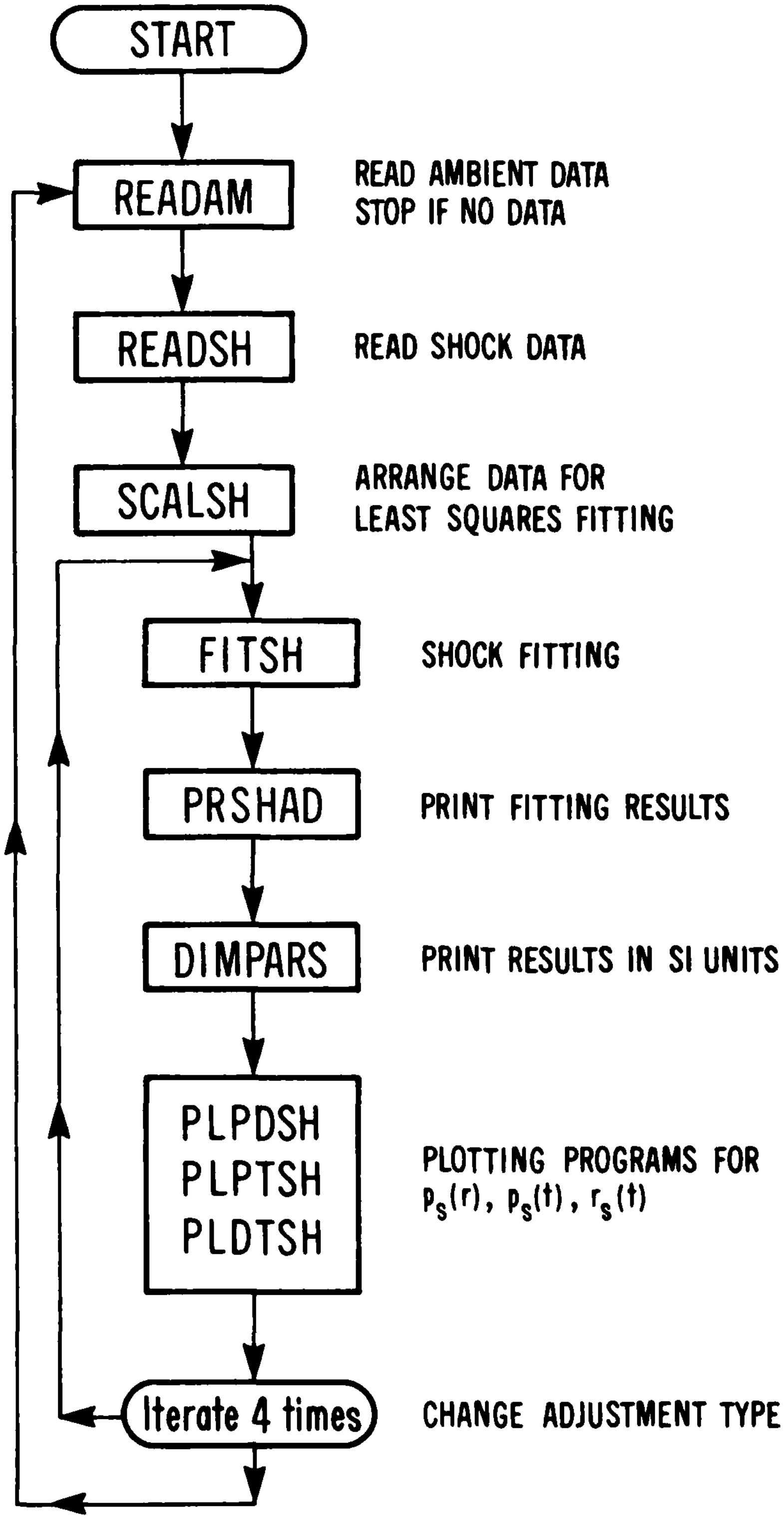


Figure 2. Main Program SHOKFIT for Shock Fitting

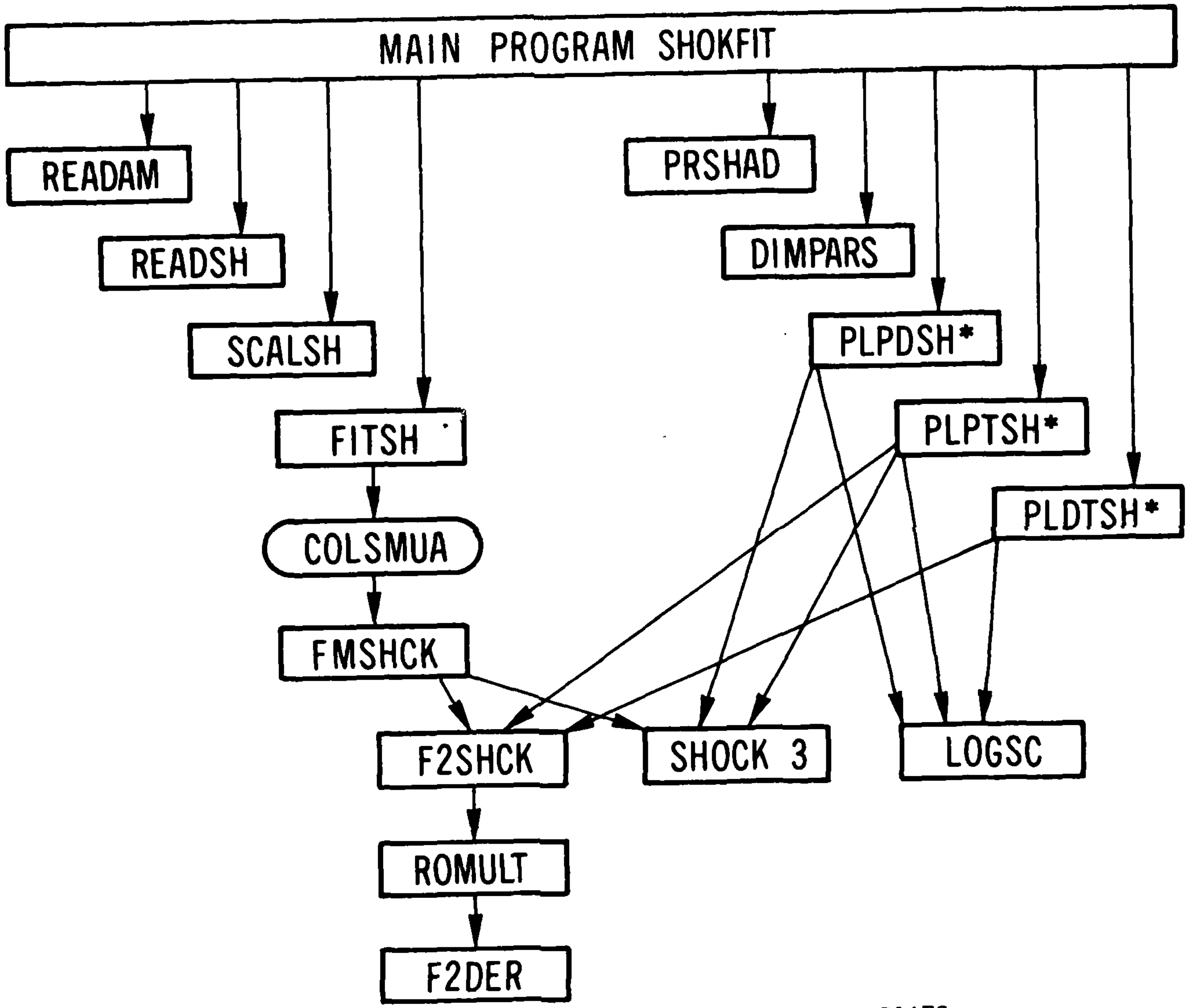


Figure 3. Hierarchy of the Shock Fitting Program BLAFS.

Arrows indicate subroutine calling direction. COLSMUA is a general utility routine for data fitting. Other general routines for plotting are used by the starred subroutines.

Subroutines \ Common Blocks	AMBCHA	CF2DER	CMISFM	CMPLSH	COMSHDT	PLOT
DIMPARS		×				
FITSH		×	⊗			
FMSHCK			×			
F2DER		×				
F2SHCK		⊗				
LOGSC				×		
PLDTSH	×	×	×		×	×
PLPDSH	×		×		×	×
PLPTSH	×	×	×		×	×
PRSHAD			×			
READAM	⊗					⊗
READSH					⊗	
SCALSH	×	⊗	⊗	⊗	×	

Figure 4. Access to COMMON Blocks by Shock Fitting Subroutines.

A circle indicates the subroutine which enters data into the COMMON block.

This block is filled by the subroutine READAM and its contents are

p_o = ambient pressure, Pa,

T_o = ambient temperature, K,

γ = ratio of specific heats,

M = molar mass, kg/mol,

V = volume of fire ball, m^3,

E = released energy, J,

H = height of burst, m,

e_H = standard error of H, m.

COMMON/CF2DER/ Γ, c_o, a, b, c, d, x_{min}, s_r, s_p, s_t.

This block is filled by the subroutines SCALSH and F2SHCK. Its contents are

Γ = $[(1+\gamma)/(2\gamma)](p_s/p_o)$, (factor in Equation 5.2),

c_o = $(\gamma T_o\ 8.3143/M)^{1/2}\ s_t/s_r$, (sound speed),

a, b, c, d = shock parameters, see Equations 5.1 and 5.2,

x_{min} = $(x_i/s_r)_{min}$,

s_r = distance scale, m,

s_p = pressure scale, Pa,

s_t = time scale, s.

COMMON/EMISFM/MISPDT(3,50), DISTN(50), NODIST, SCD.

This block is filled by the subroutines SCALSH and FITSH. Its contents are

MISPDT(3,50) = a non-zero in this array indicates a missing component of the observation vector (p_i, r_i, t_i), i = 1, ..., 50.

DISTN (50) = scaled distances r_i/s_r

NODIST = a non-zero indicates for the subroutine FMSHCK that the distances are not to be adjusted, but the values from DISTN used. This is set by the subroutine FITSCH.

SCD = distance scale s_r, m.

COMMON/CMPLSH/ p_{min}, p_{max}, r_{min}, r_{max}, t_{min}, t_{max}

This block is filled by the subroutine SCALSH and its contents are the extremes of the observed values of overpressure p (Pa), distance r (m) and time t (s).

COMMON/COMSHDT/TPXH(4,5Ø), ERTPXH(4,5Ø), TITLE(3), ALAB(2,5Ø)

This block contains the raw shock observations. It is filled by the subroutine READSH and its contents are

TPXH(4,5Ø) = observation vectors (t,p,x,h) for up to 5Ø observation sets. The units of the observations are (s, Pa, m, m).

ERTPXH(4,5Ø) = estimated standard errors of the observations in TPXH.

TITLE(3) = alphanumeric title of the computer run, read from the TITLE card.

ALAB(2,5Ø) = alphanumeric identifications of the observation sets.

COMMON/PLOT/PD(6),PLABL(4)

This block is filled by the subroutine READAM and it contains information for the plotting routines.

PD(6) = contents of the PLOTTING DATA card. Only the first two components are used: PD(1) = f_p, PD(2) = f. See Section 5.2.

PLABL(4) = label for Calcomp plots, read from the PLOTLABEL card.

6. BLAST FIELD OVERPRESSURE FITTING PROGRAM BLAFOP

6.1. Purpose of the Program

The purpose of the program is to determine from measurements of overpressure histories at a number of stations a model function that approximately describes the overpressure field within a limited region behind the shock. The model function has the form

$$p_f = [p_s(r) - C(r)]e^{\tau A(r) + \tau^2 B(r)} + C(r), \qquad (6.1)$$

where $\tau = t - t_s(r)$, (6.2)

$p_s(r)$ and $t_s(r)$ are known functions describing the incidental shock overpressure and arrival time, and A(r), B(r) and C(r) are unknown functions of the distance r from the center of the explosion, to be determined by the program. The region in which the fitted overpressure field function p_f approximates the overpressure field is indicated in Figure 1. The three adjustable functions of r are defined by

$$A(r) = (A_1 + A_2 r)/r^{n_A},$$
$$B(r) = (B_1 + B_2 r)/r^{n_B}, \qquad (6.3)$$
$$C(r) = C_1/r^{n_C}.$$

The three exponents, n_A, n_B and n_C, are determined by a trend analysis of the overpressure histories, and the functions $p_s(r)$ and $t_s(r)$ are determined by shock fitting (see Section 5). Thus the function given by Equation 6.1 contains five free parameters, A_1, A_2, B_1, B_2 and C_1, which are determined by a least squares approximation to the pressure history data. A program listing is given in Appendix B and the subroutines of the program are described in Section 8.

6.2. Input for the Blast Field Overpressure Fitting Program

The input consists of three parts: general data, results of the shock fitting described in Section 5, and overpressure history observations.

The general data are provided by three mandatory and three optional cards. The format and the contents of the cards are the same as for the general data input for shock fitting described in Section 5.2. (The cards are read by identical subroutines.) The end of the general data batch is indicated by a blank card.

The shock fitting results are provided by four cards in arbitrary order. The cards contain the shock fitting parameters and their error estimates. The format of all four cards is (2A1Ø,6E1Ø.3) and their contents are as follows

1	21
SHOCKPAR	a, b, c, d, r_o

1	21
SHOCKPARERRORS	e_a, e_b, e_c, e_d

1	21
SHOCKPARCORCOEF	c_{ab}, c_{ac}, c_{ad}, c_{bc}, c_{bd}, c_{cd}

1	21
SHOCKSCALESbR,P,T	s_r, s_p, s_t .

The end of the shock fitting data is indicated by a blank card.

The numerical contents of the four cards is normally taken from the results of the third step of shock fitting. (See Section 5.3.) The meaning of the contents of the cards is as follows

a, b, c, d = shock fitting parameters, see Equations 5.1 and 5.2.

r_o = shock distance for arrival time d.

e_a, e_b, e_c, e_d = standard errors of the shock fitting parameters. The standard error of weight one, e_o, generally should be included as a factor in these estimates, if e_o is larger than one, or deviates considerably from one.

c_{ab} through c_{cd} = correlation coefficients of the shock fitting parameters.

s_r, s_p, s_t = scales, in metres, pascals and seconds, of distance, pressure and time which are used to express the shock parameters. If the shock parameters are expressed in SI base units, then the scales are 1 m, 1 Pa and 1 s, respectively.

The third batch of input consists of cards containing overpressure history observations. Each overpressure history is entered by one card containing the range and elevation of the pressure transducer, and a number of other cards each containing an observed time and corresponding overpressure at the station. The number of t,p-observation sets must be at least four for each station. The total number of stations must be at least two and not more than 50, and the total number of t,p-observations in all stations is limited to 5000. All cards pertaining to one history, including the range and elevation card, should be in one batch. Their order within the batch is arbitrary. The format of the cards is (2A10, 6E10.3). The first word (A10) is a label identifying the station, that is, the overpressure history, and it should be the same in all cards belonging to that history. A different label indicates for the computer the beginning of a new batch pertaining to a different history.

The contents of the cards are as follows:

1	11 20	21
Label	RANGE,ELEV	x, e_x, h, e_h

1	11 20	21
Label	TIME,PRESb	t, e_t, p, e_p

where x = range (ground distance) of the station, m

e_x = standard error of x, m,

h = elevation of the station, m,

e_h = standard error of h, m,

t = time after detonation, s,

e_s = standard error of t, s,

p = overpressure at time t, Pa,

c_p = standard error of p, Pa.

The end of all data is indicated by a blank card.

The computing time for a typical case (5 histories, and a total of 150 t,p-observations) is less than 100 seconds on the CDC 7600.

6.3 Overpressure Field Fitting Process and Output

The overpressure field function is determined in two steps. First, a three parameter exponential function

$$p_h = (p_s + C)e^{A\tau + B\tau^2} - C, \tag{6.4}$$

with $\tau = t-t_s$, is fitted to each overpressure history. Then the dependence of the fitting parameters A, B and C of the individual histories on the distance r from the explosion is analyzed, and power function approximations are determined in the form

$$A(r) = A_o/r^{n_A}, \quad B(r) = B_o/r^{n_B}, \quad C(r) = C_o/r^{n_C}. \tag{6.5}$$

The ensuing values of the exponents n_A, n_B and n_C are used in Equation 6.3 to construct the overpressure field function.

The second step consists of a joint fitting of all observations to the overpressure model 6.1 through 6.3. Free parameters for that fitting are the five constants A_1, A_2, B_1, B_2, and C_1.

The output starts with a comprehensive summary of all input data. Next, the individual histories are fitted using a version of the least squares utility routine COLSAC (Reference 4) with the constraint function

$$f_i = p_h(t_i + c_{ti}; A, B, C) - (p_i + c_{pi}), \; i = 1,\ldots,s, \tag{6.6}$$

where t_i and p_i are the observed times and pressures, c_{ti} and c_{pi} are the corresponding residuals, and the function p_h is defined by Equation 6.4. (The function p_h is different for each history because the shock values p_s and t_s are different for each history.) COLSAC prints the adjustment results in a standard form, which is supplemented by a self-explaining list of adjusted data and parameter values. In addition, Calcomp plots are generated of each adjusted history, providing a visual check of data and adjustments. At the end of the first step a list of the parameters A, B and C of all histories is provided together with the exponents n_A, n_B and n_C, and the values of A_o, B_o, and C_o. The three parameters A, B and C are also shown in log,log-plots as functions of r.

In the second step, the joint fitting of all observations is done in substeps to avoid algorithmic difficulties. First, only overpressure observations are adjusted; then overpressure and time observations are adjusted, and finally, overpressure, time and distance observations are adjusted. The adjustments are again done by the COLSAC routine, now using constraints derived from the model function 6.1 through 6.3. The constraints are formulated as the function

$$f_i = p_f(r_i + c_{ri}, t_i + c_{ti}; A_1, A_2, B_1, B_2, C_1) - (p_i + c_{pi}) = 0, i = 1,\ldots s, \tag{6.7}$$

where p_f is defined by Equation 6.1. The output consists of the standard output by COLSAC, and after the third substep, a list of the adjusted observations and a list of the overpressure field parameters in SI base units. For each history a plot is provided of the overpressure field function, its confidence limits and the corresponding observations. A final plot gives in the r,t-plane the locations of the observed histories, the shock trajectory and some particle path lines. The latter plot can be used for the planning of experiments, because it provides an indication of the domain in which the flow field can be reconstructed and checked by test calculations. (See Figure 1.) Examples of the various plots are given in Reference 1.

6.4. Structure of the Overpressure Field Fitting Program

The overpressure field fitting program consists of a main program and 41 subroutines. Five of the subroutines (COLSACA, COLSACB, MTRINDB, LUDATD, LUELMD) belong to the least squares model fitting utility routine COLSAC (Reference 4), and usually are not included in a special application program, but attached as needed for a particular computer run. For the present application the set of routines was modified, and the program package contains the modified version. The modifications concern the use of the LEVEL2 option for certain arguments of these subroutines. LEVEL2 variables were necessary in order to accomodate the possibly large number of data within the present computer configuration at BRL. (The shock fitting program described in Section 5 uses a standard version of the least squares routine COLSMU, which is therefore not included in the program package, but attached at run time.)

A flowchart of the main program is shown in Figure 5. Most of the subroutines that are called from the main program are quite simple. The structures of the two more complicated subroutines, FITPR and FTPFLD, are illustrated by Figures 6 and 7. At a lower level, the subroutine PFIELD for the computation of the overpressure field is more involved and its hierarchy is shown in Figure 8.

A list of COMMON blocks is given in Figure 9 together with the names of subroutines which have access to the blocks. Seven of the 16 blocks are dummy blocks, and needed only because of idiosyncrasies of the LEVEL2 option. (They are not used to transmit information between different parts of the program.) Several other blocks are identical to those used in the shock fitting program, Section 5. A description of the contents of the COMMON blocks follows.

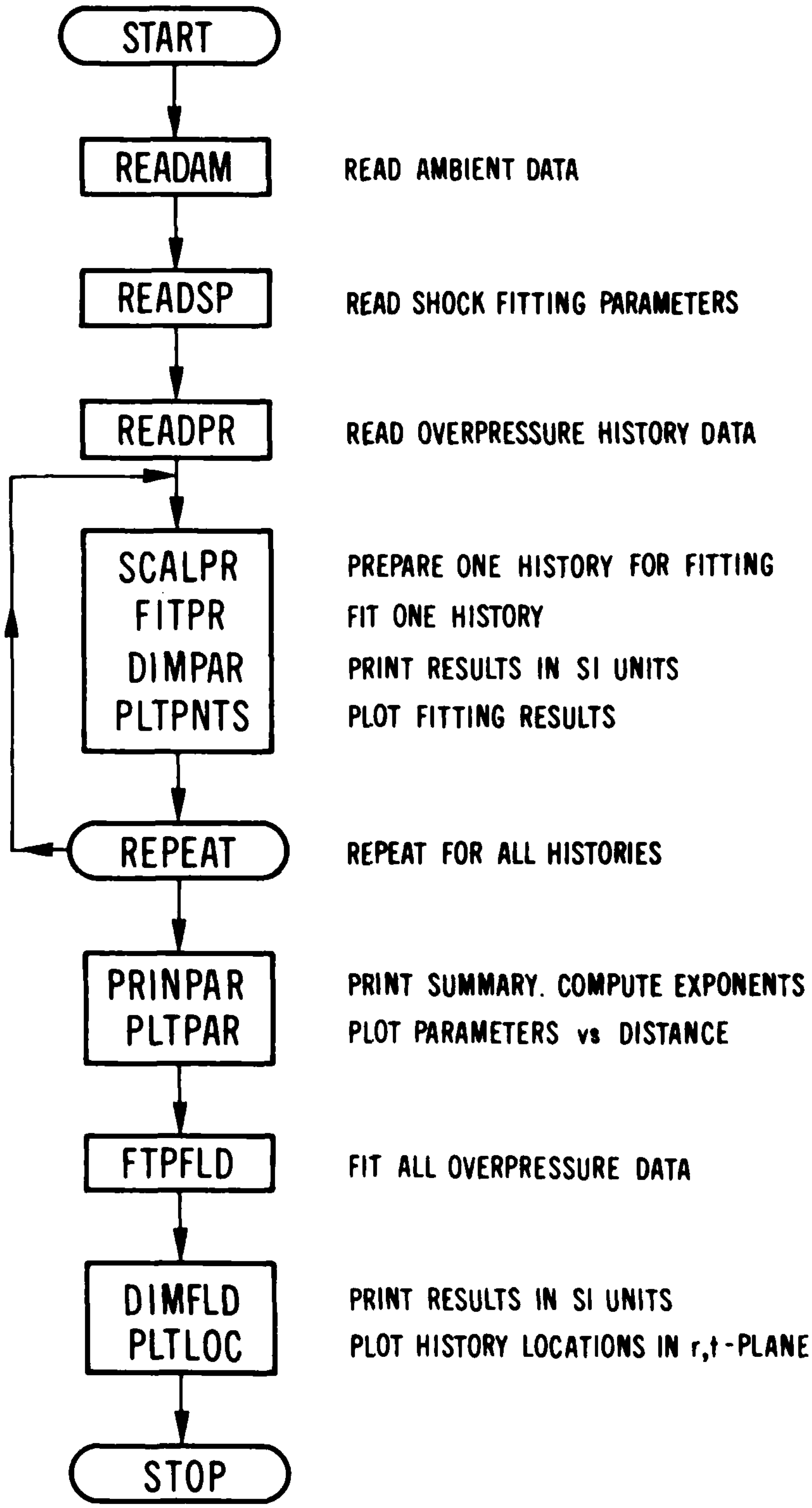
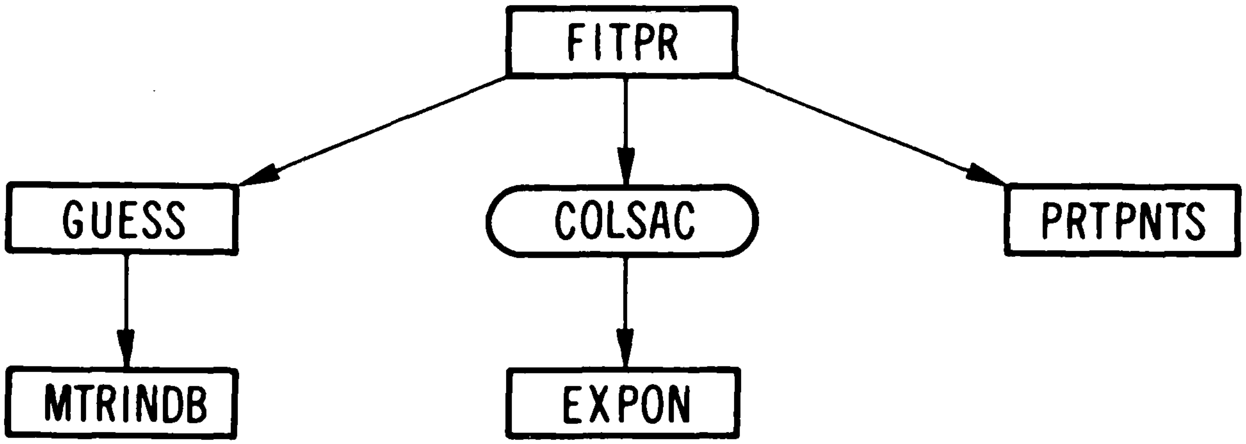


Figure 5. Main Program OPREFIT for Overpressure Field Fitting.



DETERMINE INITIAL APPROXIMATIONS OF PARAMETERS A,B & C OF OVERPRESSURE HISTORY, SEE EQ.(6.4). MTRINDB IS A LINEAR EQUATION SOLVER FROM THE COLSAC COMPLEX.

DETERMINE FINAL VALUES OF PARAMETERS A B & C. COLSAC IS THE LEAST SQUARES UTILITY ROUTINE COMPLEX. EXPON IS THE CONSTRAINT FUNCTION ACCORDING TO EQ (6.4). SEVERAL CALLS TO COLSAC ARE PROGRAMMED TO HANDLE ALGORITHMIC DIFFICULTIES.

PRINT ADJUSTED OBSERVATIONS IN A COMPREHENSIVE LIST.

Figure 6. Hierarchy of the Subroutine FITPR.

The subroutine handles overpressure fitting for a single history. Arrows indicate calling direction.

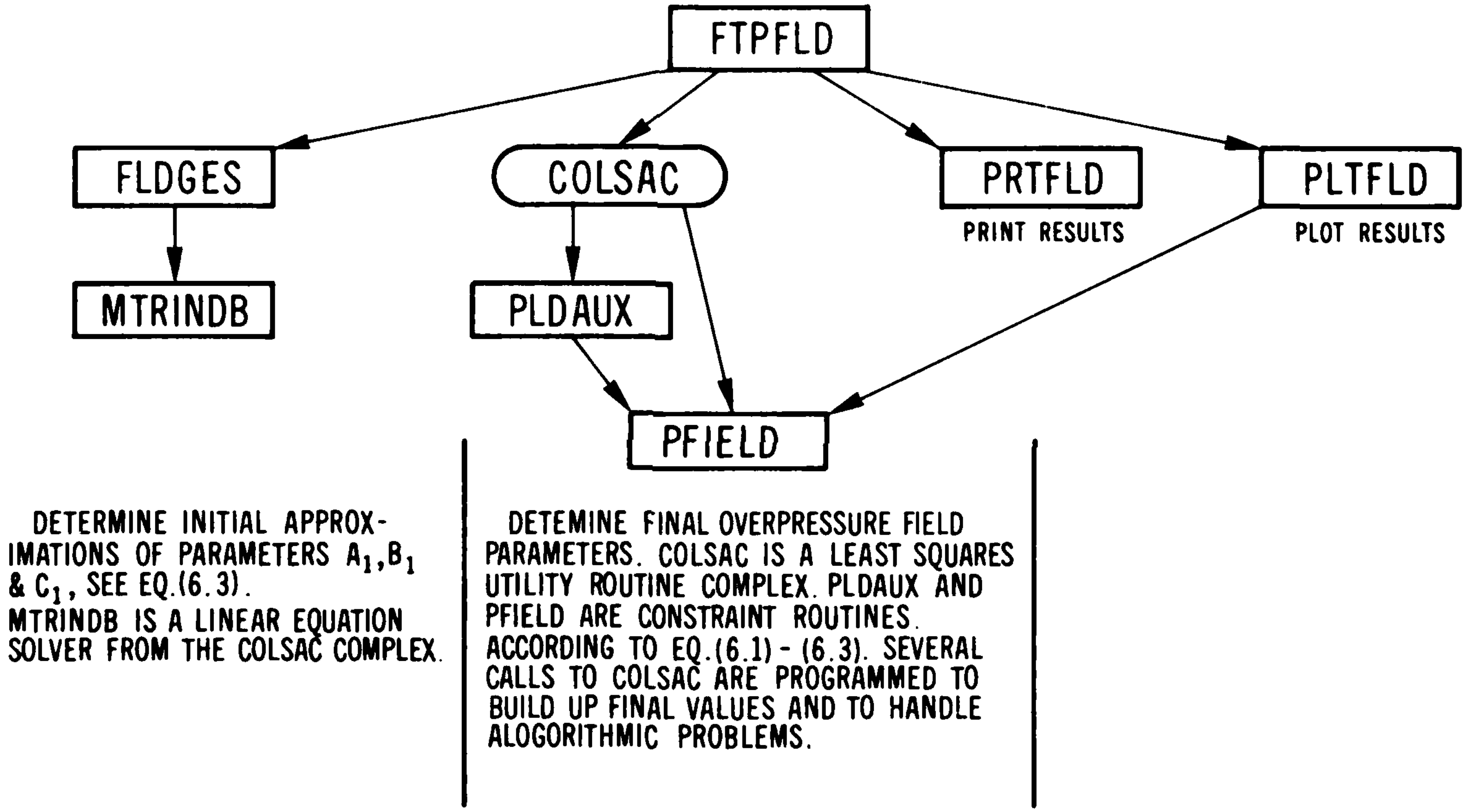


Figure 7. Hierarchy of the Subroutine FTPFLD.

The subroutine handles the total overpressure field fitting. Arrows indicate calling direction.

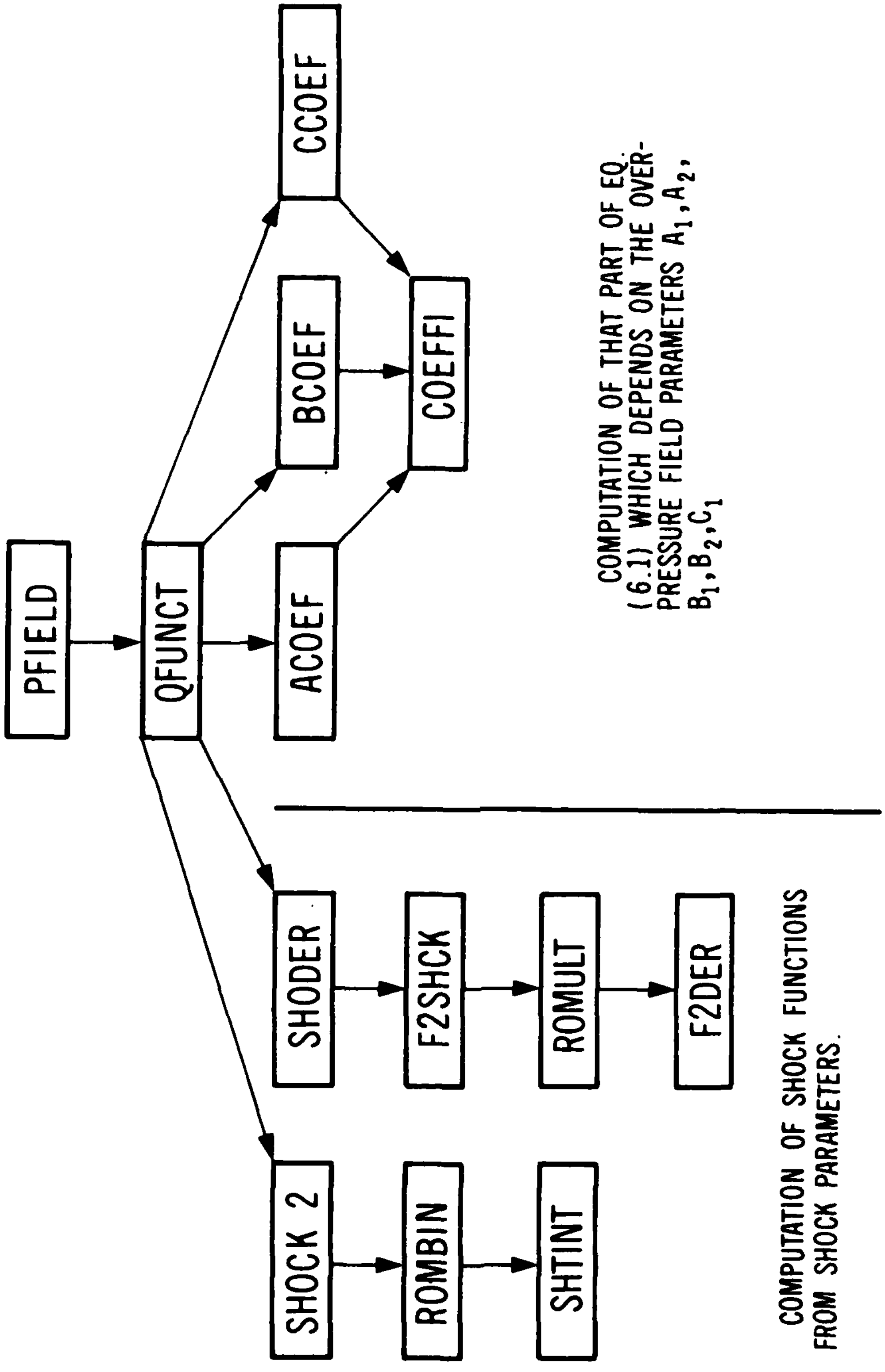


Figure 8. Hierarchy of the Subroutine PFIELD.
The subroutine computes the overpressure field function defined by equations 6.1 thru 6.3. Arrows indicate calling directions.

COMMON Block NAME and Length	Subroutines with access to the COMMON Block
AMBCHA, 8	READAM, READSP, READPR, PLTLOC, SHOCK, STRBEG
CFLDEX, 3	ACOEF, BCOEF, CCOEF, FTPFLD
CF2DER, 10	F2DER, FRSHCK, READSP, SHOCK, SHOCK2, SHTINT, SHODER, STRBEG
COMPR, 30150	FTPFLD, READPR, SCALPR, PLTFLD
COMSHK, 24	READSP, QFUNCT, SHOCK, SHOCK2, SHODER, STRBEG
CPARG, 155	PLTFLD
CSCALE, 3	FTPFLD,QFUNCT, PLTLOC, STRBEG, PLTFLD
GUECM, 60	GUESS, FLDGES
PLOT, 10	READAM, PLTPAR, PLTPNTS, PLTLOC, PLTFLD
PSTS, 2	FITPR, EXPON, PLTPNTS
SCRCH, 13660	FITPR
SCRCHA, 195	PLTPNTS
SCRCH2, 114307	FPTFLD, PLTFLD
SCRCH3, 155	STRLIN
SCRCH4, 140	PLDAUX
TPINDX, 2	FTPFLD, FLDGES, FIELD, QFUNCT, PRTFLD, STRLIN, PLTFLD

Figure 9. List of COMMON Blocks in the Overpressure Field Fitting Program BLAFOP

The underlined subroutines enter data into the COMMON Block.

COMMON/AMBCHA/ - see the description in Section 5.4.

COMMON/CFLDEX/n_A, n_B, n_C.

This block contains the three exponents in the field function, Equations 6.1 through 6.3. The block is filled by the subroutine FTPFLD.

COMMON/CF2DER/ - See the description in Section 5.4.

COMMON/COMPR/TP (2,5000), ERIP (2,5000), ALB(2,5000), NSET(50), DIST(50), EROIST(50).

This block contains the raw input from history observations. It is filled by the subroutine READPR. Its contents are

TP(2,5000) - time and pressure observations,

ERTP(2,5000) - corresponding standard errors,

ALB(2,5000) - labels of the observations,

NSET(50) - numbers of t,p-observations in each history; up to 50 histories are permitted,

DIST(50) - ranges (ground distances) of up to 50 pressure transducer locations,

ERDIST(50) - standard errors of the ranges in DIST.

COMMON/COMSHK/NPS,PAR(4),VPAR(4,4), s_r, s_p, s_t.

This block contains the shock fitting parameters and their variances. The block is filled by the subroutine READSP and its contents are

NPS - number of shock parameters; this is a set equal to four,

PAR(4) - shock parameters a,b,c,d,

VPAR(4,4) - variance-covariance matrix of the shock parameters,

s_r, s_p, s_t - length, pressure and time scales which are used to express the shock parameters.

COMMON/CPARG/

This is a dummy block, necessary to use the LEVEL2 memory option.

COMMON/CSCALE/s_r, s_p, s_t

This block contains the scales for distance, pressure and time which are used for the calculations in this program. They are set by FTPFLD in accordance with the general input.

COMMON/GUECM/

This is a dummy block, necesary to use the LEVEL2 memory option.

COMMON/PLOT/

See description in Section 5.4.

COMMON/PSTS/p_s,t_s

This block contains a shock overpressure and a corresponding shock arrival time. It is set by the subroutine FITPR.

COMMON/SCRCH/
COMMON/SCRCHA/
COMMON/SCRCH2/
COMMON/SCRCH3/
COMMON/SCRCH4/

Dummy blocks necessary to use the LEVEL2 memory option.

COMMON/TPINDX/i_t, i_p

This block contains two indices signifying the time and pressure components of the three component observation (p,t,r). Subroutine FTPFLD sets $i_t = 2$, $i_p = 1$.

7. BLAST FIELD HISTORY COMPUTATION PROGRAM BLAFHI

7.1. Purpose of the Program

The purpose of the program is to compute blast field histories at given locations using a previously determined overpressure field function. The computation process is schematically described in Section 2 and illustrated by Figure 1. It consists in essence of numerical integrations of a number of selected path line equations and of quadratures over flow field functions along lines t = const. The results of these calculations produce, at specified distances r, histories of overpressure p, particle velocity u, density ρ, dynamic pressure $\rho u^2/2$ and temperature T, all with estimated standard errors. A program listing is given in Appendix C and the subroutines of the program are described in Section 8.

7.2. Input for the Blast Field History Computation Program

The input consists of four parts: general data, results of the shock fitting described in Section 5, results of the overpressure field fitting described in Section 6, and instructions as to what calculations are to be done. The four data groups are entered as four batches of input cards, separated by a blank card at the end of each batch.

The general data are provided by three mandatory and three optional cards. The format and the contents of the cards are the same as for the general data input for shock fitting described in Section 5.2.

The shock fitting results are provided by the four cards described in Section 6.2.

The overpressure field fitting results are entered by seven cards containing the overpressure field parameters and their estimated standard errors. The format of the cards is (2A10, 6E10.3) and their order is arbitrary. The contents of the cards are as follows:

1	21
FIELDPAR	A_1, A_2, B_1, B_2, C_1

These are the five overpressure field parameters, see Equations 6.1 through 6.3.

1	21
FIELDPARERRORS	e_{A1}, e_{A2}, e_{B1}, e_{B2}, e_{C1}

These are the standard errors of the overpressure field parameters.

1	21
FIELDPARCOb1	c_{12}, c_{13}, c_{14}, c_{15}, c_{23}

1	21
FIELDPARCOb2	c_{24}, c_{25}, c_{34}, c_{35}, c_{45}

These cards contain the correlation coefficients between the overpressure field parameters.

1	21
FIELDPAREXPONENTS	n_A, n_B, n_C

These are the exponents in the overpressure field function, see Equation 6.3.

1	21
FIELDPARSCALES	s_r, s_p, s_t

Scales in metres, pascals and seconds, of distance, pressure and time that are used to express the overpressure field parameters.

1	21
FIELDPARRANGE	r_{min}, r_{max}

Distances in metres between which the overpressure field function is assumed to approximate the real overpressure.

The end of the pressure field data is indicated by a blank card.

The computing instructions are entered by one card for each set of histories that are to be calculated. The card has the format (2A10,6E10.3) and the following contents:

1	21
HISTORYbR,TMAX,NRPTS	r, t_{max}, n

where

r = distance from the center of explosion at which the histories should be computed, m,

t_{max} = end time for history calculations, s,

n = approximate number of nodes to be calculated; n should not exceed 100.

The program starts the calculations after a HISTORY card is read. After completing calculations the program tries to read the next HISTORY card. A blank card indicates the end of the input and will cause the program to stop.

A typical computing time for a history with 80 nodes is 150 s on the CDC 7600.

7.3 Blast Field History Computation Process and Output

A short description of the computation process is given in Section 2 and the process illustrated by Figure 1. More detailed information about the numerical integration of the path line and derivative equations is given in Reference 1, Section 3. The actual history is obtained at the prescribed distance r and for equidistant time values by interpolation in the r,t-plane between path lines. Details of the interpolation process are given in the description of the subroutine FLINTER.

The output of the program consists of a comprehensive summary of all input data, that is, the general (ambient) conditions, the shock fitting results and the overpressure field fitting results, followed by a printed list of the computed histories. The list contains values of time t, overpressure p, velocity u, density ρ, and dynamic pressure $\rho u^2/2$, all with estimated standard errors, at equidistant time intervals. In addition to these histories a list of the test velocities is printed together with the corresponding original velocities and the dynamic pressures computed using the test velocities.

The printed output is supplemented with plots of the five histories of p, $u, \rho, \rho u^2/2$ and T, and a plot of the dynamic pressure history computed using the test velocities instead of the original velocities. Examples of the plots are given in Reference 1.

7.4. Structure of the Blast Field History Computation Program

The program consists of a main program and 28 subroutines. Most of the subroutines are identical to those used in the shock fitting and the pressure field fitting programs. A flowchart of the main program is shown in Figure 10, and a flowchart of the principal subroutine FLOWFLD is shown in Figure 11. The routine computes the flow history at $r = r_B$ by calculating a number of particle path lines (each line is generated by calling STRBEG and STRLIN) and by interpolation between the lines to obtain history values at $r = r_B$ and for equidistant t-values. After calculations are completed the output routines PRIHIS, UTEST and PRITST are called to print results and to compute test velocities. Other subroutines of the program have quite simple structures. The somewhat more involved structure of PFIELD is shown in Figure 8. Short descriptions of all subroutines are given in Section 8.

A list of COMMON blocks is given in Figure 12, showing also the names of those subroutines which have access to the various blocks. Most of the COMMON blocks have the same contents as corresponding blocks in the other two program parts, BLAFS and BLAFOP. Next, we give a description of the COMMON blocks.

COMMON/AMBCHA/

This block contains general data and is described in Section 5.4.

COMMON/CFLDEX/n_A, n_B, n_C

This block contains three exponents of the overpressure field function, and it is filled up by the subroutine READFP. (See also Section 6.4.)

COMMON/CF2DER/ - See the description in Section 5.4.

COMMON/COMFLD/P(5),V(5,5),s_r, s_p, s_t, r_{min}, r_{max}

This block is filled by READFP and it contains the parameters of the overpressure field function. The contents of the block are

P(5) = $(A_1, A_2, B_1, B_2, C_1)$ = overpressure field parameter vector;

V(5,5) = variance-covariance matrix of the parameter vector P;

s_r, s_p, s_t = scales in metres, pascals and seconds of distance, pressure and time in which the parameters P are expressed;

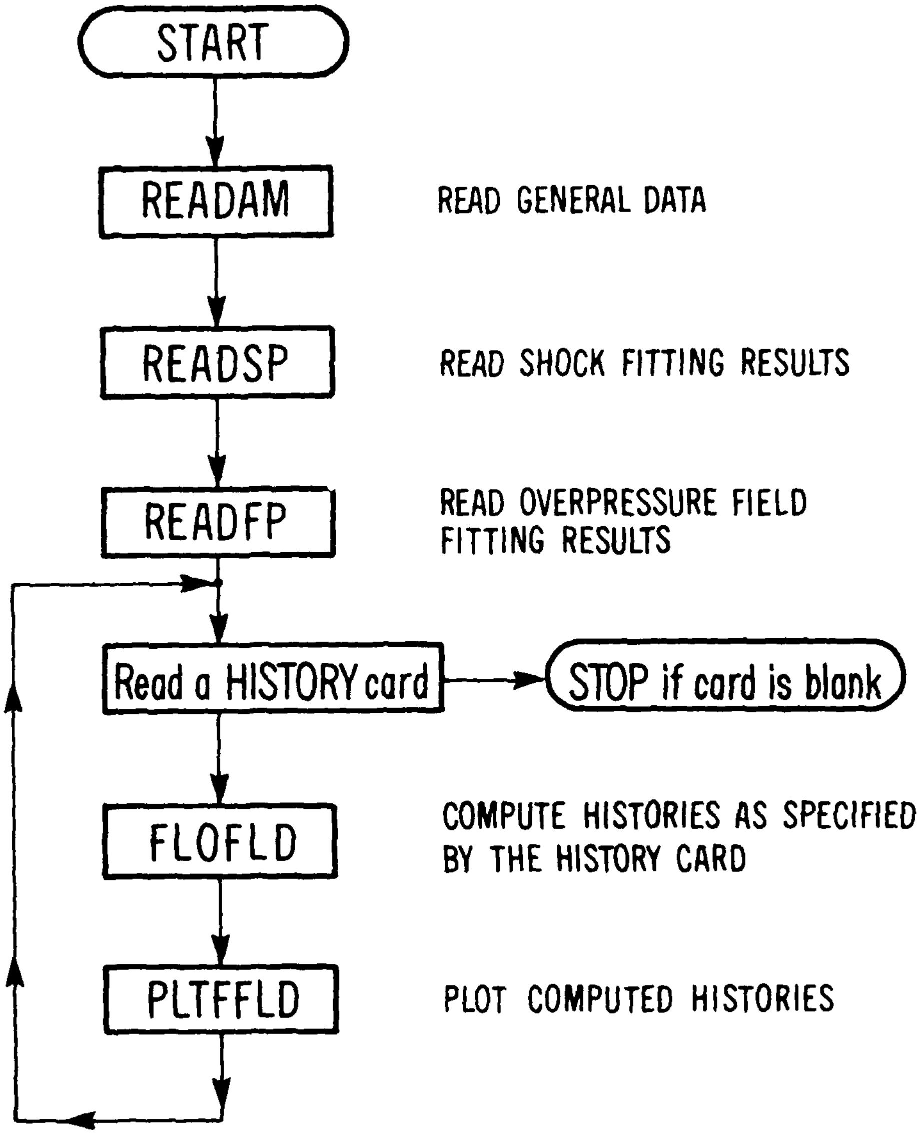


Figure 10. Main Program HISTORY for Flow History Computation.

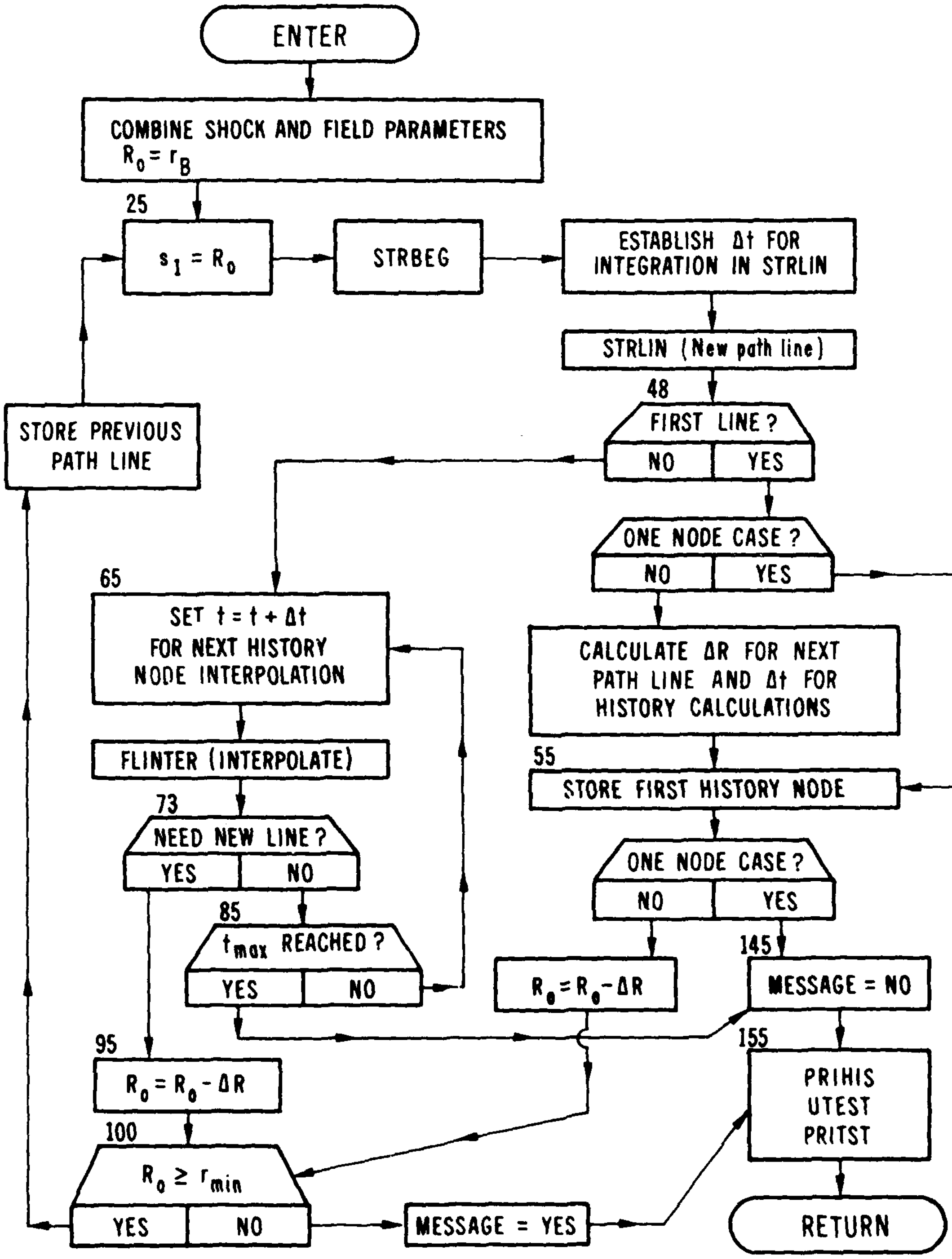


Figure 11. Flowchart of Subroutine FLOFLD.

Subroutines \ Common Blocks	AMBCHA	CFLDEX	CF2DER	COMFLD	COMSHK	COUTST	CSCALE	PLOT
ACOEF		×						
BCOEF		×						
CCOEF		×						
FLOFLD	×				×		⊗	
F2SHCK			⊗					
PLFFLD	×							×
PRIHIS							×	
PRITST							×	
QFUNCT					×		×	
READAM	⊗							⊗
READFP		⊗		⊗				
READSP	×		⊗		⊗			
SHOCK	×		×		×			
SHOCK 2			×		×			
SHODER			×		×			
SHTINT			×					
STRBEG	×		×		×		×	
UTEST	×					⊗		
UTINT						×		
MAIN PR.		×		×				

Figure 12. Access to COMMON Blocks by History Computing Subroutines.

A circle indicates data entry into the COMMON Block.

r_{min}, r_{max} = distance range in metres for which the overpressure field function is assumed to hold.

COMMON/COMSHK/ - See the description in Section 6.4.

COMMON/COUTST/t,P(1Ø), γ, p_o

This block contains information about the test computation by the quadrature Equation 2.12. It is filled by the subroutine UTEST and its contents are

t = time for which the integration is done, expressed in the s_t units that are used for calculations

P(1Ø) = the nine overpressure field parameters A_1, A_2, B_1, B_2, C_1, a, b, c, d. The tenth component of P is not used.

γ = ratio of specific heats,

p_o = ambient pressure expressed in the s_p units that are used for calculations.

COMMON/CSCALE/ - see the description in Section 6.4.

COMMON/PLOT/ - see the description in Section 5.4.

8. DESCRIPTIONS OF SUBROUTINES

This section contains short descriptions of all subroutines in alphabetical order. The listings of the subroutines in Appendices A, B and C contain additional comments. Some subroutines are used in more than one of the BLAF programs, and listed in more than one Appendix, as indicated in the headings of the following descriptions.

ACOEF (Appendices B and C)

This subroutine computes the function 6.3,

$$A(r) = (A_1 + A_2 r)/r^{n_A}$$

and its first and second order derivatives with respect to t,p,r and the five parameters PAR = (A_1,A_2,B_1,B_2,C_1). It is called from QFUNCT and it uses COEFFI for the actual calculations. The conventions for the arguments are

t = X(1,...), p = X(2,...), r = X(3,...).

BCOEF (Appendices B and C)

This routine computes the function 6.3

$$B(r) = (B_1 + B_2 r)/r^{n_B}$$

and its first and second order derivatives. Its structure and conventions are the same as those of ACOEF.

CCOEF (Appendices B and C)

This subroutine computes the function 6.3,

$$C(r) = C_1/r^{n_C},$$

and its first and second derivatives. (See also ACOEF.)

COEFFI (Appendices B and C)

This is an auxiliary routine for ACOEF, BCOEF and CCOEF and it calculates the function

$$A = (p_1 + p_2 r)/r^{ex}$$

with its first and second derivatives with respect to r, p_1 and p_2.

COLSACA, COLSACB (Appendix B)

This is a version of the COLSAC routine (Reference 4), modified to conform with the LEVEL2 memory option for certain of its arguments. The COLSAC routines are general least squares adjustment routines for scalar constraints, generally non-linear in terms of the observations and parameters.

DIMFLD (Appendix B)

This routine computes the overpressure field parameter values in base SI units, and prints a comprehensive summary of the parameters and their estimated errors. The routine is called from the main program for overpressure field fitting after completed calculations, and DIMFLD produces the last page of printed output for that program. Information from this page is used as input for the history calculation program.

DIMPAR (Appendix B)

This routine computes and prints the individual overpressure history paramaters A, B and C of Equation 6.4 in base SI units. It is called from the main program for overpressure field fitting after the individual fitting of each overpressure history.

DIMPARS (Appendix A)

This routine is called from the main program for shock fitting after each of the four fitting steps. It calculates the shock fitting parameters in base

SI units and prints a comprehensive list of the shock parameters and their estimated variances.

ERELCM (Appendix B)

This routine computes 201 nodes of an error ellipse for a given variance-covariance matrix. It is used by several plotting routines.

EXPON (Appendix B)

This is the constraint routine for the three parameter exponential function, Equation 6.4. It computes

$$f = (p_{shock} + C)e^{A\tau + B\tau^2} - C - p$$

where $\tau = t - t_{shock}$

and the first and second derivatives of f. EXPON is used as constraint by FITPR when the latter routine calls the least squares routine COLSACA to fit an individual overpressure history.

FITPR (Appendix B)

This routine is called by the main routine for overpressure field fitting to carry out a fitting of an overpressure history. Figure 6 shows the hierarchy of FITPR.

FITSH (Appendix A)

This routine is called from the main program for shock fitting. It prepares the shock data for least squares fitting and calls the fitting routine COLSMUA. A modifier KA in the argument of FITSH indicates which observations (pressure, distance, time) should be adjusted and the data preparation is done accordingly. The constraint routine for the fitting is FMSHCK.

FLDGES (Appendix B)

This routine is called from FTPFLD to provide initial estimates for the overpressure field parameters. Of the five parameters in Equation 6.3, the initial estimates of A_2 and B_2 are zero. The estimates A, B and C of A_1, B_1 and C_1 are computed by the following algorithm.

The constraint corresponding to Equations 6.1 through 6.3 can be expressed for $A_2 = B_2 = 0$ by

$$\ln\left[\frac{p - C/r^{n_C}}{p_s - C/r^{n_C}}\right] - A\,\frac{t - t_s}{r^{n_A}} - B\,\frac{(t - t_s)^2}{r^{n_B}} = 0.$$

Let $\bar{C}$ be an approximation to C. Then the above equation can be linearized in terms of a correction epsilon ϵ of $\bar{C}$ with the result

$$\ln\left[\frac{p-\bar{C}/r^{n_C}}{p_s-\bar{C}/r^{n_C}}\right] - \epsilon\,\frac{(p_s - p)/r^{n_C}}{(p-\bar{C}/r^{n_C})\;(p_s-\bar{C}/r^{n_C})}$$

$$- A\,\frac{t-t_s}{r^{n_A}} - B\,\frac{(t-t_s)^2}{r^{n_B}} = 0 .$$

We use this equation as a constraint equation with the first term as "observation." We define for each observed point

$$y_i = \ln\left[\frac{p_i - \bar{C}/r_i^{n_C}}{p_{si} - \bar{C}/r_i^{n_C}}\right] ,$$

$$\gamma_i = \frac{(p_{si} - p_i)/r_i^{n_C}}{(p_i - \bar{C}/r_i^{n_C})\;(p_{si} - \bar{C}/r_i^{n_C})} ,$$

$$\alpha_i = (t_i - t_{si})/r_i^{n_A} ,$$

$$\beta_i = (t_i - t_{si})^2/r_i^{n_B} ,$$

and $$w_i = (p_i - \bar{C}/r_i^{n_C})^2/e^2_{pi} ,$$

where e_{pi} is the estimated standard error of the observation p_i. As the least squares objective function we chose

$$W = \sum_{i=1}^{s} (y_i - \alpha_i A - \beta_i B - \gamma_i \epsilon)^2 w_i .$$

The normal equations for this problem are

$$A\sum w_i\alpha_i^2 + B\sum w_i\alpha_i\beta_i + \varepsilon\sum w_i\alpha_i\gamma_i = \sum w_i\alpha_i y_i$$

$$A\sum w_i\alpha_i\beta_i + B\sum w_i\beta_i^2 + \varepsilon\sum w_i\beta_i\gamma_i = \sum w_i\beta_i y_i$$

$$A\sum w_i\alpha_i + B\sum \gamma_i\beta_i\gamma_i + \varepsilon\sum w_i\gamma_i^2 = \sum w_i\gamma_i y_i .$$

The subroutine FLOGES solves these normal equations and iterates four times, replacing $\bar{C}$ by $\bar{C} + \varepsilon$ after each iteration. The initial approximation $\bar{C}$ is furnished by the calling program. In order to avoid unreasonable $\bar{C} + \varepsilon$ due to a bad initial guess the following restrictions are applied to the corrected values at each iteration:

$$-0.5(p_i r_i^{n_C})_{max} \le \bar{C} + \varepsilon \le (p_i r_i^{n_C})_{min} - 0.001 \left| p_i r_i^{n_C} \right|_{max} .$$

FLINTER (Appendix C)

This is an interpolation routine. It is called by the subroutine FLOFLD to interpolate between two given particle paths and calculate at a specified point in the r,t-plane the vector of flow variables ($p, u, \rho, u^2\rho/2$) and the corresponding variance-covariance matrix. The interpolation is done in two steps. First, along each particle path the point with the prescribed time is determined by linear interpolation. Then a linear interpolation is done between these two nodes in the r-direction. Error returns are programmed for cases which would require extrapolation.

FLOFLD (Appendix C)

This subroutine is called from the main program for blast field history calculations and it is the most important subroutine of that program. A flowchart of FLOFLD is shown in Figure 11. The program computes the history at a given location (given distance r) and calls other subroutines to print the results and to compute the test velocity according to Figure 1. In order to calculate the history, FLOFLD computes a series of particle path lines (by calling STRBEG and STRLIN). When two lines are computed and stored, FLOFLD calls FLINTER to calculate the flow variables at specified r,t-nodes by interpolation between the two path lines. If this requires an extrapolation, FLINTER returns with a corresponding error indicator. FLOFLD then calculates a new particle path, starting at a proper initial point, discards one of the previous path lines and calls FLINTER again. After all required nodes of the history have been computed, the program calls PRHIS to print the results, UTEST to compute test velocities and PRTST to print the test velocities.

FMSHCK (Appendix A)

This is the constraint routine, Equation 5.3, for the shock fitting. The particular form of the constraint function and its derivatives are given in

Reference 1, pages 21-23. The program is called from the least squares subroutine COLSMU. It contains some logic to handle observations with missing time or pressure values. Information about missing data is passed to FMSHCK through the COMMON/CMISFM/. The routine uses SHOCK3 and F2SHCK to compute the two components of the constraint function.

FTPELD (Appendix B)

This subroutine is called from the main program for overpressure field fitting. It takes the raw input data from COMMON/COMPR/, stores the data in arrays according to the requirements of the COLSAC routine, calls FLDGES to obtain initial approximations for the overpressure field parameters, and calls the least squares routine COLSAC to compute their final values. The adjustment results are printed by calling the subroutine PRTFLD and plotted by calling the subroutine PLTFLD. Normally there are three successive calls to COLSAC: for adjusting pressure; pressure and time; and pressure, time and distance, respectively. Other calls are programmed to handle cases with algorithmic troubles in COLSAC. Such problems can arise if the initial approximations of the parameters are bad and/or large residuals are present.

F2DER (Appendices A,B, and C)

The calculation of the shock arrival time by Equation 4.2, and its derivatives requires the numerical evaluation of nine integrals (see Reference 1, pages 22-23). These integrals are calculated simultaneously by a special Romberg routine (ROMULT). The subroutine F2DER computes the nine components of the integrand, and it is called from ROMULT, which is activated by F2SHCK.

F2SHCK (Appendices A,B, and C)

This subroutine represents the second component of the constraint for shock fitting, Equation 5.2. The constraint is formulated in the form

$$f_2 = (t_s-d)\ c_o + (d-t_i-c_{ti})\ c_o = 0,$$

where $t_i + c_{ti}$ is the corrected time observation and t_s-d is the integral in Equation (5.2). The formal derivatives of this function are listed in Reference 1, pages 22-23. The subroutine computes the function f_2 and its first and second order derivatives. In programs other than the shock fitting program, F2SHCK is used to compute the shock arrival time for a given distance, and the corresponding derivatives.

GRAPH (Appendix C)

This is an auxiliary routine for the plotting routine PLFFLD. It establishes scales and plots those parts of the legend that are common to all plots.

GUESS (Appendix B)

This routine provides initial estimates of the overpressure history function parameters for individual history fitting. It is called from the subroutine FITPR (see Figure 6). The initial estimates are obtained by solving a linearized version of the nonlinear problem defined by Equation 6.4. The linearization is done by expressing the constraint in the form

$$\ln(p-\hat{C}) - \ln(p_s - \hat{C}) = A\tau + B\tau^2,$$

where $\tau = t - t_s$, and linearizing this expression with respect to a correction ε of the approximation C:

$$\ln \frac{p-\hat{C}}{p_s-\hat{C}} = \varepsilon \frac{p_s-p}{(p_s-\hat{C})\ (p-\hat{C})} + A\tau + B\tau^2.$$

This expression is linear with respect to ε, A and B. We use it in a least squares algorithm as follows. First, we define for each observed p_i, t_i the quantities

$$y_i = \ln \frac{p_i - \hat{C}}{p_s - \hat{C}}$$

$$\gamma_i = \frac{p_s - p_i}{(p_s-\hat{C})\ (p_i-\hat{C})}$$

$$\tau_i = t_i - t_s$$

$$w_i = (p_i - \hat{C})^2$$

and formulate an objective function by

$$W = \sum_{i=1}^{s} (y_i - \varepsilon\gamma_i - A\tau_i - B\tau_i^{\,2})^2\ w_i\ .$$

If one considers the y_i as observations, then the normal equations for this problem are

$$A\sum w_i\tau_i^2 + B\sum w_i\tau_i^3 + \varepsilon\sum w_i\gamma_i\tau_i = \sum w_i y_i \tau_i,$$

$$A\sum w_i\tau_i^3 + B\sum w_i\tau_i^4 + \varepsilon\sum w_i\gamma_i\tau_i^2 = \sum w_i y_i \tau_i^2,$$

$$A\sum w_i\gamma_i\tau_i + B\sum w_i\gamma_i\tau_i^2 + \varepsilon\sum w_i\gamma_i^2 = \sum w_i y_i \gamma_i.$$

The subroutine solves this system of equations (calling MTRINDB), replaces $\hat{C}$ by $\hat{C} + \varepsilon$ and iterates four times. For this iteration the initial values are $A = 0$, $B = 0$, and $\hat{C} = \min(0, p_i - 0.05\, p_s)$. In order to avoid unreasonable values of $\hat{C} + \varepsilon$, the following restrictions are applied after each iteration

$$-0.5\, p_s \leq \hat{C} + \varepsilon \leq p_{imin} - 0.05\, p_s.$$

Because the signs of the parameters $\hat{C}$ and of the parameter C in Equation 6.4 (used in the subroutine EXPON) are reversed, the negative value of $\hat{C}$ is communicated as parameter C to the calling routine.

LOGSC (Appendix A)

This is an auxiliary routine for the plotting of shock fitting results. The routine establishes proper plotting scales for logarithmic plotting.

LUDATD, LUELMD (Appendix B)

These are modified IMSL routines for the solution of linear equations. They are part of the least squares package COLSAC and are included here because the use of the LEVEL2 memory option makes a special version of the routines necessary.

MTRINDB (Appendix B)

This is a matrix inversion routine. It belongs to the least squares package COLSAC and is included here because the use of the LEVEL2 option makes a special version of this routine necessary.

PFIELD (Appendices B and C)

This subroutine represents the overpressure field model function defined by Equations 6.1 through 6.3. It has two entries. If entry PFIELD is used then the function

$$f = (p_s - C)e^{A\tau + B\tau^2} + C - p$$

is computed including its first and second order derivatives with respect to t,p,r, the five overpressure field parameters A_1, A_2, B_1, B_2 and C_1, and the four shock parameters a, b, c and d. If the entry PFIELDC is used, then the derivatives with respect to the shock parameters are not computed. The latter entry is used as a constraint routine for the overpressure field fitting. The entry PFIELD is used for the computation of the overpressure field with corresponding accuracy estimates. Formulas for the derivatives of f are given in Reference 1, Section 6. The hierarchy of the routine is shown in Figure 8.

PLDAUX (Appendix B)

This is an auxiliary routine that permits one to make an overpressure field fitting with the model function of Equations 6.1 through 6.3, simplified by $A_2 = 0$ and $B_2 = 0$. It is used as a least squares constraint routine by FIPFLD if fitting with the full constraint function PFIELD (entry PFIELDC) is not possible because of algorithmic difficulties.

PLDTSH (Appendix A)

This is the plotting routine to plot shock distance as a function of time with corresponding confidence limits. The plot also contains the shock distance and arrival time observations.

PLFFLD (Appendix C)

This is the plotting routine for the flow field history computation program. It generates five history plots: overpressure, particle velocity, density, dynamic pressure, temperature, and dynamic pressure computed from the test velocity. All plots except for the last one include confidence limits and the velocity plot also contains the history of the test velocity.

PLPDSH (Appendix A)

This is the plotting routine to plot shock overpressure versus distance with corresponding confidence limits and observations.

PLPTSH (Appendix A)

Plotting routine to plot shock overpressure versus shock arrival time with corresponding confidence limits and observations.

PLTFLD (Appendix B)

This routine is called from FTPFLD after adjustment of the overpressure field to plot at the observation sites the observed overpressures and the adjusted overpressure histories. The plots provide a visual check of the adjustment results and a comparison with the individual pressure history adjustment plots by PLTPNTS.

PLTLOC (Appendix B)

This routine is called from the main program for overpressure field fitting after completed calculations. The routine plots in the r,t-plane the shock trajectory, the locations of the observed histories and five particle path lines.

PLTPAR (Appendix B)

This subroutine plots in a log,log-scale the absolute values of the overpressure history parameters A, B and C (see Equation 6.4, Section 6.3) versus the distances of the histories. The plot provides a visual check for anomalies of individual histories and for the validity of the assumed dependence of the parameters on a power of the distance.

PLTPNTS (Appendix B)

This routine plots the overpressure history observations and the corresponding individual history fitting results (first fitting step, Section 6.3). It is called from the main program for overpressure field fitting after the fitting of each individual history.

PRIHIS (Appendix C)

This routine is called from the subroutine FLOFLD (see Figure 11) after completed calculation of a flow field history. It prints a history table containing t, p, u, ρ, $u^2\rho/2$ and corresponding estimates of standard errors.

PRINPAR (Appendix B)

This routine is called from the main program for overpressure field fitting after the adjustment of all individual histories (see Section 6.3, first adjustment step). It prints two lists of the parameters A, B and C with their standard errors for all histories, one in the scales used for the computation and the other in base SI units. The subroutine also computes the exponents n_A, n_B and n_C for the overpressure field function and initial estimates of the field function parameters A_1, B_1 and C_1. (These estimates are improved by FLDGES before the actual field fitting is started, see Figures 5 and 7.) The computation of the exponents is done as follows:

Let D_i be a parameter determined at the distance r_i. We determine a function Dr^n by minimizing the objective function

$$W = \sum_{i=1}^{s} (\ln|D_i| - \ln|D| - n \ln r_i)^2 D_i^2 .$$

The normal equations for this problem are

$$\ln|D| \sum D^2_i + n \sum D^2_i \ln r_i = \sum D^2_i \ln |D_i|$$

$$\ln|D| \sum D^2_i \ln r_i + n \sum D^2_i (\ln r_i)^2 = \sum D^2_i \ln r_i \ln |D_i|.$$

The solution of this system provides the exponent n and $\hat{D} = D \quad sgnD_1$, where D_1 is the parameter corresponding to the smallest distance r_i. The exponents n_A, n_B, and n_C are rounded to one decimal and the D(C) is used as an initial estimate of the parameter C_1 by FLDGES.

PRITST (Appendix C)

This routine prints results of the computation of the test velocity (see Figures 1, 10 and 11) by Equation 2.12. It also calculates and prints the dynamic pressure $u^2\rho/2$, computed using for u the test velocity instead of the original particle velocity. The subroutine is called from FLOFLD after the completion of calculations of the histories and after calling UTEST to compute the test velocities.

PRSHAD (Appendix A)

This routine prints shock observations, their standard errors and the corresponding adjusted values of the observations. It is called from the main program for shock fitting after each adjustment (see Figure 2).

PRTFLD (Appendix B)

This routine prints all overpressure field observations, their standard errors and their least squares residuals. It is called from FTPFLD after completing the overpressure field adjustment (see Figure 7). Observations belonging to different histories are printed in different tables.

PRTPNTS (Appendix B)

This routine prints the overpressure fitting results for individual history adjustments. It is called from FITPR (see Figure 6) after the least squares adjustment of data from one history.

QFUNCT (Appendices B and C)

This routine computes the exponent Q in the overpressure field function, Equations 4.3, 4.4 or 6.1, and all first and second order derivatives of Q. It is called from the subroutine PFIELD which computes the overpressure field (see Figure 8).

READAM (Appendices A,B and C)

This routine reads the data cards containing ambient conditions and general data (first batch of cards), and prints their contents in a comprehensive format. It is called by the main programs of all three programs.

READFP (Appendix C)

This routine reads the overpressure field fitting results (field parameters and their accuracies) in the form of seven cards (see Section 7.2). It is called by the main program for history calculations (see Figure 10).

READPR (Appendix B)

This routine is part of the overpressure field fitting program (see Figure 5). It is called from the main program and it reads all pressure history data from cards described in Section 6.2.

READSH (Appendix A)

This routine reads shock data from SHOCK and RANGE cards, see Section 5.2 and Figure 2. The routine is called from the main program for shock fitting. The input is printed out by this routine in a simple list.

READSP (Appendices B and C)

This routine reads the cards with the results from shock fitting (shock parameters, their error estimates, etc.). The input is described in Section 6.2. The routine is called from the main programs for overpressure field fitting and history calculations. After reading and checking the data for completeness, READSP prints the input data in a comprehensive format.

ROMBIN (Appendices B and C)

This is a Romberg integration routine. It is used by the routine SHOCK2 to compute the shock arrival time at a given distance according to Equation 4.2. The arguments of ROMBIN have the following meaning.

F = name of the subroutine that computes the integrand.

A,B = integration limits

FINT = integral value

NBAD = error indicator, set equal to zero if the integral has been computed, and equal to a non-zero value if the integral cannot be computed.

The repeated subdivision of the integration interval is limited to 20 steps and the convergence test is on the changes in the latest row of extrapolated values. If at least one relative change of less that 10^{-10} is detected, then the highest order extrapolated term is taken as the final result.

ROMBIN2 (Appendix C)

This routine is the same as ROMBIN. It is used by UTEST to compute the integral given in Equation 2.12. Because the integrand contains the function $t_s(r)$ which is calculated using ROMBIN, a second copy of the general integration routine was needed.

ROMULT (Appendices A,B, and C)

A Romberg integration routine for a vector function with nine components. It is used by the routines SHOCK, SHOCK2 and F2SHCK to compute the shock arrival time and its derivatives with respect to all arguments. (See Reference 1, pages 22-23.) The integrations are done simultaneously for all components of the integrand. Iteration end is tested on the last corrections of all components. If all relative corrections are smaller than 10^{-10}, then the iteration stops. The arguments of ROMULT are the same as those of ROMBIN.

SCALPR (Appendix B)

This routine is called from the main program for overpressure field fitting (See Figure 5). It takes from the COMMON/COMPR/ data belonging to one pressure history (specified by NRCASE) and arranges the data in the format required by the least squares program COLSAC.

SCALSH (Appendix A)

This routine is called from the main program for shock fitting. (See Figure 2.) It takes the raw shock data from COMMON/COMSHDT/and arranges them in arrays compatible with the least squares program COLSMU. It also expresses the data in scales specified in the argument list of the subroutine. Some special logic is used to handle observations with missing data. Information about such data is communicated to the constraint routine FMSHCK through the COMMON/CMISFM/.

SHOCK (Appendices B and C)

This subroutine computes for a given distance from the center of explosion the corresponding shock overpressure, arrival time, shock velocity, particle velocity and density. The formulas that are used for the computation are given in Section 4 of Reference 1. The routine is called from the main program for pressure field fitting in order to establish the initial point of a history, and also from the subroutines SCALPR, PLTLOC and UTEST.

SHOCK2 (Appendices B and C)

This routine computes for a given distance r from the explosion center the corresponding shock arrival time t_s and overpressure p_s, and the first and second order derivatives of t_s and p_s with respect to r. The corresponding formulas are given in Section 4 of Reference 1. The routine is called from the subroutine QFUNCT.

SHOCK3 (Appendix A)

This is the constraint routine for a shock overpressure model with three parameters. It computes the function

$$f = pr^3 - ar^2 - br - c$$

and its derivatives. It is used by FMSHCK to calculate the first component of the constraint function given by Equation 5.3.

SHODER (Appendices B and C)

This routine computes for a given distance r from the center of the explosion the shock arrival time t_s, the shock overpressure p_s, and all first and second derivatives of t_s and p_s with respect to r and the shock parameters. The routine uses the subroutine F2SHCK to compute t_s and its derivatives. It is called from the subroutine QFUNCT.

SHTINT (Appendices B and C)

This is the integrand in the integral given in Equation 4.2 for the calculation of the shock arrival time.

STRBEG (Appendices B and C)

This routine computes the initial values for the differential equation systems given in Equations 3.1 and 3.4 and the derivatives $\partial t_s/\partial\theta$ and $\partial \dot{u}_s/\partial\theta$ at the shock. ($\dot{u}_s$ is the particle acceleration at the shock, $\partial \dot{u}_s/\partial\theta$ is the initial value of the right hand side of the second Equation 3.4.) It also calculates an expression DPIN, which is part of the right hand side of the second Equation 3.4. The routine is called from PLOTLOC and FLOFLD to initiate the numerical integration of Equations 3.1 and 3.4. The calling program provides the shock distance r = SOLIN(3) and STRBEG uses the following formulas to calculate the other variables. (The formulas are derived in Reference 1.):

The shock overpressure is computed by Equation 5.1:

$$\text{SOLIN}(2) = p_s = a/r + b/r^2 + c/r^3.$$

The shock parameters a, b, c are taken from COMMON/COMSMK/. Let p_o be the ambient pressure, ρ_o be the ambient density, γ be the ratio of specific heats, c_o be the sound speed,

$$\Gamma_1 = (\gamma + 1)/(2\gamma p_o),$$

and

$$\Gamma_2 = (\gamma - 1)/(2\gamma p_o).$$

Then the shock velocity is

$$U = c_o \, (1 + \Gamma_1 \, p_s)^{1/2}.$$

The density behind the shock is

$$\text{SOLIN}(5) = \rho_s = \rho_o \, (1 + \Gamma_1 p_s)/(1 + \Gamma_2 p_s)$$

and the particle velocity behind the shock is

$$\text{SOLIN}(4) = u_s = p_s/(U\rho_o).$$

The shock arrival time SOLIN(1) = t_s is computed by calling the subroutine F2SHCK which evaluates the integral Equation 4.2. The acceleration $\dot{u}_s$ is given by

$$\text{UPT} = \dot{u}_s = -\frac{1}{\rho_s}\frac{\partial p_s}{\partial r}.$$

The derivatives with respect to the model parameters θ are calculated as follows

$$\text{TPIN} = \partial t_s/\partial\theta \qquad \text{provided by F2SHCK}$$

$$\text{XPP} = \partial r/\partial\theta = 0$$

$$\text{UPP} = \partial u_s/\partial\theta =$$

$$= u_s\left[1/p_s - 0.5\Gamma_1/(1 + \Gamma_1 p_s)\right]\partial p_s/\partial\theta$$

$$\rho_{s\theta} = \partial\rho_s/\partial\theta = \left[c_o^2\,(1 + \Gamma_1 p_s)\,(1 + \Gamma_2 p_s)\right]^{-1}\partial p_s/\partial\theta$$

$$= \text{ROFACT}\cdot\partial p_s/\partial\theta$$

$$\text{UPTP} = \partial\dot{u}_s/\partial\theta = \dot{u}_s\left\{-\rho_{s\theta}/\rho_s + \frac{1}{\partial p_s/\partial r}\frac{\partial^2 p_s}{\partial r\partial\theta}\right\}.$$

The derivatives of p_s with respect to r and ρ are easily computed. The above mentioned expression DPIN is defined by

$$\text{DPIN} = \rho_{s\theta}/\rho_s - \frac{1}{(p_o + p_s)\gamma}\,\partial p_s/\partial\theta.$$

STRLIN (Appendices B and C)

This routine carries out the numerical integration of the differential equation systems given in Equations 3.1 and 3.4. Initial values for the integrals are provided by the calling program which also specifies a time increment DT for which the results are needed and an end time for the integration. The actual integration increment is DTS = Ø.2 DT, but results are stored at DT-increments. The numerical integration is done using a two level fourth order scheme for Equation 3.1 and a two level third order scheme for Equation 3.4. The schemes are described in Reference 1, Section 3. The important results of the integration are the flow variables $(t,p,r,u,\rho,\rho u^2/2)$ which are stored as a six component vector in SLINA, and the corresponding variance covariance matrices at each computed node. These 6x6-matrices are stored in VSLINA. The values of $\partial r/\partial\theta$ and $\partial u/\partial\theta$, that is, the solution of Equation 3.4, are only needed to calculate the variance-covariance matrices. They are stored internally only at two current integration levels in the arrays XP and UP, together with the other quantities (u, $\dot{u}$, $\ddot{u}$ and $\dot{u}_\theta$ in U, UT, UTT and UTP) that are needed for the integration. The subroutine STRLIN is called from PLTLOC and FLOFLD (See Figure 11).

UTEST (Appendix C)

This routine computes test velocities by evaluating the integral given in Equation 2.12 (see also Figure 1). It is called from FLOFLD (Figure 11) to evaluate the integral at specified t_s-values. The corresponding shock points provide the additive term in Equation 2.12 and are obtained by calling the subroutine SHOCK. Because SHOCK computes shock values for given r, but t_s is specified, the proper r-value is found by a regula falsi iteration. The evaluation of the integral is done by calling the subroutine ROMBIN2.

UTINT (Appendix C)

This is the integrand in Equation 2.12. The routine is used by UTEST as argument when calling the ROMBIN2 quadrature to evaluate the integral.

LIST OF REFERENCES

1. Aivars Celmiņš, "Reconstruction of a Blast Field from Pressure History Observations," ARBRL-TR-02367, September 1981 (AD-A106141).

2. Ray C. Makino, "An Approximation Method in Blast Calculations," BRL-MR-1023, February 1956 (AD-114 875).

3. Richard von Mises, "Mathematical Theory of Compressible Fluid Flow," Academic Press, N.Y. 1958.

4. Aivars Celmiņš, "A Manual for General Least Squares Model Fitting," ARBRL-TR-02167, June 1979 (AD-B040229L).

Appendix A
Shock Fitting Program BLAFS

PAGE

```
      PROGRAM SHOKFIT(INPUT,OUTPUT,TAPE6=OUTPUT,TAPE13)
C
C  MAIN PROGRAM FOR SHOCK FITTING
C
      DIMENSION X(5,50),R(5,5,50),ALABEL(2,50),LSTX(50),PARS(10),
     ANXNK(2,50),XC(5,50),C(5,50),LSTN(50),VPARS(10,10),ERPARS(10),
     BPARSD(10),VPARSD(10,10),TITLE(3)
C  THESE DIMENSIONS ALLOW TO TREAT UP TO 50 SHOCK OBSERVATIONS
C  CORRESPONDING LIMITS ARE IMPLIED BY ARRAYS IN SUBROUTINE READSH
C
25    CALL READAM (SCDIS,SCPRE,SCTIM,TITLE,NBAD)
C  READ AMBIENT DATA
      IF(NBAD.NE.0)STOP
C
      CALL READSH (NRSHOK,TITLE)
C  READ ALL SHOCK OBSERVATIONS
      IF(NRSHOK.LE.0)STOP
C
      CALL SCALSH (SCDIS,SCPRE,SCTIM,X,R,ALABEL,LSTX,NXNK,NRSHOK,NBD)
      IF(NBD.NE.0)GOTO 25
C  THIS STORED SCALED OBSERVATIONS IN LSQ ARRAYS  X THROUGH NRSHOK
C
      PARS(1)=1.  $  PARS(2)=1.  $  PARS(3)=1. $ PARS(4)=0.
C  INITIAL VALUES OF SHOCK FITTING PARAMETERS
C
      DO 65 KA = 1,4
C  MAKE 4 ADJUSTMENTS: PRESSURE, PRESSURE+DISTANCE,
C     PRESSURE+DISTANCE+TIME, PRESSURE+TIME
C
      CALL FITSH(SCDIS,SCPRE,SCTIM,KA,X,R,ALABEL,LSTX,NXNK,NRSHOK,PARS,
     1 NP,XC,C,LSTN,NRGO,ERZS,VPARS,ERPARS,NBAD)
C
C  NEXT PRINT ADJUSTED OBSERVATIONS
      CALL PRSHAD(SCDIS,SCPRE,SCTIM,KA,XC,C,R,LSTN,ALABEL,NRSHOK,
     A TITLE)
C
      IF(NBAD.NE.0)GOTO 25
C
C  NEXT COMPUTE DIMENSIONAL VALUES PARSD OF THE PARAMETERS
      CALL DIMPARS(KA,SCDIS,SCPRE,SCTIM,PARS,NP,VPARS,ERZS,PARSD,VPARSD,
     A TITLE)
C
      SCDI = 1. $ SCPR = 1. $ SCTI = 1.
C  THESE SCALES CORRESPOND TO PARSD AND VPARSD
C  THEY WILL CAUSE PLOTTING IN SI BASE UNITS
      ERFACT=3.
C  ERROR FACTOR FOR PLOTTING OF CONFIDENCE LIMITS
C
      CALL PLPDSH(KA,SCDI,SCPR,SCTI,NRSHOK,PARSD,NP,VPARSD,
     AERZS,ERFACT)
C  PLOT PRESSURE OVER DISTANCE
      CALL PLPTSH(KA,SCDI,SCPR,SCTI,NRSHOK,PARSD,NP,VPARSD,
     AERZS,ERFACT)
C  PLOT PRESSURE OVER TIME
      CALL PLDTSH(KA,SCDI,SCPR,SCTI,NRSHOK,PARSD,NP,VPARSD,
     AERZS,ERFACT)
C  PLOT DISTANCE OVER TIME
```

```
   65 CONTINUE
C
60    GOTO 25
      END
```

```
      SUBROUTINE READAM(SCDIST,SCPRES,SCTIME,TITLE,NBAD)
C  THIS ROUTINE READS TITLE, PLOTLABEL AND DATA CARDS DESCRIBING
C  AMBIENT CONDITIONS AND THE CHARGE
C  FIRST TWO CARDS ARE MANDATORY AND ALPHANUMERIC  (TITLE AND PLOTLABEL)
C  THE REST OF THE CARDS HAVE THE FORMAT (2A10,6E10.3)
C  CHARGE CARD IS MANDATORY
C  IF AMBIENT DATA ARE NOT PROVIDED THEN STANDARD AIR WILL BE ASSUMED
C
C  SEQUENCE OF MANDATORY INPUT CARDS
C      TITLE CARD  (ALPHANUMERIC)
C      PLOTLABEL CARD   (ALPHANUMERIC)
C      CHARGE CARD = VOLUME, ENERGY, HIGHT, ERROR OF HIGHT
C
C  THE FOLLOWING ARE OPTIONAL INPUT CARDS IN ARBITRARY SEQUENCE
C      AMBIENT = P,TEMPERATURE, GAMMA, MOLAR MASS
C          DEFAULT VALUES CORRESPOND TO A STANDARD AIR
C      SCALES = SCALES OF R,P,T  TO BE USED IN COMPUTATIONS
C           DEFAULT VALUES ARE COMPUTED AFTER STATEMENT 1110
C      PLOTTING DATA = ERROR FACTORS FOR THE PLOTTING OF CONFIDENCE
C                      LIMITS IN HISTORY PLOTS
C           DEFAULT VALUES ARE FACTORS 2.0 FOR ALL PLOTS
C
C  END OF INPUT IS INDICATED BY A BLANK CARD
C
      DIMENSION TITLE(3)
      DIMENSION D(8),AMSTAR(4)
      COMMON/AMBCHA/AIRPR,AIRTEM,AIRGAM,AIRMOL,CHARVO,CHAREN,
     ACHARHI,CHARHER
      COMMON/PLOT/PD(6),PLABL(4)
      DATA(TITL =10HTITLE      ), (PLAB=10HPLOTLABEL )
      DATA (BLANK=10H          ),(AMB=10HAMBIENT    )
      DATA (CHA=10HCHARGE    )
      DATA(PLT=10HPLOTTING D),(SCAL=10HSCALES R,P)
   15 FORMAT(1H1,10X,20HINPUT READ BY READAM,/,1H ,10X,20(1H-),/)
   25 FORMAT(8A10)
   26 FORMAT(1H ,10X,8A10)
   35 FORMAT(2A10,6E10.3)
   36 FORMAT(1H ,10X,2A10,6(2X,1PE10.3))
C
      PD(1)=2.0
C  DEFAULT VALUE FOR PLOTTING ERROR LIMITS IN PRESSURE HISTORIES
      PD(2)=2.0
C  DEFAULT VALUE FOR PLOTTING FIELD HISTORIES (P,V,RHO,V**2*RHO/2.)
      AIRPR=101325.0  $  AIRTEM=293.0  $  AIRGAM=1.4
      AIRMOL=0.02896  $  AIRDEN=(AIRMOL/8.3143)*(AIRPR/AIRTEM)
C  THESE ARE STANDARD AIR DEFAULT VALUES FOR AMBIENT CONDITIONS
C
      NSCAL=0   $ NAMSTAR=0
      NAMB=0 $ NCHA=0
      DO 37 J=1,4
   37 AMSTAR(J)=1H
      PRINT 15
      DO 46 KK=1,2
      READ 25,(D(J),J=1,8)
      PRINT 26,(D(J),J=1,8)
      IF(D(1).EQ.TITL ) GOTO 42
      IF(D(1).EQ.PLAB) GOTO 44
```

```
          PRINT 48 $ NBAD=1 $ RETURN
C
42        DO 43 KA=1,3
43        TITLE(KA)=D(KA+1)
          GOTO 46
44        DO 45 KA=1,4
45        PLABL(KA)=D(KA+1)
46        CONTINUE
C
     47 READ 35,(D(J),J=1,8)
          PRINT 36,(D(J),J=1,8)
          IF(D(1).EQ.AMB)GOTO 55
          IF(D(1).EQ.CHA)GOTO 65
          IF(D(1).EQ.PLT) GOTO 66
          IF(D(1).EQ.SCAL) GOTO 68
          IF(D(1).EQ.BLANK) GOTO 69
    475 PRINT 48 $ NBAD=2 $ RETURN
     48 FORMAT(1H0,10X,13HINVALID INPUT)
C
     55 IF(NAMB.EQ.1)GOTO 475
C   ONLY ONE AMBIENT DATA CARD WILL BE CONSIDERED
          NAMB=1
          IF(D(3).GT.0.)AIRPR=D(3)  $  IF(D(4).GT.0.)AIRTEM=D(4)
          IF(D(5).GT.0.)AIRGAM=D(5)  $  IF(D(6).GT.0.)AIRMOL=D(6)
C   IF INPUT IS ZERO THEN USE AIR DEFAULT VALUES
          DO 57 KA=1,4  $  AMSTAR(KA)=1H
          IF(D(KA+2).GT.0.) GOTO 57
          AMSTAR(KA)=1H*  $  NAMSTAR=1
57        CONTINUE
          AIRDEN=(AIRMOL/8.3143)*(AIRPR/AIRTEM)
          GOTO 47
C
     65 IF(NCHA.EQ.1)GOTO 475
          CHARVO=D(3) $ CHAREN=D(4)
          CHARHI=D(5) $ CHARHER=D(6)
          NCHA=1
          GOTO 47
C
66        DO 67 KA=1,6
67        PD(KA)=D(KA+2)
          GOTO 47
C   PLOTTING DATA CARD SPECIFIES PLOTTED OUTPUT
C   PD(1)= ERROR FACTOR FOR PRESSURE HISTORIES
C   PD(2)= ERROR FACTOR FOR OTHER FLOW HISTORIES
C
68        NSCAL=1
          SCD=D(3)  $  SCP=D(4)  $  SCT=D(5)
C   SCALE CARD OVERRIDES SCALES COMPUTED FROM AMBIENT AND CHARGE DATA
          IF(SCD.GT.0..AND.SCP.GT.0..AND.SCT.GT.0.) GOTO 47
          NSCAL=0  $  PRINT 681
681        FORMAT(1H ,10X,36HNON-POSITIVE SCALES ARE NOT ACCEPTED)
          GOTO 47
C
69        IF(NCHA.EQ.0.OR.NAMB.EQ.0) PRINT 70
70        FORMAT(1H0,10X,16HINCOMPLETE INPUT)
     75 PRINT106,(TITLE(J),J=1,3)
106       FORMAT(1H1,/,1H ,10X,5HEVENT,/,1H ,10X, 5(1H-),/,1H0,15X,3A10,//)
```

```
      PRINT 107
  107 FORMAT(1H0,10X,18HAMBIENT CONDITIONS,/,1H ,10X,18(1H
      IF(NAMB.EQ.0) PRINT 1071
 1071 FORMAT(1H0,10X,36HTHE FOLLOWING AMBIENT CONDITIONS ARE,
     A /,1H ,10X,27HSTANDARD AIR DEFAULT VALUES,/)
      PRINT 108,AMSTAR(1),AIRPR,AMSTAR(2),AIRTEM,AMSTAR(3),AIRGAM,
     A AMSTAR(4),AIRMOL
  108 FORMAT(1H ,13X,A1,1X,8HPRESSURE,11X,7HAIRPR=,1PE12.5,4H  PA,/,
     A 1H ,13X,A1,1X,11HTEMPERATURE,8X,7HAIRTEM=,1PE12.5,3H  K,/,
     B 1H ,13X,A1,1X,16HSPEC. HEAT RATIO,3X,7HAIRGAM=,1PE12.5,/,
     C 1H ,13X,A1,1X,10HMOLAR MASS,9X,7HAIRMOL=,1PE12.5,9H  KG/MOLE,/)
      AIRSND=SQRT(AIRGAM*AIRPR/AIRDEN)
      PRINT 109,AIRSND,AIRDEN
  109 FORMAT(1H ,15X,11HSOUND SPEED,8X,7HAIRSND=,1PE12.5,5H  M/S,/,
     A 1H ,15X,7HDENSITY,12X,7HAIRDEN=,1PE12.5,9H  KG/M**3,/)
      IF(NAMSTAR.EQ.1) PRINT 1081
 1081 FORMAT(1H ,13X,35H* THE STARRED DATA ARE STANDARD AIR,
     A 15H DEFAULT VALUES,/)

      IF(NCHA.EQ.1) GOTO 1100
      NBAD=4  $  PRINT 1101,NBAD  $  RETURN
 1101 FORMAT(1H0,10X,29HRETURN FROM READAM WITH NBAD=,I2,
     A 33H, BECAUSE CHARGE DATA ARE MISSING)
C
 1100 PRINT 110
  110 FORMAT(1H0,10X,18HCHARGE DESCRIPTION,/,1H ,10X,18(1H-),/)
      PRINT 111, CHARVO,CHAREN
  111 FORMAT(1H ,15X,13HCHARGE VOLUME,6X,7HCHARVO=,1PE12.5,6H  M**3,/,
     A 1H ,15X,13HCHARGE ENERGY,6X,7HCHAREN=,1PE12.5,3H  J,/)
      SCDIST=CHARVO**(1./3.)
      PRINT 1110,CHARHI,CHARHER
 1110 FORMAT(1H ,15X,16HCHARGE ELEVATION,3X,7HCHARHI=,1PE12.5,4H +- ,
     A 1PE12.5,3H  M,/)
      SCTIME=SCDIST/AIRSND
      SCPRES=AIRPR
      SCEVEN=CHAREN/(CHARVO*AIRPR)
      PRINT 112
  112 FORMAT(1H0,10X,7HSCALING,/,1H ,10X,7(1H-),/)
      PRINT 113,SCDIST,SCTIME,SCPRES,SCEVEN
  113 FORMAT(1H ,15X,12HLENGTH SCALE,4X,20HSCDIST=CHARVO**(1/3),
     A 2X,1H=,1PE12.5,3H  M,/,
     B 1H ,15X,10HTIME SCALE,6X,20HSCTIME=SCDIST/AIRSND,
     C 2X,1H=,1PE12.5,3H  S,/,
     D 1H ,15X,14HPRESSURE SCALE,2X,13HSCPRES=AIRPR ,
     E 9X,1H=,1PE12.5,4H  PA,/,
     F 1H ,15X,14HSCALE OF EVENT,2X,21HCHAREN/(CHARVO*AIRPR),
     G 1X,1H=,1PE12.5,/)
      IF(SCEVEN.EQ.0.0)PRINT 114
  114 FORMAT(1H ,15X,30HEVENT CANNOT BE SCALED BECAUSE,
     A29H CHAREN IS NOT GIVEN BY INPUT,/)

      IF(NSCAL.EQ.0) GOTO 115
C  USE SCALES FROM SCALE CARD IF SUCH A CARD WAS READ
      SCDIST=SCD  $  SCPRES=SCP  $  SCTIME=SCT

  115 PRINT 116,SCDIST,SCTIME,SCPRES
  116 FORMAT(1H ,////,1H ,10X,27HSCALES USED IN THIS PROGRAM,/,
```

```
      A 1H ,10X,27(1H-),//,1H ,20X,16HLENGTH SCALE    =,1PE12.5,3H  M,/,
      B 1H ,20X,16HTIME SCALE      =,1PE12.5,3H  S,/,
      C 1H ,20X,16HPRESSURE SCALE =,1PE12.5,4H  PA)
175    NBAD=0
       RETURN
       END
```

```
      SUBROUTINE READSH(NRSH,TIT)
C  THIS READS SHOCK DATA
C
C  ALL CARDS HAVE THE FORMAT  (2A10,6(E10.3))
C  SHOCK CARD CONTAINS  LABEL,TIME,ERROR OF T, PRESSURE, ERROR OF P
C  RANGE CARD CONTAINS  LABEL, X, ERROR OF X, HIGHT, ERROR OF H
C  THE SEQUENCE OF THE INPUT CARDS IS ARBITRARY
C
C  END OF INPUT IS INDICATED BY A BLANK CARD
C
      COMMON/COMSHOT/TPXH(4,50),ERTPXH(4,50),TITLE(3),ALAB(2,50)
C  T,P,X,H OF SHOCK OBSERVATIONS.  CORRESPONDING ERRRORS
      DIMENSION TIT(3),D(6)
      DATA(NMAX=50)
C  MAXIMUM NUMBER OF SHOCK DATA THAT CAN BE STORED
C
      DATA (RANGE=10HRANGE     ),(SHOCK=10HSHOCK     )
     A,(BLANK=10H          )
C
      DO 10 J=1,3
10    TITLE(J)=TIT(J)
      NRSH=0
      DO 12 J=1,50  $  ALAB(1,J)=BLANK
      DO 11 K=1,4  $  ERTPXH(K,J)=0.
11    TPXH(K,J)=0.
12    ALAB(2,J)=BLANK
15    FORMAT(2A10,6(E10.3))
16    FORMAT(1H ,5X,2A10,6(2X,1PE12.5))
      PRINT 18
18    FORMAT(1H1,10X,20HINPUT READ BY READSH,//)
27    CONTINUE
      READ15,(D(J),J=1,6)
      PRINT 16,(D(J),J=1,6)
      IF(D(1).EQ.BLANK) GOTO 75
      IF(D(2).EQ.RANGE) GOTO 35
      IF(D(2).EQ.SHOCK) GOTO 55
      PRINT 28
      STOP
28    FORMAT(1H ,10X,13HINVALID INPUT)
C
35    DO 37 KA=1,NMAX
      IF(KA.GT.NRSH) GOTO 40
      IF(D(1).EQ.ALAB(1,KA)) GOTO 42
37    CONTINUE
      GOTO 85
40    NRSH=NRSH+1  $  KA=NRSH
      ALAB(1,KA)=D(1)  $  ALAB(2,KA)=TIT(1)
42    TPXH(3,KA)=D(3)  $  ERTPXH(3,KA)=D(4)
      TPXH(4,KA)=D(5)  $  ERTPXH(4,KA)=D(6)
      GOTO 27
C
55    DO 57 KA=1,NMAX
      IF(KA.GT.NRSH) GOTO 60
      IF(D(1).EQ.ALAB(1,KA)) GOTO 62
57    CONTINUE
      GOTO 85
60    NRSH=NRSH+1  $  KA=NRSH
```

```
      ALAB(1,KA)=D(1)  $  ALAB(2,KA)=TIT(1)
62    TPXH(1,KA)=D(3)  $  ERTPXH(1,KA)=D(4)
      TPXH(2,KA)=D(5)  $  ERTPXH(2,KA)=D(6)
      GOTO 27
C
75    IF(NRSH.LE.0) STOP
85    DO 105 KA=1,NRSH
      IF(MOD(KA,45).NE.1) GOTO 101
      PRINT 95,(TIT(J),J=1,3)
95    FORMAT(1H1,10X,22HSHOCK DATA FROM EVENT ,3A10,/)
      PRINT 98
   98 FORMAT(1H0,4H NR.,11X,6HLABELS,12X,4HTIME,6X,9HSTD.ERROR,4X,
     A 12HOVERPRESSURE,2X,9HSTD.ERROR,7X,5HRANGE,5X,
     B 9HSTD.ERROR,6X,9HELEVATION,3X,9HSTD.ERROR,/,
     C 1H ,33X,3H(S),10X,3H(S),11X,4H(PA),9X,4H(PA),10X,3H(M),
     D 9X,3H(M),11X,3H(M),10X,3H(M),/)

  101 CONTINUE
      PRINT 102,KA,ALAB(1,KA),ALAB(2,KA),(TPXH(J,KA),ERTPXH(J,KA),J=1,2)
  102 FORMAT(1H ,I4,1X,2A10,4(4X,1PE12.5,2X,1PE9.2))
      IF((KA/5)*5.EQ.KA) PRINT 103
103    FORMAT(1H )
105    CONTINUE
      RETURN  $  END
```

```
      SUBROUTINE SCALSH(SCDI,SCPR,SCTI,X,R,ALAB,LSTX,NXNK,
     ANRSHOK,NBAD)
C  THIS STORES PROPERLY SCALED SHOCK DATA IN LSQ ARRAYS
C  THE SCALES ARE PROVIDED BY THE CALLING PROGRAM
C  X(1)=PRESSURE, X(2)=DISTANCE, X(3)=TIME
C  IF PRESSURE DATA ARE MISSING THEN X(1)=TIME
C
      DIMENSION X(5,50),R(5,5,50),ALAB(2,50),LSTX(50),NXNK(2,50)
C
      COMMON/AMBCHA/AMPR,AMTEM,GAM,AMBMOL,CHVOL,CHEN,CHH,ECHH
      COMMON/COMSHOT/TPXH(4,50),ERTPXH(4,50),TITLE(3),ALB(2,50)
C  THIS CONTAINS RAW INPUT
C
      COMMON/CMISFM/MISPDT(3,50),DISTN(50),NODIST,SCDD
C  THIS INDICATES FOR SUBROUTINE FMSHCK MISSING P,D OR T BY 1 IN MISPDT
C  NODIST.NE.0 INDICATES THAT ERROR FREE DISTANCES ARE IN DISTN
C
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),DMINSC,SCD,SCP,SCT
C  /CF2DER/ IS USED BY CONSTRAINT ROUTINES F2SHCK AND F2DER
C
      COMMON/CMPLSH/PMIN,PMAX,DMIN,DMAX,TMIN,TMAX
C  THE EXTREME VALUES IN CMPLSH WILL DETERMINE PLOTTING LIMITS
C
      GAMCAP=((1.+GAM)/(2.*GAM))*(SCPR/AMPR)
      SNDSPD=SQRT(GAM*AMTEM*8.31431/AMBMOL)*(SCTI/SCDI)
      SCD=SCDI  $  SCP=SCPR  $  SCT=SCTI
C  THIS TELLS IN WHAT UNITS GAMCAP AND SNDSPD ARE EXPRESSED
C
      PMIN=0 $ PMAX=0 $ DMIN=0 $ DMAX=0 $ TMIN=0 $ TMAX=0
      NRS=0
C
      SCDD=SCDI
      DO 55 KA=1,NRSHOK
      IF(TPXH(3,KA).GT.0..AND.ERTPXH(3,KA).GT.0.)GOTO 15
      MISPDT(2,KA)=1  $  LSTX(KA)=1
      MISPDT(1,KA)=0
      IF(TPXH(2,KA).LE.0..OR.ERTPXH(2,KA).LE.0.) MISPDT(1,KA)=1
      MISPDT(3,KA)=0
      IF(TPXH(1,KA).LE.0..OR.ERTPXH(1,KA).LE.0.) MISPDT(3,KA)=1
      GOTO 45
   15 X(2,KA)=SQRT(TPXH(3,KA)**2+(CHH-TPXH(4,KA))**2)
      R(2,2,KA)=(TPXH(3,KA)*ERTPXH(3,KA)/X(2,KA))**2+
     A((CHH-TPXH(4,KA))/X(2,KA))**2*(ECHH**2+ERTPXH(4,KA)**2)
C
      IF(DMIN.GT.0.)GOTO 16
      DMIN=X(2,KA)  $  DMAX=DMIN
   16 DMIN=AMIN1(DMIN,X(2,KA))   $ DMAX=AMAX1(DMAX,X(2,KA))
C
      X(2,KA)=X(2,KA)/SCDI
      R(2,2,KA)=R(2,2,KA)/SCDI**2
      DISTN(KA)=X(2,KA)
      R(1,3,KA)=0 $ R(3,1,KA)=0 $ R(2,3,KA)=0 $ R(3,2,KA)=0
      R(1,2,KA)=0 $ R(2,1,KA)=0 $ LSTX(KA)=0 $ MISPDT(2,KA)=0
      J=1 $ MISPDT(1,KA)=1
      IF(TPXH(2,KA).LE.0..OR.ERTPXH(2,KA).LE.0.)GOTO 25
C
      J=3 $ MISPDT(1,KA)=0
```

```
      X(1,KA)=TPXH(2,KA)/SCPR
      R(1,1,KA)=(ERTPXH(2,KA)/SCPR)**2
C
      IF(PMIN.GT.0.)GOTO 22
      PMIN=TPXH(2,KA) $ PMAX=PMIN $ GOTO 25
   22 PMIN=AMIN1(PMIN,TPXH(2,KA)) $ PMAX=AMAX1(PMAX,TPXH(2,KA))
C
   25 IF(TPXH(1,KA).GT.0..AND.ERTPXH(1,KA).GT.0.)GOTO 35
      MISPDT(3,KA)=1 $ IF(MISPDT(1,KA).NE.0)LSTX(KA)=1 $ GOTO 45
   35 X(J,KA)=TPXH(1,KA)/SCTI
      R(J,J,KA)=(ERTPXH(1,KA)/SCTI)**2
      MISPDT(3,KA)=0
C
      IF(TMAX.GT.0.)GOTO 38
      TMIN=TPXH(1,KA) $ TMAX=TMIN $ GOTO 45
   38 TMIN=AMIN1(TMIN,TPXH(1,KA)) $ TMAX=AMAX1(TMAX,TPXH(1,KA))
C
   45 ALAB(1,KA)=ALB(1,KA) $ ALAB(2,KA)=ALB(2,KA)
      IF(LSTX(KA).EQ.0)NRS=NRS+1
   55 CONTINUE
C
      DMINSC=DMIN/SCDI
      NBAD=0 $ IF(NRS.EQ.0)NBAD=1
      RETURN
      END
```

```
      SUBROUTINE FITSH(SCD,SCP,SCT,KA,X,R,ALABEL,LSTX,NXNK,NRSCK,PAR,NP,
     1 XC,C,LSTN,NRGD,ERZ,VPAR,ERPAR,NBAD)
C  THIS FITS SHOCK DATA ACCORDING TO MODIFIER KA
C
C  ROUTINE USES LSQ PROGRAM COLSMUA FOR FITTING
C
C  SCD, SCP, SCT    =  SCALES IN TERMS OF WHICH THE ARGUMENTS X IS EXPRESSED
C  KA        =    MODIFIER FOR FITTING
C  KA=1 - FIT PRESSURE. KA=2 - FIT PRESSURE+DISTANCE
C  KA=3 - FIT PRESSURE+DISTANCE+TIME. KA=4 - FIT PRESSURE+TIME
C
C  X(5,50)      =  LEAST SQUARES DATA ARRAY
C  X(1)=PRESSURE, X(2)=DISTANCE, X(3)=TIME
C  IF PRESSURE DATA ARE MISSING THEN X(1)=TIME
C
C  THE REMAINING ARGUMENTS ARE STANDARD LEAST SQUARES ARGUMENTS
C
      DIMENSION X(5,50),R(5,5,50),ALABEL(2,50),LSTX(50),NXNK(2,50),
     APAR(10),XC(5,50),C(5,50),LSTN(50),VPAR(10,10),ERPAR(10)
C
      DIMENSION XA(5,50),RA(5,5,50),XCA(5,50),CA(5,50)
      DIMENSION WORK(4000)
C
      COMMON/CMISFM/MISPDT(3,50),DISTN(50),NODIST,SCDD
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCDI,SCPR,SCTI
C  MISPDT INDICATES MISSING P,D OR T BY CORRESPONDING ONES
C  NODIST.NE.0 INDICATES THAT ERROR FREE DISTANCES ARE IN DISTN
C  BOTH COMMON BLOCKS ARE NEEEDED BY CONSTRAINT ROUTINES
C
      EXTERNAL FMSHCK
C
      DATA(NWORK=4000)
C  MAXIMUM DIMENSION OF WORK, NEEDED BY COLSMUA
C
      SNDSPD=SNDSPD*SCDI*SCT/(SCTI*SCD)
      GAMCAP=GAMCAP*SCP/SCPR
      ALOW=ALOW*SCDI/SCD
      SCDI=SCD
      SCTI=SCT
      SCPR=SCP
      S1=SCDD/SCD
      DO 10 I=1,50
      DISTN(I)=DISTN(I)*S1
   10 CONTINUE
      SCDD=SCD
C  NOW ALL COMMON BLOCK DATA ARE EXPRESSED IN SCALES GIVEN BY THE ARGUMENT
C
      NX=MINO(KA,3) $ NODIST=0 $ ITYPE=0
      NP=MAXO(3,KA+1) $ NP=MINO(NP,4)
C
      DO 45 KB=1,NRSCK $ IF(LSTX(KB).EQ.1)GOTO 45
C
      DO 25 KC=1,3 $ XA(KC,KB)=X(KC,KB)
      XCA(KC,KB)=X(KC,KB) $ DO 25 KD=1,3
   25 RA(KC,KD,KB)=R(KC,KD,KB)
C
      NXNK(1,KB)=NX $ LSTX(KB)=0
```

```
      NXNK(2,KB)=MAXO(1,KA-1) $ IF(NXNK(2,KB).GT.2)NXNK(2,KB)=2
      IF(KA.EQ.1.AND.MISPDT(1,KB).NE.0)LSTX(KB)=2
      IF(KA.EQ.2.AND.MISPDT(1,KB).NE.0)LSTX(KB)=3
      IF(KA.LE.2)GOTO 45
      IF(KA.EQ.3)GOTO 35
C
      NODIST=1 $ NXNK(1,KB)=2 $ NX=2
      IF(MISPDT(1,KB).NE.0.OR.MISPDT(3,KB).NE.0)NXNK(1,KB)=1
      NXNK(2,KB)=NXNK(1,KB)
      IF(MISPDT(1,KB).NE.0)GOTO 45
      XA(2,KB)=X(3,KB) $ RA(2,2,KB)=R(3,3,KB)
      GOTO 45
C
   35 IF(MISPDT(1,KB).EQ.0.AND.MISPDT(3,KB).EQ.0)GOTO 45
      NXNK(1,KB)=2 $ NXNK(2,KB)=1
   45 CONTINUE
C
      IF(KA.EQ.3) ITYPE=4
      NXD=5 $ NPD=10 $ NKD=3
      CALL COLSMUA(XA,RA,ALABEL,LSTX,NXNK,NRSCK,PAR,NP,FMSHCK,ITYPE,
     AXCA,CA,LSTN,NRGD,ERZ,VPAR,ERPAR,NBAD,NXD,NKD,NPD,WORK,NWORK)
      IF(NBAD.EQ.0) GOTO 50
C
      PRINT 46,(PAR(J),J=1,NP)
46    FORMAT(1H0,10X,4HPAR=,4(3X,1PE12.5))
      PRINT 47,(LSTN(J),J=1,NRSCK)
47    FORMAT(1H ,10X,5HLSTN=,10(3X,I7))
C
50    CONTINUE
C
      DO 65 KB=1,NRSCK $ IF(LSTN(KB).NE.0)GOTO 65
      DO 55 KC=1,3 $ XC(KC,KB)=XCA(KC,KB)
   55 C(KC,KB)=CA(KC,KB)
      IF(KA.LE.3)GOTO 65
      IF(MISPDT(1,KB).NE.0)GOTO 65
      XC(2,KB)=X(2,KB) $ C(2,KB)=0
      XC(3,KB)=XCA(2,KB) $ C(3,KB)=CA(2,KB)
   65 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FMSHCK(XX,CK,NXNK,KA,PAR,F,FX,FP,FXX,FXP,FPP,NB)
C  MULTIPLE CONSTRAINT FOR SHOCK FITTING
C  ARGUMENTS ARE DESCRIBED IN COLSMU MANUAL
C
      DIMENSION XX(5,100),CK(3,100),NXNK(2,100),PAR(10),F(3),
     A FX(3,5),FP(3,10),FXX(5,5),FXP(5,10),FPP(10,10)
      DIMENSION DFX(5),DFP(10),DFXX(5,5),DFXP(5,10),DFPP(10.10),DX(5,1)
      COMMON/CMISFM/MISPDT(3,50),DIST(50),NODIST,SCD
C  MISPDT INDICATES BY 1 IF P,D OR T IS MISSING
C  DIST ARE DISTANCES OBSERVED. IF NODIST.NE.0 THEN DIST ARE ERROR FREE
C  SCD IS THE SCALE USED FOR DIST
C
      NB=0
      DO 4 KB=1,2  $  F(KB)=0  $  DO 4 KC=1,4  $  FX(KB,KC)=0
4     FP(KB,KC)=0
      DO 5 KB=1,4  $  DO 5 KC=1,4  $  FXX(KB,KC)=0 $ FXP(KB,KC)=0
5     FPP(KB,KC)=0
      IF(MISPDT(2,KA).NE.0) GOTO 6
C  BRANCH TO ERROR RETURN IF DISTANCE IS MISSING
C
      DX(1,1)=XX(1,KA) $ DX(2,1)=XX(2,KA) $ DX(3,1)=XX(3,KA) $ M=3
      IF(NODIST.NE.0) GOTO 7
      IF(MISPDT(1,KA).EQ.0) GOTO 10  $  IF(MISPDT(3,KA).EQ.0) GOTO 8
C
6     NB  =99  $  RETURN
C
7     DX(2,1)=DIST(KA)  $  M=1  $  IF(MISPDT(1,KA).EQ.0) GOTO 9
8     DX(3,1)=XX(1,KA)  $  J=1  $  GOTO 60

9     DX(3,1)=XX(2,KA)
C  ENTER 9 AND COMPUTE FIRST COMPONENT OF CONSTRAINT FUNCTION
10    CALL SHOCK3(DX, 1,PAR,F(1),DFX,DFP,DFXX,DFXP,DFPP,NBAD)
      IF(NBAD.EQ.0) GOTO 15 $ NB=NBAD+100 $ RETURN
   15 DO 45 KB=1,M $ FX(1,KB)=DFX(KB) $ DO 25 KC=1,M
   25 FXX(KB,KC)=CK(1,KA)*DFXX(KB,KC)
      DO 35 KC=1,4
   35 FXP(KB,KC)=CK(1,KA)*DFXP(KB,KC)
   45 CONTINUE
      DO 55 KB=1,4 $ FP(1,KB)=DFP(KB) $ DO 55 KC=1,4
   55 FPP(KB,KC)=CK(1,KA)*DFPP(KB,KC)
C
      IF(NXNK(2,KA).LT.2) RETURN          $  J=2
C
60    CALL F2SHCK(DX, 1,PAR,F(J),DFX,DFP,DFXX,DFXP,DFPP,NBAD)
C  THIS IS THE SECOND CONSTRAINT COMPONENT.  ENTER 60 FROM 8 IF
C  ONLY THE SECOND CONSTRAINT COMPONENT IS USED.
      IF(NBAD.EQ.0) GOTO 65 $ NB=NBAD+200 $ RETURN
65    L=NXNK(1,KA)
      DO 95 KB=1,L  $  KJ=KB+(2-J)*(4-2*KB)
      IF(J*M.EQ.2.AND.KB.EQ.2)KJ=3  $  FX(J,KB)=DFX(KJ)
      DO75 KC=1,L$ KK=KC+(2-J)*(4-2*KC) $ IF(J*M.EQ.2.AND.KC.EQ.2)KK=3
   75 FXX(KB,KC)=FXX(KB,KC)+CK(J,KA)*DFXX(KJ,KK)
      DO 85 KC=1,4
   85 FXP(KB,KC)=FXP(KB,KC)+CK(J,KA)*DFXP(KJ,KC)
   95 CONTINUE
      DO 105 KB=1,4 $ FP(J,KB)=DFP(KB) $ DO 105 KC=1,4
  105 FPP(KB,KC)=FPP(KB,KC)+CK(J,KA)*DFPP(KB,KC)
```

```
RETURN
END
```

```
      SUBROUTINE SHOCK3(XX,KA,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  SHOCK FITTING CONSTRAINT WITH 3 PARAMETERS
C  THIS IS USED BY FMSHCK AS THE FIRST CONSTRAINT COMPONENT
C
      DIMENSION XX(5,100),PAR(10),FX(5),FP(10),FXX(5,5),FXP(5,10),
     1FPP(10,10)
C
      NBAD=0 $ X=XX(2,KA)
      FX(1)=X*X*X
      F=((XX(1,KA)*X-PAR(1))*X-PAR(2))*X-PAR(3)
      FX(2)=(3.*XX(1,KA)*X-2.*PAR(1))*X-PAR(2)
      FX(3)=0
      FP(1)=-X*X $ FP(2)=-X $ FP(3)=-1. $ FP(4)=0
      FXX(1,1)=0. $ FXX(1,2)=3.*X*X $ FXX(2,1)=FXX(1,2)
      FXX(2,2)=6.*XX(1,KA)*X-2.*PAR(1)
      DO 15 KB=1,3 $ FXX(3,KB)=0. $ FXX(KB,3)=0 $ DO 15 KC=1,4
   15 FXP(KB,KC)=0
      DO 25 KB=1,4 $ DO 25 KC=1,4
   25 FPP(KB,KC)=0
      FXP(2,1)=-2.*X $ FXP(2,2)=-1.
      RETURN
      END
```

```
      SUBROUTINE F2SHCK(XX,KA,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS IS SECOND CONSTRAINT COMPONENT FOR SHOCK FITTING,
C  CALLED FROM FMSHCK.
C
      DIMENSION XX(5,100),PAR(10),FX(5),FP(10),FXX(5,5),FXP(5,10),
     A FPP(10,10),SF(9)
      EXTERNAL F2DER
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCD,SCP,SCT
C  GAMCAP=((1.+GAM)/(2.*GAM))*(SCPR/AMBPR)
C  GAMCAP, SNDSPD AND ALOW ARE SET BY SUBROUTINE SCALSH
C
      DO 15 KB=1,4
   15 CPAR(KB)=PAR(KB)
C  THE PARAMETERS CPAR WILL BE USED BY SUBROUTINE F2DER
      X=XX(2,KA)
      DO 25 KB=1,3 $ DO 25 KC=1,3
   25 FXX(KB,KC)=0
      IF(X.GT.1.E-30) GOTO 35 $ NBAD=1 $ RETURN
   35 NBAD=0
      SQ=1.+GAMCAP*((PAR(3)/X+PAR(2))/X+PAR(1))/X
      IF(SQ.GT.1.E-50) GOTO 45 $ NBAD=2 $ RETURN
   45 FX(1)=0.  $  FX(2)=1./SQRT(SQ)  $  FX(3)=-SNDSPD
      FXX(2,2)=0.5*GAMCAP*FX(2)*((3.*PAR(3)/X+2.*PAR(2))/X
     A+PAR(1))/(X*X*SQ)
      CALL ROMULT(F2DER,ALOW,X,SF,NBAD)
      IF(NBAD.EQ.0) GOTO 55 $ NBAD=NBAD+10 $ RETURN
   55 F=SF(1)+(PAR(4)-XX(3,KA))*SNDSPD
      FP(1)=SF(2)  $  FP(2)=SF(3)  $  FP(3)=SF(4)  $  FP(4)=SNDSPD
      FPP(1,1)=SF(5) $ FPP(1,2)=SF(6) $ FPP(1,3)=SF(7)
      FPP(2,1)=SF(6) $ FPP(2,2)=SF(7) $ FPP(2,3)=SF(8)
      FPP(3,1)=SF(7) $ FPP(3,2)=SF(8) $ FPP(3,3)=SF(9)
      DO 65 KB=1,4 $ FPP(4,KB)=0 $ FPP(KB,4)=0 $ FXP(1,KB)=0
   65 FXP(3,KB)=0
      FXP(2,1)=-0.5*GAMCAP*FX(2)/(X*SQ)
      FXP(2,2)=FXP(2,1)/X $ FXP(2,3)=FXP(2,2)/X $ FXP(2,4)=0
      RETURN
      END
```

```
      SUBROUTINE F2DER(X,F,NBAD)
C  INTEGRAND FOR NINE COMPONENTS OF F2 AND DERIVATIVES
      DIMENSION F(9)
      COMMON/CF2DER/GAMCAP,SNDSPD, PAR(4),ALOW,SCD,SCP,SCT
C  GAMCAP=((1.+GAM)/(2.*GAM))*(SCP /AMBPR)
C  GAMCAP, SNDSPD, ALOW AND SCALES ARE SET BY SUBROUTINE SCALSH
      NBAD=0 $ IF(X.GT.1.E-30) GOTO 15 $ NBAD=1 $ RETURN
   15 Y=1./X
      SQ=1.+GAMCAP*((PAR(3)*Y+PAR(2))*Y+PAR(1))*Y
      IF(SQ.GT.1.E-50) GOTO 25 $ NBAD=2 $ RETURN
C  INTEGRANDS CORRESPOND TO FOLLOWING QUANTITIES
C  F,FP(1),(2),(3),FPP(1,1),(1,2),(1,3)=(2,2),(2,3),(3,3)
   25 F(1)=1./SQRT(SQ)
      F(2)=-0.5*GAMCAP*F(1)*Y/SQ
      F(3)=F(2)*Y $ F(4)=F(3)*Y
      F(5)=-1.5*GAMCAP*F(3)/SQ
      F(6)=F(5)*Y $ F(7)=F(6)*Y $ F(8)=F(7)*Y $ F(9)=F(8)*Y
      RETURN
      END
```

```
      SUBROUTINE ROMULT(F,A,B,SF,NBAD)
C  ROMBERG INTEGRATION OF A 9-DIMENSIONAL VECTOR FUNCTION
C
      DIMENSION SF(9),T(9,10,20),FA(9),FB(9),FN(9),FM(9),CORKM(9,10)
C
      NBAD=0
      CALL F(A,FA,NBAD) $ IF(NBAD.NE.0) RETURN
      CALL F(B,FB,NBAD) $ IF(NBAD.NE.0) RETURN
      DO 14 KD=1,9
   14 T(KD,1,1)=(FA(KD)+FB(KD))*0.5
      KM=1 $ KMA=1
   15 DO 16 KD=1,9
   16 FM(KD)=0
      DEN=FLOAT(KMA)*2.
      DO 25 KA=1,KMA
      AC=FLOAT(1+2*(KMA-KA))/DEN $ BC=FLOAT(2*KA-1)/DEN
      ARG=AC*A+BC*B
      CALL F(ARG,FN,NBAD) $ IF(NBAD.NE.0) RETURN
      DO 23 KD=1,9
   23 FM(KD)=FM(KD)+FN(KD)
   25 CONTINUE
      DO 26 KD=1,9 $ FM(KD)=FM(KD)/FLOAT(KMA)
   26 T(KD,1,KM+1)=(T(KD,1,KM)+FM(KD))*0.5
C
C  THIS IS TRAPEZ. NEXT COMPUTE ROMBERG
      KM=KM+1 $ KC=1 $ DDEN=1.
   35 KC=KC+1 $ DDEN=DDEN*4.
      DO 37 L=1,9
      CORKM(L,KC)=(T(L,KC-1,KM)-T(L,KC-1,KM-1))/(DDEN-1.)
37    T(L,KC,KM)=T(L,KC-1,KM)+CORKM(L,KC)
      IF(KC.LT.KM.AND.KC.LT.10) GOTO 35
C
C  NEXT TEST CONVERGENCE
      IF(KM.GE.3) GOTO 45 $ KMA=KMA*2 $ GOTO 15
   45 IF(KM.GE.20) GOTO 56
      DO 53 L=1,9
      TEST=ABS(CORKM(L,KC))
C  KC=MIN(KM,10)
      IF(TEST.LE.1.E-100) GOTO 53
      IF(TEST.LE.ABS(T(L,KC,KM))*1.E-10) GOTO 53
      KMA=KMA*2 $ GOTO 15
   53 CONTINUE
C
   56 DO 58 L=1,9
   58 SF(L)=T(L,KC,KM)*(B-A)
      RETURN
      END
```

```
      SUBROUTINE PRSHAD(SCDIS,SCPRE,SCTIM,KK,XC,C,R,LSTN,ALAB,
     A NRSHOK,TITLE)
C  THIS PRINTS ADJUSTED SHOCK DATA
C  ROUTINE SHOULD BE CALLED AFTER RETURN FROM FITSH
C
      DIMENSION XC(5,50),R(5,5,50),C(5,50),ALAB(2,50)
      DIMENSION TITLE(3),LSTN(50)
      COMMON/CMISFM/MISPDT(3,50),DISTN(50),NODIST,SCDD
C
      TB=1H
      PB=1H
      K=0
      DO 100 I=1,NRSHOK
      IF(LSTN(I).NE.0) GO TO 100
      K=K+1
      IF(MOD(K,40).NE.1) GOTO 18
      PRINT 2,TITLE
    2 FORMAT(1H1,45X,3A10)
      PRINT 5
    5 FORMAT(1H ,45X,*ADJUSTED SHOCK OBSERVATIONS*,//)
      PRINT 10
   10 FORMAT(1H ,4H NR.,8X,6HLABELS,12X,6HTIME +,25X,12HOVERPRESSURE,
     A 23X,10HDISTANCE +,/,
     B1H ,28X,10HCORRECTION,2X,8HCORRECT.,2X,9HSTD.ERROR,2X,
     C11H+CORRECTION,2X,8HCORRECT.,2X,9HSTD.ERROR,3X,
     D10HCORRECTION,2X,8HCORRECT.,2X,9HSTD.ERROR,/)
      IF(SCTIM.EQ.1.)PRINT 11
   11 FORMAT(1H+,31X,3H(S),2(8X,3H(S)))
      IF(SCTIM.NE.1.)PRINT 1101
 1101 FORMAT(1H+,29X,8H(SCTIME),1X,2(2X,8H(SCTIME)))
      IF(SCPRE.EQ.1.)PRINT 12
   12 FORMAT(1H+,64X,4H(PA),2(7X,4H(PA)))
      IF(SCPRE.NE.1.)PRINT 1201
 1201 FORMAT(1H+,63X,8H(SCPRES),2(3X,8H(SCPRES)))
      IF(SCDIS.EQ.1.)PRINT 13
   13 FORMAT(1H+,99X,3H(M),2(8X,3H(M)))
      IF(SCDIS.NE.1.)PRINT 1301
 1301 FORMAT(1H+,97X,8H(SCDIST),1X,2(2X,8H(SCDIST)))
      PRINT 15
   15 FORMAT(1H )
   18 CONTINUE
      IF(KK.EQ.1) GO TO 30
      IF(KK.EQ.2) GO TO 40
      IF(KK.EQ.3) GO TO 50
      IF(KK.EQ.4) GO TO 60

   30 R1=SQRT(R(1,1,I))
      C(2,I)=0.0
      R2=0.0
      PRINT 21,I,ALAB(1,I),ALAB(2,I),TB,TB,TB,XC(1,I),C(1,I),R1,XC(2,I),
     1 C(2,I),R2
   21 FORMAT(1H ,I4,1X,2A10,3X,A10,1X,A9,1X,A10,2(3X,1PE10.3,1X,
     A 1PE9.2,1X,1PE10.3))
      GO TO 90

   40 R1=SQRT(R(1,1,I))
      R2=SQRT(R(2,2,I))
```

```
      PRINT 21,I,ALAB(1,I),ALAB(2,I),TB,TB,TB,XC(1,I),C(1,I),R1,XC(2,I),
     1 C(2,I),R2
      GO TO 90

   50 IF(MISPDT(1,I).EQ.0.0.AND.MISPDT(3,I).EQ.0.0) GO TO 51
      IF(MISPDT(1,I).NE.0.0) GO TO 52
      IF(MISPDT(3,I).NE.0.0) GO TO 53
   51 R1=SQRT(R(1,1,I))
      R2=SQRT(R(2,2,I))
      R3=SQRT(R(3,3,I))
      PRINT 20,I,ALAB(1,I),ALAB(2,I),XC(3,I),C(3,I),R3,XC(1,I),C(1,I),
     1 R1,XC(2,I),C(2,I),R2
   20 FORMAT(1H ,I4,1X,2A10,3(3X,1PE10.3,1X,1PE9.2,1X,1PE10.3))
      GO TO 90

   52 R1=SQRT(R(1,1,I))
      R2=SQRT(R(2,2,I))
      PRINT 22,I,ALAB(1,I),ALAB(2,I),XC(1,I),C(1,I),R1,PB,PB,PB,XC(2,I),
     1 C(2,I),R2
   22 FORMAT(1H ,I4,1X,2A10,3X,1PE10.3,1X,1PE9.2,1X,1PE10.3,3X,A10,
     A 1X,A9,1X,A10,3X,1PE10.3,1X,1PE9.2,1X,1PE10.3)
      GO TO 90

   53 R1=SQRT(R(1,1,I))
      R2=SQRT(R(2,2,I))
      PRINT 21,I,ALAB(1,I),ALAB(2,I),TB,TB,TB,XC(1,I),C(1,I),R1,XC(2,I),
     1 C(2,I),R2
      GO TO 90

   60 IF(MISPDT(1,I).NE.0) GO TO 61
      IF(MISPDT(3,I).NE.0) GO TO 62
      R2=0.0
      R1=SQRT(R(1,1,I))
      R3=SQRT(R(3,3,I))
      PRINT 20,I,ALAB(1,I),ALAB(2,I),XC(3,I),C(3,I),R3,XC(1,I),C(1,I),
     1 R1,XC(2,I),0.0,0.0
      GO TO 90

   61 R1=SQRT(R(1,1,I))
      R2=SQRT(R(2,2,I))
      PRINT 22,I,ALAB(1,I),ALAB(2,I),XC(1,I),C(1,I),R1,PB,PB,PB,XC(2,I),
     1 0.0,0.0
      GO TO 90

   62 R1=SQRT(R(1,1,I))
      PRINT 21,I,ALAB(1,I),ALAB(2,I),TB,TB,TB,XC(1,I),C(1,I),R1,XC(2,I),
     1 0.0,0.0
   90 IF(MOD(K,5).EQ.0) PRINT 15
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DIMPARS(KK,SCDIS,SCPRE,SCTIM,PARS,NP,VPARS,ERZS,
     APARSD,VPARSD,TITLE)
C  THIS COMPUTES DIMENSIONAL VALUES OF SHOCK PARAMETERS
C
C  KK    =  MODIFIER INDICATING WHAT HAS BEEN ADJUSTED
C  SCDIS, SCPRE, SCTIM  =  SCALES OF PARS AND VPARS
C  PARS(10)    =  SHOCK FITTING PARAMETERS
C  NP         =  NUMBER OF SHOCK FITTING PARAMETERS
C  VPARS(10,10)  =  VARIANCE MATRIX OF PARAMETERS PARS
C  ERZS     =  STANDARD ERRROR OF A SET WITH WEIGHT ONE
C  PARSD(10)    =  SHOCK FITTING PARAMETERS IN SI UNITS
C  VPARSD(10,10)  =  VARIANCE MATRIX OF PARAMETERS PARSD
C  TITLE(3)    =  NAME OF EVENT
C
      DIMENSION PARS(10),VPARS(10,10),PARSD(10),VPARSD(10,10),TITLE(3)
      DIMENSION SCMAT(10,10),DIM(10)
      COMMON/CF2DER/  GAMCAP,SNDSPD,CPAR(4),DLIM,SCD,SCP,SCT
C
      DATA((DIM(J),J=1,4)=7HPA*M   ,7HPA*M**2,7HPA*M**3,7HS      )
C
      PRINT 11,(TITLE(J),J=1,3)
11    FORMAT(1H1,10X,5HEVENT,5X,3A10,/,1H ,10X,5(1H-))
C
      DO 15 KA=1,10 $ DO 15 KB=1,10
   15 SCMAT(KA,KB)=0
      SCMAT(1,1)=SCPRE*SCDIS
      SCMAT(2,2)=SCPRE*SCDIS**2
      SCMAT(3,3)=SCPRE*SCDIS**3
      SCMAT(4,4)=SCTIM
C
      DO 45 KA=1,4 $ PARSD(KA)=0
      DO 35 KB=1,4 $ VPARSD(KA,KB)=0
      DO 25 KC=1,4 $ DO 25 KD=1,4
   25 VPARSD(KA,KB)=VPARSD(KA,KB)+SCMAT(KA,KC)*VPARS(KC,KD)*SCMAT(KD,KB)
   35 PARSD(KA)=PARSD(KA)+SCMAT(KA,KB)*PARS(KB)
   45 CONTINUE
C
      PRINT 55
   55 FORMAT(1H0,///,1H ,10X,32HDIMENSIONAL VALUES OF PARAMETERS,/)
      PRINT 65
   65 FORMAT(1H0,10X,10HPARAMETERS,5X,8HSTANDARD,7X,8HSTANDARD,5X,
     A9HDIMENSION,/,1H ,26X,6HERRORS,7X,10HERRORS*ERZ,/)
      DO 85 KA=1,NP
      PER=SQRT(VPARSD(KA,KA)) $ PERZ=PER*ERZS
      PRINT 75,PARSD(KA),PER,PERZ,DIM(KA)
   75 FORMAT(1H ,9X,1PE12.5,3X,1PE10.3,4X,1PE10.3,6X,A7)
   85 CONTINUE
      DLIMD=DLIM*SCD
      IF(NP.EQ.4)PRINT 88,DLIMD
88    FORMAT(1H+,62X,23H= SHOCK ARRIVAL TIME AT,1PE12.3,7H METRES)
      PRINT 95
   95 FORMAT(1H0,///,1H ,10X,31HTHE SHOCK OVERPRESSURE FUNCTION,
     A 12H IS GIVEN BY,
     B //,1H ,20X,40HP = PAR(1)/R + PAR(2)/R**2 + PAR(3)/R**3,//)

      PRINT 135
135   FORMAT(1H0,//,1H ,10X,37HADJUSTED ARE OBSERVATIONS OF PRESSURE)
```

```
      IF(KK.EQ.2) PRINT 136
136   FORMAT(1H+,47X,13H AND DISTANCE)
      IF(KK.EQ.3) PRINT 137
137   FORMAT(1H+,47X,19H, DISTANCE AND TIME)
      IF(KK.EQ.4) PRINT 138
138   FORMAT(1H+,47X,9H AND TIME)

C  COMPUTE CORRELATION MATRIX
      DO 185 KA=1,NP $ DO 185 KB=1,NP
  185 SCMAT(KA,KB)=VPARS(KA,KB)/SQRT(VPARS(KA,KA)*VPARS(KB,KB))
      PRINT 195
  195 FORMAT(1H ,///,1H ,10X,18HCORRELATION MATRIX,//)
      DO 215 KA=1,NP
      PRINT 205,(SCMAT(KA,J),J=1,NP)
  205 FORMAT(1H ,10X,6(0PF13.8))
  215 CONTINUE

      PRINT 105
  105 FORMAT(1H ,//,1H ,10X,27HVARIANCE-COVARIANCE MATRIX ,
     A33H(NOT INCLUDING THE FACTOR ERZ**2),//)
      DO 125 KA=1,NP
      PRINT 115,(VPARSD(KA,J),J=1,NP)
  115 FORMAT(1H ,10X,6(3X,1PE12.5))
  125 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLPDSH(KK,SCDI,SCPR,SCTI,NRSHOK,PAR,NP,VPAR,ERZ,
     AERFACT)
C  THIS PLOTS PRESSURE OVER DISTANCE (DATA AND FITTED CURVE)
C
C  KK  =  INDICATES WHAT HAS BEEN ADJUSTED. SEE STAT. 185 FF.
C  SCDI,SCPR,SCTI = SCALES TO BE USED ON INPUT DATA
C  NRSHOK = NUMBER OF INPUT DATA SETS
C  PAR = PARAMETERS OF SHOCK FITTING FUNCTION
C  NP = NUMBER OF PARAMETERS
C  VPAR = VARIANCE-COVARIANCE MATRIX OF PARAMETERS
C  ERZ = STANDARD ERROR OF SET WITH WEIGHT ONE
C  ERFACT = ERROR FACTOR TO BE USED FOR CONFIDENCE CURVES
C
C  PROGRAM CALLS ROUTINE SHOCK3 TO GET FITTED CURVE
C
      DIMENSION PAR(10),VPAR(10,10)
C
      COMMON/COMSHOT/TPXH(4,50),ERTPXH(4,50),TITLE(3),ALB(2,50)
      COMMON/AMBCHA/AMB(8)
C  THIS CONTAINS INPUT DATA
C
      COMMON/PLOT/PD(6),PLABL(4)
C  FROM THIS COMMON BLOCK USE ONLY THE PLOTTING LABEL
C
      COMMON/CMISFM/MISPDT(3,50),DISTN(50),NODIST,SCDD
C
      DIMENSION PMIMA(2),DMIMA(2),TMIMA(2),
     AQ(5,1),FX(5),FP(10),FXX(5,5),FXP(5,10),FPP(10,10),
     BTEXT(6),XP(201),YP(201),YPE(201),ERYP(201)
C
      DATA(ANAME=6HPLPDSH)
      CALL LOGSC(SCDI,SCPR,SCTI,ANAME,DMIMA,PMIMA,TMIMA,SCL,NBD)
      IF(NBD.NE.0)RETURN
C  THIS ESTABLISHED LOGARITHMIC PLOTTING SCALES
C
      CALL PLTBEG(21.0,28.0,0.3973,13,PLABL)
      XSC=SCL $ XOR=DMIMA(1) $ XRAN=DMIMA(2)-DMIMA(1)
      YSC=SCL $ YOR=PMIMA(1) $ YRAN=PMIMA(2)-PMIMA(1)
      CALL PLTSCA(5.0,9.0,XOR,YOR,XSC,YSC)
      DX=1. $ XLEFT=XOR $ XRIGHT=XOR+AMAX1(XRAN,AINT(10.*XSC))
      DY=1. $ YBOT=YOR $ YTOP=YOR+AMAX1(YRAN,AINT(10.*YSC))
      NTYPE=7
      CALL PLTAXS(DX,DY,XLEFT,XRIGHT,YBOT,YTOP,NTYPE)
      CALL LABLOG(DX,DY,XLEFT,XRIGHT,YBOT,YTOP,0.0,0.0)
   25 FORMAT(3A10,1H>)
      ENCODE(31,25,TEXT(1))(TITLE(J),J=1,3)
      CALL PLTSYM(0.4,TEXT(1),0.0,XLEFT,YBOT-YSC*4.0)
   35 FORMAT(13HDISTANCE (M)>)
      ENCODE(13,35,TEXT(1))
      TX=(XLEFT+XRIGHT)*0.5-6.0*0.3*XSC
      TY=YBOT-1.5*YSC
      CALL PLTSYM(0.3,TEXT(1),0.0,TX,TY)
   36 FORMAT(18HOVERPRESSURE (PA)>)
      ENCODE(18,36,TEXT(1))
      TX=XLEFT-1.8*XSC
      TY=(YBOT+YTOP)*0.5-8.0*0.3*YSC
      CALL PLTSYM(0.3,TEXT(1),90.0,TX,TY)
```

```
C
      DO 45 KA=1,NRSHOK
      IF(MISPOT(2,KA).NE.0) GO TO 45
      IF(MISPOT(1,KA).NE.0)GOTO 45
      XP(1)=0.5*ALOG10((TPXH(3,KA)**2+(TPXH(4,KA)-AMB(7))**2)/SCDI**2)
C  AMB(7) = CHARGE ELEVATION   (IN COMMON/AMBCHA/ )
      YP(1)=ALOG10(TPXH(2,KA)/SCPR)
      NS=MISPOT(3,KA) $ CALL PLTDTS(3,NS,XP,YP,1,0)
   45 CONTINUE
C  THIS PLOTTED DATA POINTS
C
C  NEXT PLOT FITTED CURVE
      CALL PLTWND(XLEFT,XRIGHT,YBOT,YTOP)
      IP=1
      DO 65 KA=1,201
      XP(IP)=XLEFT+(XRIGHT-XLEFT)*FLOAT(KA-1)/200.
      Q(1,1)=0 $ Q(2,1)=10.**XP(IP)*SCDI
C
      CALL SHOCK3(Q,1,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS CALL TO THE CONSTRAINT FUNCTION FURNISHES FITTED CURVE
C
      IF(F.GE.0..OR.NBAD.NE.0)GOTO 65
      YP(IP)=ALOG10(-F/(Q(2,1)**3*SCPR))
      FY=0
      DO 55 KB=1,NP $ DO 55 KC=1,NP
   55 FY=FY+FP(KB)*VPAR(KB,KC)*FP(KC)
      ERYP(IP)=SQRT(FY)/(ALOG(10.)*(-F))
C  LOGARITHMIC ERROR IS INDEPENDENT OF SCALE
      IP=IP+1
   65 CONTINUE

      IPM=IP-1 $ IF(IPM.LE.0)GOTO 120
      DO 105 KE=1,2
      DO 95 KB=1,3 $ ERF=ERFACT*FLOAT(KB-2)
      IF(KE.EQ.1)GOTO 75 $ IF(ERZ.LT.1.5)GOTO 105 $ ERF=ERF*ERZ
   75 DO 85 KP=1,IPM
   85 YPE(KP)=YP(KP)+ERF*ERYP(KP)
      CALL PLTDTS(1,0,XP,YPE,IPM,0)
   95 CONTINUE
  105 CONTINUE

  115 FORMAT(21HCONFIDENCE LIMITS FOR,F4.1,17H STANDARD ERRORS>)
  120 ENCODE(42,115,TEXT(1))ERFACT
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.0)
      IF(ERZ.GE.1.5)GOTO 145
  125 FORMAT(24HWITHOUT THE FACTOR ERZ =,F6.3,1H>)
      ENCODE(31,125,TEXT(1))ERZ
      GOTO 155
  135 FORMAT(33HWITH AND WITHOUT THE FACTOR ERZ =,F6.3,1H>)
  145 ENCODE(40,135,TEXT(1))ERZ
  155 CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.4)
      IF(KK.NE.1) GOTO 175
165   FORMAT(38HADJUSTED ARE OBSERVATIONS OF PRESSURE>)
      ENCODE(38,165,TEXT(1))
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC *5.8)
      GOTO 265
175   ENCODE(29,185,TEXT(1))
```

```
185   FORMAT(29HADJUSTED ARE OBSERVATIONS OF>)
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC *5.8)
      IF(KK.EQ.2) GO TO 195
      IF(KK.EQ.3) GOTO 215
      IF(KK.EQ.4) GOTO 235
      GOTO 265
195   ENCODE(22,205,TEXT(1))  $  GOTO 255
205   FORMAT(22HPRESSURE AND DISTANCE>)
215   ENCODE(28,225,TEXT(1))         $  GOTO 255
225   FORMAT(28HPRESSURE, DISTANCE AND TIME>)
235   ENCODE(18,245,TEXT(1))  $  GOTO 255
245   FORMAT(18HPRESSURE AND TIME>)
255   CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC *6.2)
265   CONTINUE
      CALL PLTPGE
      RETURN
      END
```

```
      SUBROUTINE PLPTSH(KK,SCDIST,SCPRES,SCTIME,NRSHOK,PAR4,NP,V4,
     1 ERZ4,ERFAC)
C  THIS PLOTS PRESSURE OVER TIME (DATA AND FITTED CURVE)
C
C  KK   =  INDICATES WHAT HAS BEEN ADJUSTED
C  SCDIST,SCPRES,SCTIME  =  SCALES TO BE USED ON INPUT DATA
C  NRSHOK  =  NUMBER OF SHOCK OBSERVATION STATIONS
C  PAR4(10)  =  SHOCK PARAMETERS
C  NP     =  NUMBER OF SHOCK PARAMETERS
C  V4(10,10)  =  VARIANCE MATRIX OF SHOCK PARAMERERS PAR4
C  ERZ4    =  STANDARD ERROR OF A SET WITH WEIGHT ONE
C  ERFAC    =  FACTOR FOR CONFIDENCE LIMIT PLOTTING
C
C  ROUTINE USES SHOCK3 AND F2SHCK FOR THE COMPUTATION OF FITTED PRESSURE
C
      DIMENSION PAR4(10),V4(10,10),TEXT(6)                          CURVE
C
      DIMENSION PMIMA(2),DMIMA(2),TMIMA(2)
      DIMENSION XP(201),YP(201),EYP(201),YPE(201),Q(5,1)
      DIMENSION FX(5),FP(10),FXX(5,5),FXP(5,10),FPP(10,10)
C
      COMMON/COMSHDT/TPXH(4,50),ERTPXH(4,50),TITLE(3),ALAB(2,50)
      COMMON/AMBCHA/AMPR,AMTEM,GAMMA,AMMOL,CHVOL,CHEN,HC,ERCHEL
CC  THESE TWO COMMON BLOCKS CONTAIN INPUT DATA
C
      COMMON/PLOT/PD(6),PLABL(4)
C  FROM THIS COMMON BLOCK USE ONLY THE PLOTTING LABEL
C
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCD,SCP,SCT
C  THIS COMMON BLOCK IS NEEDED BY THE CONSTRAINT ROUTINES
C
      COMMON/CMISFM/MISPDT(3,50),DISTN(50),NODIST,SCDD
C  MISPDT IS USED TO IDENTIFY MISSING DATA
C
      DATA(ANAME=6HPLPTSH)

      IF(KK.LE.2) RETURN
C  PLOT OVER TIME ONLY IF TIME IS AN OBSERVABLE
C
      SNDSPD=SNDSPD*SCD*SCTIME/(SCT*SCDIST)
      ALOW=ALOW*SCD/SCDIST
      GAMCAP=GAMCAP*SCPRES/SCP
      SCD=SCDIST
      SCT=SCTIME
      SCP=SCPRES
C  THIS WILL CAUSE F2DER TO PRODUCE RESULTS IN THE PROPER SCALES
C
      CALL LOGSC(SCDIST,SCPRES,SCTIME,ANAME,DMIMA,PMIMA,TMIMA,SCL,NBD)
      IF(NBD.NE.0) RETURN
C  LOGSC COMPUTED PROPER PLOTTING SCALES
C
      CALL PLTBEG(21.0,28.0,0.394,13,PLABL)
      XSC=SCL $ XOR=TMIMA(1) $ XRAN=TMIMA(2)-TMIMA(1)
      YSC=SCL $ YOR=PMIMA(1) $ YRAN=PMIMA(2)-PMIMA(1)
      DX=1. $ XLEFT=XOR $ XRIGHT=XLEFT+AMAX1(XRAN,AINT(10.*XSC))
      DY=1. $ YBOT=YOR $ YTOP=YBOT+AMAX1(YRAN,AINT(10.*YSC))
      CALL PLTSCA(5.0,9.0,XOR,YOR,XSC,YSC)
```

```
      NTYPE=7
      CALL PLTAXS(DX,DY,XLEFT,XRIGHT,YBOT,YTOP,NTYPE)
      DXL=1.0 $ DYL=1.0
      CALL LABLOG(DXL,DYL,XLEFT,XRIGHT,YBOT,YTOP,0.0,0.0)
      TEX=10HTIME (S)>  $ ENCODE(10,179,TEXT(1))TEX
  179 FORMAT(A10)
      TX=(XLEFT+XRIGHT)*0.5-4.0*0.3*YSC
      TY=YBOT-1.5*YSC
      CALL PLTSYM(0.3,TEXT(1),0.0,TX,TY)
  301 FORMAT(18HOVERPRESSURE (PA)>)
      ENCODE(13,301,TEXT(1))
      TX=XLEFT-XSC*1.7
      TY=(YBOT+YTOP)*0.5-8.0*0.3*YSC
      CALL PLTSYM(0.3,TEXT(1),90.0,TX,TY)
      ENCODE(31,178,TEXT(1))(TITLE(J),J=1,3)
      CALL PLTSYM(0.4,TEXT(1),0.0,XLEFT,YBOT-YSC*4.0)
      NPP=0 $ DO 197 KP=1,NRSHOK
      IF(MISPDT(1,KP).NE.0.OR.MISPDT(3,KP).NE.0) GOTO 197
      NPP=NPP+1
      XP(NPP)=ALOG10(TPXH(1,KP)/SCTIME)
      YP(NPP)=ALOG10(TPXH(2,KP)/SCPRES)
197   CONTINUE
      CALL PLTDTS(3,0,XP,YP,NPP,0)
C  THIS PLOTTED DATA
C
C  NEXT FIND SUCH DISTANCE LIMITS THAT CORRESPOND TO P,T-WINDOW
      DPLRAN=DMIMA(2)-DMIMA(1)
      DISTMI=DMIMA(1)
      DISTMA=DISTMI+AMAX1(DPLRAN,AINT(10.*SCL))
      DELDX=(DISTMA-DISTMI)/20.
      Q(1,1)=0.$ Q(3,1)=0
      LOW=0
      DX=DISTMI
  405 Q(2,1)=10.**DX*SCDIST
      CALL SHOCK3(Q,1,PAR4,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS ROUTINE COMPUTES THE NEGATIVE OVERPRESSURE
      OVP=-F/(Q(2,1)**3*SCPRES)
      IF(OVP.LE.0.) GOTO 425
      IF(NBAD.NE.0.OR.ALOG10(OVP).GT.YTOP) GOTO 415
C  BRANCH IF PRESSURE IS OUTSIDE WINDOW
      CALL F2SHCK(Q,1,PAR4,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS ROUTINE COMPUTES TIME
      TIM=F/(SNDSPD*SCTIME)
      IF(TIM.LE.0.) GOTO 425
      IF(NBAD.NE.0.OR.ALOG10(TIM).LT.XLEFT) GOTO 415
C  BRANCH IF TIME IS OUTSIDE WINDOW
      LOW=1
C  AN INSIDE POINT FOUND. GET A LOWER LIMIT OUTSIDE POINT
      DX=DX-DELDX
      GOTO 405
  415 IF(LOW.EQ.1)GOTO 425
      DX=DX+DELDX
      GOTO 405
C  NEXT SEARCH FOR UPPER LIMIT
  425 DISTMI=DX
      LAR=0
      DX=DISTMA
```

```
  435 Q(2,1)=10.**DX*SCDIST
      CALL F2SHCK(Q,1,PAR4,F,FX,FP,FXX,FXP,FPP,NBAD)
      TIM=F/(SNDSPD*SCTIME)
      IF(TIM.LE.0.) GOTO 455
      IF(NBAD.NE.0.OR.ALOG10(TIM).GT.XRIGHT) GOTO 445
C  BRANCH IF TIME OUTSIDE WINDOW
      CALL SHOCK3(Q,1,PAR4,F,FX,FP,FXX,FXP,FPP,NBAD)
      OVP=-F/(Q(2,1)**3*SCPRES)
      IF(OVP.LE.0.) GOTO 455
      IF(NBAD.NE.0.OR.ALOG10(OVP).LT.YBOT) GOTO 445
C  BRANCH IF PRESSURE OUTSIDE WINDOW
      DX=DX+DELDX
      LAR=1
C  AN INSIDE POINT HAS BEEN FOUND. GET AN OUTSIDE POINT
      GOTO 435
  445 IF(LAR.EQ.1)GOTO 455
      DX=DX-DELDX
      GOTO 435
  455 DISTMA=DX

C  NEXT COMPUTE FITTED CURVE FOR PLOTTING
      IP=1
      DO 201 KP=1,201
      PXP=DISTMI+(DISTMA-DISTMI)*FLOAT(KP-1)/200.
      Q(1,1)=0.$ Q(2,1)=10.**PXP*SCDIST$ Q(3,1)=0
C
      CALL SHOCK3(Q,1,PAR4,F,FX,FP,FXX,FXP,FPP,NBAD)
C  FIRST SHOCK FITTING CONSTRAINT ROUTINE PROVIDES PRESSURE
      IF(F.GE.0..OR.NBAD.NE.0) GOTO 201
C
      YP(IP)=ALOG10(-F/(Q(2,1)**3*SCPRES))
      EY=0. $ DO 199 KB=1,NP  $  DO 199 KC=1,NP
199   EY=EY+FP(KB)*V4(KB,KC)*FP(KC)
      EYP(IP)=SQRT(EY)/(ALOG(10.)*(-F))
C
      CALL F2SHCK(Q,1,PAR4,F,FX,FP,FXX,FXP,FPP,NBAD)
C  SECOND SHOCK FITTING CONSTRAINT ROUTINE PROVIDES TIME
      IF(F.LE.0..OR.NBAD.NE.0) GOTO 201
C
      XP(IP)=ALOG10(F/(SNDSPD*SCTIME))
      IP=IP+1
  201 CONTINUE

C  NEXT PLOT FITTED CURVE
      CALL PLTWNO(XLEFT,XRIGHT,YBOT,YTOP)
      DO 2031 KE=1,2
      KPM=IP-1  $  IF(KPM.LE.0) GOTO    2031
      DO 203 KB=1,3    $    ERF=ERFAC*FLOAT(KB-2)
      IF(KE.NE.2)GOTO 2011 $ IF(ERZ4.LT.1.5)GOTO 203 $ ERF=ERF*ERZ4
2011  CONTINUE
      DO 202 KP=1,KPM
      YPE(KP)=YP(KP)+EYP(KP)*ERF
  202 CONTINUE
      CALL PLTDTS(1,0,XP,YPE,KPM,0)
203   CONTINUE
2031  CONTINUE
```

```
      ENCODE(60,5,TEXT(1))ERFAC
    5 FORMAT(*CONFIDENCE LIMITS FOR *,F4.1,* STANDARD ERRORS>*)
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.0)
      IF(ERZ4.GE.1.5) GO TO 14
      ENCODE(31,10,TEXT(1)) ERZ4
   10 FORMAT(*WITHOUT THE FACTOR ERZ =*,F6.3,1H>)
      GO TO 16
   14 ENCODE(40,15,TEXT(1)) ERZ4
   15 FORMAT(*WITH AND WITHOUT THE FACTOR ERZ =*,F6.3,1H>)
   16 CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.4)
      IF(KK.NE.1) GO TO 24
      ENCODE(38,20,TEXT(1))
   20 FORMAT(*ADJUSTED ARE OBSERVATIONS OF PRESSURE>*)
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.8)
      GO TO 265
   24 ENCODE(29,25,TEXT(1))
   25 FORMAT(*ADJUSTED ARE OBSERVATIONS OF>*)
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.8)
      IF(KK.EQ.2) GO TO 195
      IF(KK.EQ.3) GO TO 215
      IF(KK.EQ.4) GO TO 235
  195 ENCODE(22,205,TEXT(1))
      GO TO 255
  215 ENCODE(28,225,TEXT(1))
      GO TO 255
  235 ENCODE (18,245,TEXT(1))
  255 CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*6.2)
  178 FORMAT(3A10,1H>)
  180 FORMAT(5HCASE ,I2,6H,  NX=,I1,5H, NP=,I1,1H>)
  205 FORMAT(22HPRESSURE AND DISTANCE>)
  225 FORMAT(27HPRESSURE,DISTANCE AND TIME>)
  245 FORMAT(18HPRESSURE AND TIME>)
  265 CALL PLTPGE
      RETURN
      END
```

```
      SUBROUTINE PLOTSH(KK,SCDIST,SCPRES,SCTIME,NRSHOK,PAR4,NP,V4,
     1 ERZ4,ERFAC)
C  THIS PLOTS DISTANCE OVER TIME (DATA AND FITTED CURVE)
C
C  KK   =  INDICATES WHAT HAS BEEN ADJUSTED
C  SCDIST, SCPRES, SCTIME  =  SCALES TO BE USED ON INPUT DATA
C  PAR4(10)  =  SHOCK FITTING PARAMETERS
C  NP   =   NUMBER OF SHOCK FITTING PARAMETERS
C  V4(10,10)  =  VARIANCE MATRIX OF SHOCK PARAMETERS PAR4
C  ERZ4  =  STANDARD ERROR OF A SET WITH WEIGHT ONE
C  ERFAC  =  FACTOR FOR PLOTTING OF CONFIDENCE LIMITS
C
C  ROUTINE USES CONSTRAINT ROUTINE F2SHCK TO COMPUTE TIME FOR GIVEN
C                       DISTANCE
C
      DIMENSION PAR4(10),V4(10,10),TEXT(6)
C
      DIMENSION PMIMA(2),DMIMA(2),TMIMA(2)
      DIMENSION XP(201),YP(201),EYP(201),YPE(201),Q(5,1)
      DIMENSION FX(5),FP(10),FXX(5,5),FXP(5,10),FPP(10,10)
C
      COMMON/COMSHDT/TPXH(4,50),ERTPXH(4,50),TITLE(3),ALAB(2,50)
      COMMON/AMBCHA/AMB(8)
C  THESE TWO COMMON BLOCKS CONTAIN INPUT DATA
C
      COMMON/CMISFM/MISPDT(3,50),DISTN(50),NODIST,SCDD
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCD,SCP,SCT
C  THESE TWO COMMON BLOCKS ARE NEEDED BY THE CONSTRAINT ROUTINE F2SHCK
C
      COMMON/PLOT/PD(6),PLABL(4)
C  FROM THIS COMMON BLOCK USE ONLY THE PLOTTING LABEL
C
      DATA(ANAME=6HPLOTSH)
C
      IF(KK.LE.2) RETURN
C  NO PLOTTING IF TIME WAS NOT ADJUSTED
C
      SNDSPD=SNDSPD*SCD*SCTIME/(SCT*SCDIST)
      ALOW=ALOW*SCD/SCDIST
      GAMCAP=GAMCAP*SCPRES/SCP
      SCD=SCDIST
      SCT=SCTIME
      SCP=SCPRES
C  THIS WILL CAUSE F2SHCK TO FURNISH RESULTS IN THE PROPER SCALES
C
      CALL LOGSC(SCDIST,SCPRES,SCTIME,ANAME,DMIMA,PMIMA,TMIMA,SCL,NBD)
      IF(NBD.NE.0) RETURN
C  LOGSC ESTABLISHED PLOTTING SCALES FOR LOGARITHMIC PLOTTING
C
      CALL PLTBEG(21.0,28.0,0.394,13,PLABL)
      XSC=SCL $ XOR=TMIMA(1) $ XRAN=TMIMA(2)-TMIMA(1)
      YSC=SCL $ YOR=DMIMA(1) $ YRAN=DMIMA(2)-DMIMA(1)
      DX=1. $ XLEFT=XOR $ XRIGHT=XLEFT+AMAX1(XRAN,AINT(10.*XSC))
      DY=1. $ YBOT=YOR $ YTOP=YBOT+AMAX1(YRAN,AINT(10.*YSC))
      CALL PLTSCA(5.0,9.0,XOR,YOR,XSC,YSC)
      NTYPE=7
      CALL PLTAXS(DX,DY,XLEFT,XRIGHT,YBOT,YTOP,NTYPE)
```

```
      DXL=1.0 $ DYL=1.0
      CALL LABLOG(DXL,DYL,XLEFT,XRIGHT,YBOT,YTOP,0.0,0.0)
   35 FORMAT(13HDISTANCE (M)>)
      ENCODE(13,35,TEXT(1))
      TX=XLEFT-XSC*1.7
      TY=(YBOT+YTOP)/2.-YSC*6.0*0.3
      CALL PLTSYM(0.3,TEXT(1),90.0,TX,TY)
   36 FORMAT(9HTIME (S)>)
      ENCODE(9,36,TEXT(1))
      TX=(XLEFT+XRIGHT)/2.-XSC*4.0*0.3
      TY=YBOT-YSC*1.5
      CALL PLTSYM(0.3,TEXT(1),0.0,TX,TY)
      ENCODE(31,178,TEXT(1))(TITLE(J),J=1,3)
      CALL PLTSYM(0.4,TEXT(1),0.0,XLEFT,YBOT-YSC*4.0)

      DO 197 KP=1,NRSHOK
      IF(MISPDT(2,KP).NE.0) GO TO 197
      IF(MISPDT(3,KP).NE.0) GO TO 197
      XP(1)=ALOG10(TPXH(1,KP)/SCTIME)
      YP(1)=0.5*ALOG10((TPXH(3,KP)**2+(TPXH(4,KP)-AMB(7))**2)/SCDIST**2)
      NS=MISPDT(1,KP)
      CALL PLTDTS(3,NS,XP,YP,1,0)
  197 CONTINUE
C  THE PREVIOUS LOOP PLOTTED DATA
C
C  NEXT PLOT ADJUSTED CURVE
      CALL PLTWND(XLEFT,XRIGHT,YBOT,YTOP)  $ IP=1

      DO 238 KP=1,201
      YP(IP)=YBOT+(YTOP-YBOT)*FLOAT(KP-1)/200.
      Q(1,1)=0. $ Q(2,1)=10.**YP(IP)*SCDIST$ Q(3,1)=0.
C
      CALL F2SHCK(Q,1,PAR4,F,FX,FP,FXX,FXP,FPP,NBAD)
      IF(NBAD.NE.0) RETURN
C  THE CONSTRAINT ROUTINE COMPUTED TIME FOR GIVEN DISTANCE
C
      XP(IP)=ALOG10(F/(SNDSPD*SCTIME))
      DUM=0. $ DO 236 KB=1,NP $ DO 236 KC=1,NP
  236 DUM=DUM+FP(KB)*V4(KB,KC)*FP(KC)
      EYP(IP)=SQRT(DUM)/(F*ALOG(10.))
      IP=IP+1
  238 CONTINUE

      DO 2451 KE=1,2
      KPM=IP-1 $ IF(KPM.LE.0) GO TO 2451
      DO 246 KB=1,3 $ ERF=ERFAC*FLOAT(KB-2)
      IF(KE.NE.2) GO TO 2381 $ IF(ERZ4.LT.1.5) GO TO 246 $ ERF=ERF*ERZ4
 2381 CONTINUE
      DO 243 KP=1,KPM
  243 YPE(KP)=XP(KP)+EYP(KP)*ERF
      CALL PLTDTS(1,0,YPE,YP,KPM,0)
  246 CONTINUE
 2451 CONTINUE

      ENCODE(60,5,TEXT(1))   ERFAC
    5 FORMAT(*CONFIDENCE LIMITS FOR *,F4.1,* STANDARD ERRORS>*)
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.0)
```

```
      IF(ERZ4.GE.1.5) GO TO 14
      ENCODE(31,10,TEXT(1)) ERZ4
   10 FORMAT(*WITHOUT THE FACTOR ERZ =*,F6.3,1H>)
      GO TO 16
   14 ENCODE(40,15,TEXT(1)) ERZ4
   15 FORMAT(*WITH AND WITHOUT THE FACTOR ERZ =*,F6.3,1H>)
   16 CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.4)
      IF(KK.NE.1) GO TO 24
      ENCODE(38,20,TEXT(1))
   20 FORMAT(*ADJUSTED ARE OBSERVATIONS OF PRESSURE>*)
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.8)
      GO TO 265
   24 ENCODE(29,25,TEXT(1))
   25 FORMAT(*ADJUSTED ARE OBSERVATIONS OF>*)
      CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*5.8)
      IF(KK.EQ.2) GO TO 195
      IF(KK.EQ.3) GO TO 215
      IF(KK.EQ.4) GO TO 235
  195 ENCODE(22,205,TEXT(1))
      GO TO 255
  215 ENCODE(28,225,TEXT(1))
      GO TO 255
  235 ENCODE (18,245,TEXT(1))
  255 CALL PLTSYM(.25,TEXT(1),0.0,XLEFT,YBOT-YSC*6.2)
  178 FORMAT(3A10,1H>)
  180 FORMAT(5HCASE ,I2,6H,  NX=,I1,5H, NP=,I1,1H>)
  205 FORMAT(22HPRESSURE AND DISTANCE>)
  225 FORMAT(27HPRESSURE,DISTANCE AND TIME>)
  245 FORMAT(18HPRESSURE AND TIME>)
  265 CALL PLTPGE
      RETURN
      END
```

```
      SUBROUTINE LOGSC(SCDI,SCPR,SCTI,ANAME,DMIMA,PMIMA,
     ATMIMA,SCLG10,NBAD)
C  THIS COMPUTES MINIMUM AND MAXIMUM PLOTTING LIMITS
C  AND PLOTTING SCALE FOR LOGARITHMIC PLOTS
C
C  SCDI, SCPR, SCTI  =  SCALES TO BE USED WITH DATA IN CMPLSH
C  ANAME  =  NAME OF CALLING PROGRAM
C
C  THE FOLLOWING IS COMPUTED BY LOGSCA
C
C  DMIMA(2),PMIMA(2),TMIMA(2)  = MINIMUM AND MAXIMUM VALUES OF DIST,P,T
C           REPRESENTING COORDINATE WINDOWS FOR LOGARITHMIC PLOTS
C  SCLG10  =  LOGARITHMIC SCALE DETERMINED SUCH THAT ALL QUANTITIES
C       CAN BE LOGARITHMICALLY PLOTTED WITHIN A  15 X 15 CM SQUARE
C  NBAD  =  ERROR INDICATOR.   NBAD.EQ.0  IF NO ERROR
C
      DIMENSION DMIMA(2),PMIMA(2),TMIMA(2)
      COMMON/CMPLSH/PMIN,PMAX,DMIN,DMAX,TMIN,TMAX
C  THIS COMMON BLOCK CONTAINS THE EXTREME DATA VALUES
C
      NBAD=0
      IF(SCDI.GT.O..AND.SCPR.GT.O..AND.SCTI.GT.O.)GOTO 25
      NBAD=1
      PRINT 15,ANAME,SCDI,SCPR,SCTI
      RETURN
   15 FORMAT(1HO,10X,15HNO PLOTTING BY ,A6,8H BECAUSE,
     A33H PLOTTING SCALES ARE NOT POSITIVE,/,1H ,10X,
     B20HDISTANCE SCALE SCDI=,1PE12.5,/,1H ,10X,
     C20HPRESSURE SCALE SCPR=,1PE12.5,/,1H ,10X,
     D20HTIME SCALE     SCTI=,1PE12.5,/)
   25 IF(PMIN.GT.O..AND.PMAX.GT.O.)GOTO 55
   35 NBAD=2
      PRINT 45,ANAME,PMIN,PMAX,DMIN,DMAX,TMIN,TMAX
      RETURN
   45 FORMAT(1HO,10X,15HNO PLOTTING BY ,A6,8H BECAUSE,
     A45H DATA ARE OUTSIDE RANGE FOR LOGARITHMIC PLOTS,/,
     B1H ,10X,5HPMIN=,1PE12.5,7H  PMAX=,1PE12.5,/,
     C1H ,10X,5HDMIN=,1PE12.5,7H  DMAX=,1PE12.5,/,
     D1H ,10X,5HTMIN=,1PE12.5,7H  TMAX=,1PE12.5,/)
   55 IF(DMIN.LE.O..OR.DMAX.LE.O.)GOTO 35
      IF(TMIN.LE.O..OR.TMAX.LE.O.)GOTO 35
      AP=ALOG10(PMIN/SCPR)
      PMIMA(1)=AINT(AP)+AMIN1(O.,SIGN(1.,AP))
      AP=ALOG10(PMAX/SCPR)
      PMIMA(2)=AINT(AP)+AMAX1(O.,SIGN(1.,AP))
      PMIMA(2)=AMAX1(PMIMA(2),PMIMA(1)+1.)
      AP=ALOG10(DMIN/SCDI)
      DMIMA(1)=AINT(AP)+AMIN1(O.,SIGN(1.,AP))
      AP=ALOG10(DMAX/SCDI)
      DMIMA(2)=AINT(AP)+AMAX1(O.,SIGN(1.,AP))
      DMIMA(2)=AMAX1(DMIMA(2),DMIMA(1)+1.)
      AP=ALOG10(TMIN/SCTI)
      TMIMA(1)=AINT(AP)+AMIN1(O.,SIGN(1.,AP))
      AP=ALOG10(TMAX/SCTI)
      TMIMA(2)=AINT(AP)+AMAX1(O.,SIGN(1.,AP))
      TMIMA(2)=AMAX1(TMIMA(2),TMIMA(1)+1.)
      PLOGR=PMIMA(2)-PMIMA(1)
```

```
      DLOGR=DMIMA(2)-DMIMA(1)
      TLOGR=TMIMA(2)-TMIMA(1)
60    SCLG10=AMAX1(0.2,PLOGR/15.,DLOGR/15.,TLOGR/15.)
      RETURN
      END
```

APPENDIX B
BLAST FOLD OVERPRESSURE FITTING PROGRAM BLAFOP

```
      PROGRAM OPREFIT(INPUT,OUTPUT,TAPE6=OUTPUT,TAPE13)
C  BLAST FIELD OVERPRESSURE FITTING, MAIN PROGRAM
C
      LEVEL 2,X,R,ALAB,LSTX,PRP,VPRP,PRPD,VPRPD
      COMMON X(5,100),R(5,5,100),ALAB(2,100),LSTX(100),PRP(4,50),
     1 VPRP(4,4,50),PRPD(4,50),VPRPD(4,4,50)
      COMMON/PLOT/PD(6),PLABL(4)
C
      DIMENSION TITLE(3),PAR(10),VPAR(10,10),PRDS(50),PRDSD(50),
     1 PARDIM(10),VPDIM(10,10)
      DIMENSION PIN(50),PIND(50),TAR(50),TARD(50),PRLAB(50)
      DIMENSION EXNU(3)
      DIMENSION TEND(50),TENDD(50)
C
      EXTERNAL SHOCK,PFIELD
C
      CALL READAM(SCDIS,SCPRES,SCTIME,TITLE,NBAD)
C  READ AMBIENT DATA
      IF(NBAD.NE.0.AND.NBAD.NE.3) STOP
C
      CALL READSP(NBAD)
C  THIS READS SHOCK FITTING RESULTS. THE PARAMETERS AND THEIR
C  ACCURACIES WILL BE STORED IN PROPER COMMON STORAGES.
      IF(NBAD.EQ.0) GO TO 5
      PRINT 2,NBAD
    2 FORMAT(1H ,*ERROR IN READSP,NBAD= *,I5)
      STOP
C
    5 CONTINUE
      CALL READPR(NRPROF)
C  READ ALL OVERPRESSURE HISTORY DATA.  NRPROF IS THE TOTAL NUMBER
C  OF OVERPRESSURE HISTORIES (PROFILES) IN THE INPUT.
      IF(NRPROF.GT.0) GO TO 10
      PRINT 7,NRPROF
    7 FORMAT(1H ,*ERROR IN READPR,NRPROF= *,I5)
      STOP
   10 CONTINUE
C
      DO 45 KA=1,NRPROF
      CALL SCALPR(SCDIS,SCPRES,SCTIME,KA,X,R,ALAB,LSTX,
     A NRSETS,TIMSH,PRSH,DISH,NBAD)
C  SCALE IN SI-UNITS AND STORE ONE HISTORY IN X, 1 THROUGH NRSETS.
C  SHOCK TIME, PRESSURE  AND DISTANCE ARE SCALED, TOO
      IF(NBAD.EQ.0) GO TO 15
      PRINT 12,NBAD
   12 FORMAT(1H ,*ERROR IN SCALPR,NBAD= *,I5)
      STOP
   15 CONTINUE
C
      CALL FITPR(X,R,ALAB,LSTX,NRSETS,TIMSH,PRSH,DISH,PAR,
     A VPAR,ERZ,TITLE,SCDIS,SCPRES,SCTIME,NBAD)
C  FIT THIS OVERPRESSURE HISTORY
      IF(NBAD.EQ.0) GO TO 20
      PRINT 17,NBAD
   17 FORMAT(1H ,*ERROR IN FITPR,NBAD= *,I5)
      STOP
   20 CONTINUE
```

```
C
      CALL DIMPAR(SCDIS,SCPRES,SCTIME,PAR,VPAR,ERZ,PARDIM,VPDIM)
C  COMPUTE DIMENSIONAL VALUES OF PARAMETERS AND VARIANCES (IN SI UNITS)
C
      CALL PLTPNTS(X,R,ALAB,NRSETS,PRSH,TIMSH,SCPRES,SCTIME,
     A PARDIM,VPDIM,TITLE)
C  PLOT PRESSURE HISTORY AND OBSERVED NODES WITH ERROR ELLIPSES
C  THE PLOTS WILL BE IN SI-UNITS. PARDIM IS ASSUMED TO BE IN SI.
C
      DO 35 KB=1,3 $ DO 25 KC=1,3
      VPRP(KB,KC,KA)=VPAR(KB,KC)
   25 VPRPD(KB,KC,KA)=VPDIM(KB,KC)
      PRP(KB,KA)=PAR(KB)
   35 PRPD(KB,KA)=PARDIM(KB)
C  STORE PROFILE PARAMETERS, SCALED AND DIMENSIONAL
      PRDS(KA) =DISH  $ PRDSD(KA)=DISH*SCDIS
C  STORE PROFILE DISTANCES, SCALED AND DIMENSIONAL
      PIN(KA)=PRSH $ PIND(KA)=PRSH*SCPRES
C  STORE INCIDENTAL SHOCK OVERPRESSURES
      TAR(KA)=TIMSH $ TARD(KA)=TIMSH*SCTIME
C  STORE SHOCK ARRIVAL TIMES
      TEND(KA)=X(1,NRSETS)
      DO 37 KB=1,NRSETS
37    TEND(KA)=AMAX1(TEND(KA),X(1,KB))
      TENDD(KA)=TEND(KA)*SCTIME
C  STORE HISTORY END TIMES
      PRLAB(KA)=ALAB(1,1)
C  USE LABEL OF FIRST OBSERVATION TO IDENTIFY PROFILE
C
   45 CONTINUE
C
      CALL PRINPAR(PRLAB,PRDS,TAR,PIN,PRP,VPRP,
     APRDSD,TARD,PIND,PRPD,VPRPD,NRPROF,PAR,EXNU,TITLE)
C  PRINT SUMMARY OF PRESSURE HISTORY FITTINGS
C  AND OBTAIN EXPONENTS EXNU AND INITIAL APPROXIMATIONS OF PAR
      CALL PLTPAR(NRPROF,PRPD,PRDSD,TITLE)
C  PLOT HISTORY PARAMETERS VERSUS DISTANCE
C
      CALL FTPFLD(SCDIS,SCPRES,SCTIME,TITLE,PRLAB,PRDSD,TARD,
     A PIND,NRPROF,EXNU,PAR,VPAR,ERZ,NP,NBAD)
C  FIT ALL TIME,OVERPRESSURE,DISTANCE DATA TO OBTAIN OVERPRESSURE FIELD
C
      IF(NBAD.EQ.0) GO TO 50
      PRINT 47,NBAD
   47 FORMAT(1H ,*ERROR IN FTPFLD,NBAD= *,I5)
      STOP
   50 CONTINUE
      CALL DIMFLD(SCDIS,SCPRES,SCTIME,EXNU,PAR,VPAR,ERZ,NP,
     A PARDIM,VPDIM,TITLE)
C  COMPUTE DIMENSIONAL VALUES OF OVERPRESSURE FIELD PARAMETERS
C
      SCD=1.0  $  SCP=1.0  $  SCT=1.0
C  SCALES ARE ONE IF DIMENSIONAL QUANTITIES ARE USED IN PLTLOC ARGUMENTS
      CALL PLTLOC(PRDSD,TARD,TENDD,NRPROF,PARDIM,VPDIM,NP,
     A SCD,SCP,SCT,SHOCK,TITLE)
C  PLOT HISTORY LOCATIONS IN THE  X,T PLANE
C
```

```
115       STOP
          END
```

```
      SUBROUTINE READAM(SCDIST,SCPRES,SCTIME,TITLE,NBAD)
C  THIS ROUTINE READS TITLE, PLOTLABEL AND DATA CARDS DESCRIBING
C  AMBIENT CONDITIONS AND THE CHARGE
C  FIRST TWO CARDS ARE MANDATORY AND ALPHANUMERIC  (TITLE AND PLOTLABEL)
C  THE REST OF THE CARDS HAVE THE FORMAT (2A10,6E10.3)
C  CHARGE CARD IS MANDATORY
C  IF AMBIENT DATA ARE NOT PROVIDED THEN STANDARD AIR WILL BE ASSUMED
C
C  SEQUENCE OF MANDATORY INPUT CARDS
C      TITLE CARD  (ALPHANUMERIC)
C      PLOTLABEL CARD   (ALPHANUMERIC)
C      CHARGE CARD = VOLUME, ENERGY, HIGHT, ERROR OF HIGHT
C
C  THE FOLLOWING ARE OPTIONAL INPUT CARDS IN ARBITRARY SEQUENCE
C      AMBIENT = P,TEMPERATURE, GAMMA, MOLAR MASS
C          DEFAULT VALUES CORRESPOND TO A STANDARD AIR
C      SCALES = SCALES OF R,P,T  TO BE USED IN COMPUTATIONS
C           DEFAULT VALUES ARE COMPUTED AFTER STATEMENT 1110
C      PLOTTING DATA = ERROR FACTORS FOR THE PLOTTING OF CONFIDENCE
C                      LIMITS IN HISTORY PLOTS
C           DEFAULT VALUES ARE FACTORS 2.0 FOR ALL PLOTS
C
C  END OF INPUT IS INDICATED BY A BLANK CARD
C
      DIMENSION TITLE(3)
      DIMENSION D(8),AMSTAR(4)
      COMMON/AMBCHA/AIRPR,AIRTEM,AIRGAM,AIRMOL,CHARVO,CHAREN,
     ACHARHI,CHARHER
      COMMON/PLOT/PD(6),PLABL(4)
      DATA(TITL =10HTITLE     ), (PLAB=10HPLOTLABEL )
      DATA (BLANK=10H          ),(AMB=10HAMBIENT   )
      DATA (CHA=10HCHARGE    )
      DATA(PLT=10HPLOTTING D),(SCAL=10HSCALES R,P)
   15 FORMAT(1H1,10X,20HINPUT READ BY READAM,/,1H ,10X,20(1H-),/)
25    FORMAT(8A10)
26    FORMAT(1H ,10X,8A10)
   35 FORMAT(2A10,6E10.3)
   36 FORMAT(1H , 5X,2A10,6(2X,1PE14.7))
C
      PD(1)=2.0
C  DEFAULT VALUE FOR PLOTTING ERROR LIMITS IN PRESSURE HISTORIES
      PD(2)=2.0
C  DEFAULT VALUE FOR PLOTTING FIELD HISTORIES (P,V,RHO,V**2*RHO/2.)
      AIRPR=101325.0  $  AIRTEM=293.0  $  AIRGAM=1.4
      AIRMOL=0.02896  $  AIRDEN=(AIRMOL/8.3143)*(AIRPR/AIRTEM)
C  THESE ARE STANDARD AIR DEFAULT VALUES FOR AMBIENT CONDITIONS
C
      NSCAL=0   $ NAMSTAR=0
      NAMB=0 $ NCHA=0
      DO 37 J=1,4
37    AMSTAR(J)=1H
      PRINT 15
      DO 46 KK=1,2
      READ 25,(D(J),J=1,8)
      PRINT 26,(D(J),J=1,8)
      IF(D(1).EQ.TITL ) GOTO 42
```

```
         IF(D(1).EQ.PLAB) GOTO 44
         PRINT 48 $ NBAD=1 $ RETURN
C
42       DO 43 KA=1,3
43       TITLE(KA)=D(KA+1)
         GOTO 46
44       DO 45 KA=1,4
45       PLABL(KA)=D(KA+1)
46       CONTINUE
C
     47  READ 35,(D(J),J=1,8)
         PRINT 36,(D(J),J=1,8)
         IF(D(1).EQ.AMB)GOTO 55
         IF(D(1).EQ.CHA)GOTO 65
         IF(D(1).EQ.PLT) GOTO 66
         IF(D(1).EQ.SCAL) GOTO 68
         IF(D(1).EQ.BLANK) GOTO 69
    475  PRINT 48 $ NBAD=2 $ RETURN
     48  FORMAT(1H0,10X,13HINVALID INPUT)
C
     55  IF(NAMB.EQ.1)GOTO 475
C    ONLY ONE AMBIENT DATA CARD WILL BE CONSIDERED
         NAMB=1
         IF(D(3).GT.0.)AIRPR=D(3)  $  IF(D(4).GT.0.)AIRTEM=D(4)
         IF(D(5).GT.0.)AIRGAM=D(5)  $  IF(D(6).GT.0.)AIRMOL=D(6)
C    IF INPUT IS ZERO THEN USE AIR DEFAULT VALUES
         DO 57 KA=1,4  $  AMSTAR(KA)=1H
         IF(D(KA+2).GT.0.) GOTO 57
         AMSTAR(KA)=1H*  $  NAMSTAR=1
57       CONTINUE
         AIRDEN=(AIRMOL/8.3143)*(AIRPR/AIRTEM)
         GOTO 47
C
     65  IF(NCHA.EQ.1)GOTO 475
         CHARVO=D(3) $ CHAREN=D(4)
         CHARHI=D(5) $ CHARHER=D(6)
         NCHA=1
         GOTO 47
C
66       DO 67 KA=1,6
67       PD(KA)=D(KA+2)
         GOTO 47
C    PLOTTING DATA CARD SPECIFIES PLOTTED OUTPUT
C    PD(1)= ERROR FACTOR FOR PRESSURE HISTORIES
C    PD(2)= ERROR FACTOR FOR OTHER FLOW HISTORIES
C
68       NSCAL=1
         SCD=D(3)  $  SCP=D(4)  $  SCT=D(5)
C    SCALE CARD OVERRIDES SCALES COMPUTED FROM AMBIENT AND CHARGE DATA
         IF(SCD.GT.0..AND.SCP.GT.0..AND.SCT.GT.0.) GOTO 47
         NSCAL=0  $  PRINT 681
681       FORMAT(1H ,10X,36HNON-POSITIVE SCALES ARE NOT ACCEPTED)
         GOTO 47
C
69       IF(NCHA.EQ.0.OR.NAMB.EQ.0) PRINT 70
70       FORMAT(1H0,10X,16HINCOMPLETE INPUT)
     75  PRINT106,(TITLE(J),J=1,3)
```

```
 106   FORMAT(1H1,/,1H ,10X,5HEVENT,/,1H ,10X, 5(1H-),/,1H0,15X,3A10,//)
       PRINT 107
   107 FORMAT(1H0,10X,18HAMBIENT CONDITIONS,/,1H ,10X,18(1H-),/)
       IF(NAMB.EQ.0) PRINT 1071
 1071  FORMAT(1H0,10X,36HTHE FOLLOWING AMBIENT CONDITIONS ARE,
      A /,1H ,10X,27HSTANDARD AIR DEFAULT VALUES,/)
       PRINT 108,AMSTAR(1),AIRPR,AMSTAR(2),AIRTEM,AMSTAR(3),AIRGAM,
      A AMSTAR(4),AIRMOL
   108 FORMAT(1H ,13X,A1,1X,8HPRESSURE,11X,7HAIRPR =,1PE12.5,4H  PA,/,
      A 1H ,13X,A1,1X,11HTEMPERATURE,8X,7HAIRTEM=,1PE12.5,3H  K,/,
      B 1H ,13X,A1,1X,16HSPEC. HEAT RATIO,3X,7HAIRGAM=,1PE12.5,/,
      C 1H ,13X,A1,1X,10HMOLAR MASS,9X,7HAIRMOL=,1PE12.5,9H  KG/MOLE,/)
       AIRSND=SQRT(AIRGAM*AIRPR/AIRDEN)
       PRINT 109,AIRSND,AIRDEN
   109 FORMAT(1H ,15X,11HSOUND SPEED,8X,7HAIRSND=,1PE12.5,5H  M/S,/,
      A 1H ,15X,7HDENSITY,12X,7HAIRDEN=,1PE12.5,9H  KG/M**3,/)
       IF(NAMSTAR.EQ.1) PRINT 1081
 1081  FORMAT(1H ,13X,35H* THE STARRED DATA ARE STANDARD AIR,
      A 15H DEFAULT VALUES,/)

       IF(NCHA.EQ.1) GOTO 1100
       NBAD=4  $  PRINT 1101,NBAD  $  RETURN
 1101  FORMAT(1H0,10X,29HRETURN FROM READAM WITH NBAD=,I2,
      A 33H, BECAUSE CHARGE DATA ARE MISSING)
C
 1100  PRINT 110
   110 FORMAT(1H0,10X,18HCHARGE DESCRIPTION,/,1H ,10X,18(1H-),/)
       PRINT 111, CHARVO,CHAREN
   111 FORMAT(1H ,15X,13HCHARGE VOLUME,6X,7HCHARVO=,1PE12.5,6H  M**3,/,
      A 1H ,15X,13HCHARGE ENERGY,6X,7HCHAREN=,1PE12.5,3H  J,/)
       SCDIST=CHARVO**(1./3.)
       PRINT 1110,CHARHI,CHARHER
  1110 FORMAT(1H ,15X,16HCHARGE ELEVATION,3X,7HCHARHI=,1PE12.5,4H +- ,
      A 1PE12.5,3H  M,/)
       SCTIME=SCDIST/AIRSND
       SCPRES=AIRPR
       SCEVEN=CHAREN/(CHARVO*AIRPR)
       PRINT 112
   112 FORMAT(1H0,10X,7HSCALING,/,1H ,10X,7(1H-),/)
       PRINT 113,SCDIST,SCTIME,SCPRES,SCEVEN
   113 FORMAT(1H ,15X,12HLENGTH SCALE,4X,20HSCDIST=CHARVO**(1/3),
      A 2X,1H=,1PE12.5,3H  M,/,
      B 1H ,15X,10HTIME SCALE,6X,20HSCTIME=SCDIST/AIRSND,
      C 2X,1H=,1PE12.5,3H  S,/,
      D 1H ,15X,14HPRESSURE SCALE,2X,13HSCPRES=AIRPR ,
      E 9X,1H=,1PE12.5,4H  PA,/,
      F 1H ,15X,14HSCALE OF EVENT,2X,21HCHAREN/(CHARVO*AIRPR),
      G 1X,1H=,1PE12.5,/)
       IF(SCEVEN.EQ.0.0)PRINT 114
   114 FORMAT(1H ,15X,30HEVENT CANNOT BE SCALED BECAUSE,
      A29H CHAREN IS NOT GIVEN BY INPUT,/)

       IF(NSCAL.EQ.0) GOTO 115
C  USE SCALES FROM SCALE CARD IF SUCH A CARD WAS READ
       SCDIST=SCD  $  SCPRES=SCP  $  SCTIME=SCT

 115   PRINT 116,SCDIST,SCTIME,SCPRES
```

```
116 FORMAT(1H ,////,1H ,10X,27HSCALES USED IN THIS PROGRAM,/,
   A 1H ,10X,27(1H-),//,1H ,20X,16HLENGTH SCALE    =,1PE12.5,3H  M,/,
   B 1H ,20X,16HTIME SCALE      =,1PE12.5,3H  S,/,
   C 1H ,20X,16HPRESSURE SCALE =,1PE12.5,4H  PA)
    NBAD=0
    RETURN
    END
```

```
      SUBROUTINE READSP(NBAD)
C
C  THIS ROUTINE READS SHOCK PARAMETERS NAD THEIR ACCURACIES
C
      COMMON/COMSHK/NPS,PAR(4),VPAR(4,4),SCD,SCP,SCT
      COMMON/CF2DER/GAMCAP,SNDSPD,CFPAR(4),ALOW,CFSCD,CFSCP,CFSCT
      COMMON/AMBCHA/AMP,AMT,AMG,AMM,      AMCHV,AMCHE,AMCHH,AMCHHE
C
      DIMENSION DAT(8),ER(4),COR(4,4)
      DIMENSION DSI(4),DSC(4),DPR(4)
C
      DATA(PL=10HSHOCKPAR  ),(EL=10HSHOCKPARER),(CL=10HSHOCKPARCO),
     A (SC=10HSHOCKSCALE),(BL=10H          )
C
      DATA DSI/10HPA*M      ,10HPA*M**2   ,10HPA*M**3   ,
     A 10HS        /
      DATA DSC/10HSCP*SCD   ,10HSCP*SCD**2,10HSCP*SCD**3,
     A 10HSCT       /

      KPL=1  $  KEL=1  $  KCL=1  $  KSC=1
      PRINT 12
12    FORMAT(1H1,10X,20HINPUT READ BY READSP,/)
15    FORMAT(2A10,6E10.3)
25    FORMAT(1H ,5X,2A10,6(2X,1PE14.7))
35    READ 15,(DAT(J),J=1,8)
      PRINT 25,(DAT(J),J=1,8)
      IF(DAT(1).EQ.PL) GOTO 55
      IF(DAT(1).EQ.EL) GOTO 75
      IF(DAT(1).EQ.CL) GOTO 95
      IF(DAT(1).EQ.SC) GOTO 115
      IF(DAT(1).EQ.BL) GOTO 125
      NBAD=1
      PRINT 45  $  RETURN
45    FORMAT(1H0,10X,13HINVALID INPUT)
C
55    DO 65 KA=1,4
65    PAR(KA)=DAT(KA+2)
      DALOW=DAT(7)
      IF(DALOW.GE.1.0E-90) GOTO 67
      PRINT 66,DAT(6)
66     FORMAT(1H ,10X,'5-TH NUMBER ON PREVIOUS CARD SHOULD BE '
     A 'POSITIVE INDICATING SHOCK DISTANCE AT T='1PE12.5)
      NBAD=66  $  PRINT 45
      RETURN
67    CONTINUE
      KPL=0
      GOTO 35
C
75    DO 85 KA=1,4
85    ER(KA)=DAT(KA+2)
      KEL=0
      GOTO 35
C
95    COR(1,1)=1.  $  COR(2,2)=1.  $  COR(3,3)=1.  $  COR(4,4)=1.
      COR(1,2)=DAT(3)  $  COR(2,1)=COR(1,2)
      COR(1,3)=DAT(4)  $  COR(3,1)=COR(1,3)
      COR(1,4)=DAT(5)  $  COR(4,1)=COR(1,4)
```

```
      COR(2,3)=DAT(6)  $  COR(3,2)=COR(2,3)
      COR(2,4)=DAT(7)  $  COR(4,2)=COR(2,4)
      COR(3,4)=DAT(8)  $  COR(4,3)=COR(3,4)
      KCL=0
      GOTO 35
C
115   SCD=DAT(3)  $  SCP=DAT(4)  $  SCT=DAT(5)
      KSC=0
      GOTO 35
C
125   IF(KPL.EQ.0.AND.KEL.EQ.0.AND.KCL.EQ.0.AND.KSC.EQ.0)GOTO 145
      NBAD=2
      PRINT 135  $  RETURN
135   FORMAT(1H0,10X,16HINCOMPLETE INPUT)
C
145   NPS=4
      ALOW=DALOW*SCD
      GAMCAP=((1.+AMG)/(2.*AMG))/AMP
      SNDSPD=SQRT(AMG*AMT*(8.3143/AMM))
      CFSCD=1.  $  CFSCP=1.  $  CFSCT=1.
C  /CF2DER/ IS NEEDED FOR SHOCK ARRIVAL TIME COMPUTATIONS
      DO 155 KA=1,4  $  DO 155 KB=1,4
155   VPAR(KA,KB)=ER(KA)*COR(KA,KB)*ER(KB)
      NBAD=0
      PRINT 165
  165 FORMAT(1H0,12X,16HSHOCK PARAMETERS,4X,6HERRORS,5X,
     A 10HDIMENSIONS,/)
      IF(SCD.EQ.1..AND.SCP.EQ.1..AND.SCT.EQ.1.) GOTO 167
      DO 166 KA=1,4
  166 DPR(KA)=DSC(KA)
      DISDI=10HSCD
      GOTO 169
  167 DO 168 KA=1,4
  168 DPR(KA)=DSI(KA)
      DISDI=10HMETRES
  169 PRINT 175,((PAR(J),ER(J),DPR(J)),J=1,4)
  175 FORMAT(1H ,14X,1PE12.5,4X,1PE10.3,2X,A10)
      PRINT 178,DALOW,DISDI
  178 FORMAT(1H0,10X,43HTHE LAST PARAMETER IS SHOCK ARRIVAL TIME AT,
     A 2X,1PE12.5,2X,A10)
      PRINT 185
185   FORMAT(1H ,///,1H ,15X,*SHOCK PARAMETER CORRELATION MATRIX*,/)
      PRINT 195,((COR(J,K),K=1,4),J=1,4)
195   FORMAT(4(1H ,10X,4(2X,0PF10.7),/))
      PRINT 205
205   FORMAT(1H ,///,1H ,15X,16HSHOCK PARAMETER ,
     A 26HVARIANCE-COVARIANCE MATRIX,/)
      PRINT 215,((VPAR(J,K),K=1,4),J=1,4)
215   FORMAT(4(1H ,10X,4(2X,1PE12.5),/))
      PRINT 225
225   FORMAT(1H ,///,1H ,16X,22HSHOCK PARAMETER SCALES,/)
      PRINT 235,SCD,SCP,SCT
  235 FORMAT(1H ,15X,12HLENGTH SCALE,4X,5HSCD =,1PE12.5,3H  M,/,
     A 1H ,15X,14HPRESSURE SCALE,2X,5HSCP =,1PE12.5,4H  PA,/,
     B 1H ,15X,10HTIME SCALE,6X,5HSCT =,1PE12.5,3H  S)
      RETURN
      END
```

```
      SUBROUTINE READPR(NRPR)
C  THIS READS PRESSURE HISTORIES FROM CARDS
C
      COMMON/AMBCHA/APR,ATE,AGA,AMO,CVO,CEN,CHI,CHIER
      COMMON/COMPR/TP(2,5000),ERTP(2,5000),ALB(2,5000),NSET(50),
     1 DIST(50),ERDIST(50)
      LEVEL 2,TP,ERTP,ALB,NSET,DIST,ERDIST
      DIMENSION D(8)
      DATA (TIMPRE=10HTIME,PRES ),(RANGEL=10HRANGE,ELEV)
     A,(BLANK=10H          )
C
      PRINT 8
    8 FORMAT(1H1,10X,20HINPUT READ BY READPR,/)
      NRPR=0
    9 FORMAT(2A10,6(E10.3))
   10 FORMAT(1H ,5X,2A10,6(2X,1PE12.5))
   12 READ 9,(D(J),J=1,6)
      PRINT 10,(D(J),J=1,6)
      IF(D(1).EQ.BLANK) GOTO 15
      IF(D(2).EQ.TIMPRE) GOTO 35
      IF(D(2).EQ.RANGEL) GOTO 55
C
   15 IF(NRPR.EQ.0) RETURN
      PRINT 18,DIST(NRPR),ERDIST(NRPR)
      PRINT 17,NRPR,NSET(NRPR)
      IF(DIST(NRPR).GT.0.) GOTO 16
      PRINT 40,ALB(1,NRST)
      NRPR=NRPR-1  $  NRST=NRST-KST  $  KST=0
   16 CONTINUE
      RETURN
   17 FORMAT(1H ,5X,20HNUMBER OF SETS NSET(,I3,2H)=,I4,/)
   18 FORMAT(1H0,5X,10HDISTANCE =,1PE12.5,4H +- ,1PE9.2)
C
   35 IF(NRPR.GT.0) GOTO 39
      NRPR=1  $  KST=0  $  NRST=1
      DIST(NRPR)=0.  $  ERDIST(NRPR)=0.
      GOTO 45

   39 IF(D(1).EQ.ALB(1,NRST)) GOTO 45
      PRINT 18,DIST(NRPR),ERDIST(NRPR)
      PRINT 17,NRPR,NSET(NRPR)

      IF(DIST(NRPR).GT.0.) GOTO 41
      PRINT 40,ALB(1,NRST)
      NRPR=NRPR-1  $  NRST=NRST-KST  $  KST=0
      GOTO 12
   40 FORMAT(1H ,5X,29HPREVIOUS DATA SET WITH LABEL ,A10,
     A 46H NOT ACCEPTED BECAUSE DISTANCE CARD IS MISSING,/)
   41 IF(NSET(NRPR).GT.3) GOTO 43
      PRINT 42,ALB(1,NRPR)
      NRPR=NRPR-1  $  NRST=NRST-KST  $  KST=0
      GOTO 12
   42 FORMAT(1H ,5X,29HPREVIOUS DATA SET WITH LABEL ,A10,
     A 41H NOT ACCEPTED BECAUSE NUMBER OF DATA SETS,
     A 18H IS LESS THAN FOUR,/)
```

```
 43     CONTINUE
        NRPR=NRPR+1 $ KST=0 $ NRST=NRST+1
 45     IF(KST.GT.0) NRST=NRST+1
        KST=KST+1
        ALB(1,NRST)=D(1)
 47     FORMAT(5H PT. ,I4,1H )
        ENCODE(10,47,ALB(2,NRST))KST
        TP(1,NRST)=D(3)  $  ERTP(1,NRST)=D(4)
        TP(2,NRST)=D(5)  $  ERTP(2,NRST)=D(6)
        NSET(NRPR)=KST
        GOTO 12
C
 55     IF(D(3).GT.0..AND.D(4).GT.0.) GOTO 57
        PRINT 56
        GOTO 12
 56     FORMAT(1H ,5X,38HCARD NOT ACCEPTED BECAUSE DISTANCE OR,
       A 22H ERROR IS NOT POSITIVE,/)
 57     IF(NRPR.GT.0 ) GOTO 59
        NRPR=1  $  KST=0  $  NRST=1
        DIST(NRPR)=0.  $  ERDIST(NRPR)=0.
        GOTO 65
 59     IF(D(1).EQ.ALB(1,NRST)) GOTO 70
        PRINT 18,DIST(NRPR),ERDIST(NRPR)
        PRINT 17,NRPR,NSET(NRPR)
        IF(DIST(NRPR).GT.0.) GOTO 61
        PRINT 40,ALB(1,NRST)
        NRPR=NRPR-1  $  NRST=NRST-KST  $  KST=0
 61     CONTINUE
        NRPR=NRPR+1  $  KST=0
        NRST=NRST+1
 65     ALB(1,NRST)=D(1)
 70     DSQ=D(3)**2+(CHI-D(5))**2
         DIST(NRPR)=SQRT(DSQ)
        ERSQ=(D(3)*D(4))**2/DSQ+(CHI-D(5))**2*(D(4)**2+D(6)**2)/DSQ
        ERDIST(NRPR)=SQRT(ERSQ)
        GOTO 12
        END
```

```
      SUBROUTINE SCALPR(SCDIST,SCPRES,SCTIME,NRCASE,
     AX,R,ALAB,LSTX,NRSETS,TIMSH,PRSH,DISH,NBAD)
C  THIS ROUTINE TAKES PROFILE DATA FROM COMPR AND STORES THEM
C  IN ARRAYS X, 1 THROUGH NRSETS, FOR ADJUSTMENT BY COLSAC
C  THE DATA ARE ALSO SCALED USING THE SCALES IN ARGUMENT LIST
C  USES SUBROUTINE SHOCK TO COMPUTE SHOCK VALUES AT PROFILE DISTANCE
C
      LEVEL 2,X,R,ALAB,LSTX
      DIMENSION X(5,100),R(5,5,100),ALAB(2,100),LSTX(100)
      COMMON/COMPR/TPPR(2,5000),ERTPPR(2,5000),ALBPR(2,5000),
     1 NSETPR(50),DISTPR(50),ERDIPR(50)
      LEVEL 2,TPPR,ERTPPR,ALBPR,NSETPR,DISTPR,ERDIPR
      NBAD=0
      NRSETS=NSETPR(NRCASE) $ IF(NRSETS.LE.0)GOTO 45
      KIN=1 $ IF(NRCASE.EQ.1)GOTO 25
      DO 15 KA=2,NRCASE
   15 KIN=KIN+NSETPR(KA-1)
   25 KEN=KIN+NSETPR(NRCASE)-1 $ KST=0
C
      DO 35 KA=KIN,KEN
      KST=KST+1
      X(1,KST)=TPPR(1,KA)/SCTIME
      X(2,KST)=TPPR(2,KA)/SCPRES
      R(1,1,KST)=(ERTPPR(1,KA)/SCTIME)**2
      R(2,2,KST)=(ERTPPR(2,KA)/SCPRES)**2
      R(1,2,KST)=0 $ R(2,1,KST)=0 $ LSTX(KST)=0
      ALAB(1,KST)=ALBPR(1,KA) $ ALAB(2,KST)=ALBPR(2,KA)
   35 CONTINUE
C
      DS=DISTPR(NRCASE)
      CALL SHOCK(DS,TS,PSOV,US,UP,RHO,NBAD)
      IF(NBAD.NE.0)RETURN
C  SHOCK RESULTS ARE IN SI UNITS.  SCALE THE OUTPUT ACCORDING TO
C  SCALES IN THE ARGUMENT LIST.
      TIMSH=TS/SCTIME
      PRSH=PSOV/SCPRES
      DISH=DS/SCDIST
      RETURN
   45 NBAD=1 $ RETURN
      END
```

```
      SUBROUTINE FITPR(X,R,ALAB,LSTX,NRSET,TIMSH,PRSH,DISH,PAR,VPAR,ERZ
     A TITLE,SCDIS,SCPRES,SCTIME,NBAD)
C  THIS FITS THE ONE PRESSURE HISTORY WHICH IS STORED IN X
C  THE SUBROUTINE IS CALLED FROM MAIN AFTER THE DATA HAVE BEEN PREPARED
C  BY CALLING SCALPR
C
      LEVEL 2,X,R,ALAB,LSTX,XC,C,LSTN,WORK
      COMMON/SCRCH/XC(5,100),C(5,100),LSTN(100),WORK(12560)
C
      DIMENSION PAR(10),VPAR(10,10),ERP(10),V(10,10),TITLE(3)
      DIMENSION X(5,100),R(5,5,100),ALAB(2,100),LSTX(100)
      DIMENSION PPR(10)
C
      COMMON/PSTS/PS,TS
C
      EXTERNAL EXPON
C
      NXD=5  $  NPD=10  $  NW=12560
      PS=PRSH
      TS=TIMSH
C  STORE SHOCK OVERPRESSURE AND ARRIVAL TIME IN COMMON /PSTS/
C  COMMON /PSTS/ IS USED BY THE CONSTRAINT SUBROUTINE EXPON
      CALL GUESS(X,PPR,NRSET,TIMSH,PRSH)
C  GUESS COMPUTES INITIAL ESTIMATES OF PRESSURE PROFILE PARAMETERS
      DO 15 KP=1,10
15    PAR(KP)=PPR(KP)
      NR=NRSET
      NX=2
      NP=3  $  ITYPE=0
      IF(NRSET.LT.3) GOTO 37
      CALLCOLSACA(X,R,ALAB,LSTX,NX,NR,PAR,NP,EXPON,ITYPE,XC,C,LSTN,NRGD,
     1 ERZ,VPAR,ERP,LBAD,NXD,NPD,WORK,NW)
      IF(LBAD.EQ.0) GOTO 45
C  SUBSEQUENT CALLS TO COLSACA ARE EXECUTED ONLY IN CASE OF
C  CONVERGENCE PROBLEMS
C
      DO 25 KP=1,10
25    PAR(KP)=PPR(KP)
      NP=2  $  PAR(3)=0
      ITYPE=4
      CALLCOLSACA(X,R,ALAB,LSTX,NX,NR,PAR,NP,EXPON,ITYPE,XC,C,LSTN,NRGD,
     1 ERZ,VPAR,ERP,LBAD,NXD,NPD,WORK,NW)
      NP=3
      ITYPE=1
      CALLCOLSACA(X,R,ALAB,LSTX,NX,NR,PAR,NP,EXPON,ITYPE,XC,C,LSTN,NRGD,
     1 ERZ,VPAR,ERP,LBAD,NXD,NPD,WORK,NW)
      IF(LBAD.EQ.0) GOTO 45
      ITYPE=4
      DO 35 KP=1,10
35    PAR(KP)=PPR(KP)
37    CALLCOLSACA(X,R,ALAB,LSTX,NX,NR,PAR,NP,EXPON,ITYPE,XC,C,LSTN,NRGD,
     1 ERZ,VPAR,ERP,LBAD,NXD,NPD,WORK,NW)
      IF(LBAD.EQ.25) ITYPE=1
      IF(LBAD.EQ.25)
     ACALLCOLSACA(X,R,ALAB,LSTX,NX,NR,PAR,NP,EXPON,ITYPE,XC,C,LSTN,NRGD,
     1 ERZ,VPAR,ERP,LBAD,NXD,NPD,WORK,NW)
C
```

```
C  NEXT PRINT THE RESULTS OF FITTING
45    CALL PRTPNTS(X,R,ALAB,XC,C,NRSET,TIMSH,PRSH,DISH,TITLE,
     A SCDIS,SCPRES,SCTIME)
      NBAD=LBAD
      RETURN
      END
```

```
      SUBROUTINE GUESS(X,PAR,NR,TS,PS)
C  THIS ESTABLISHES INITIAL APPROXIMATIONS OF PAR
C  X       = TIME AND OVERPRESSURE
C  PAR     = MODEL PARAMETERS A,B,C IN THE MOPEL
C                P=-C+(PS+C)*EXP(A*TAU+B*TAU**2),
C                TAU=T-TS.    PAR IS OUTPUT FOR THIS ROUTINE
C  NR      = NUMBER OF DATA POINTS
C  PS,TS   = SHOCK OVERPRESSURE AND ARRIVAL TIME

      DIMENSION X(5,100),PAR(10)
      LEVEL 2,X

      COMMON/GUECM/AN(3,3),RS(3),W(18)
      DOUBLE PRECISION AN,RS,W,DET
      LEVEL 2,AN,RS,W

      IF(NR.GT.3)GOTO 25
      PRINT 15,NR
      RETURN
   15 FORMAT(1H0,40(1H*),/,1H ,10X,12HERROR RETURN,
     A35H FROM SUBROUTINE GUESS BECAUSE NR =,I3,
     B28H IS TOO SMALL FOR ADJUSTMENT,/,1H0,40(1H*))

   25 PMIN=PS
      DO 35 KA=1,NR
      PMIN=AMIN1(PMIN,X(2,KA))
   35 CONTINUE
C  THIS ESTABLISHED LOWEST VALUE OF OVERPRESSURE

      CMIN=-PS*0.5
      CMAX=AMIN1(0.,PMIN-PS*0.05)
      C=CMAX
C  INITIAL GUESS FOR PARAMETER C
      IF(CMIN.LT.CMAX)GOTO 55
      PRINT 45,PS,PMIN
      RETURN
   45 FORMAT(1H0,40(1H*),/,1H ,10X,17HERROR RETURN FROM,
     A30H SUBROUTINE GUESS BECAUSE PS =,1PE12.5,
     B12H  AND PMIN =,1PE12.5,/,1H ,40(1H*))

   55 KIT=0
C  KIT IS ITERATION COUNTER
      NX=3 $ NA=3 $ KIN=1
C  NEXT ESTABLISH NORMAL EQS FOR SIMPLIFIED PROBLEM

   56 DO 75 KA=1,3$ DO 65 KB=1,3
   65 AN(KA,KB)=0
   75 RS(KA)=0

      DO 85 KA=1,NR
      TAU=X(1,KA)-TS
      RO=(PS-X(2,KA))/((PS-C)*(X(2,KA)-C))
      AL=ALOG((X(2,KA)-C)/(PS-C))
      WE=(X(2,KA)-C)**2
      AN(1,1)=AN(1,1)+WE*TAU**2  $  AN(1,2)=AN(1,2)+WE*TAU**3
      AN(1,3)=AN(1,3)+WE*RO*TAU     $ AN(2,2)=AN(2,2)+WE*TAU**4
      AN(2,3)=AN(2,3)+WE*RO*TAU**2  $  AN(3,3)=AN(3,3)+WE*RO
```

```
      RS(1)=RS(1)+WE*TAU*AL
      RS(2)=RS(2)+WE*TAU**2*AL
      RS(3)=RS(3)+WE*RO*AL
   85 CONTINUE

      AN(2,1)=AN(1,2) $ AN(3,1)=AN(1,3) $ AN(3,2)=AN(2,3)

      CALL MTRINDB(AN,NX,RS,NA,KIN,DET,W)
C  THIS SOLVED THE NORMAL EQUATIONS
      IF(NX.EQ.2.OR.DET.NE.0.) GOTO 95
      NX=2  $  NA=3  $  KIN=1
      GOTO 56
95    CONTINUE
      EPS=RS(3)  $  IF(NX.EQ.2) EPS=0.
      C=AMAX1(CMIN,AMIN1(C+EPS,CMAX))
      KIT=KIT+1
      NX=3  $  NA=3  $  KIN=1
      IF(KIT.LT.4) GOTO 56
C  ITERATE THREE TIMES

      PAR(1)=RS(1) $ PAR(2)=RS(2) $ PAR(3)=-C
      RETURN
      END
```

```
      SUBROUTINE EXPON(X,KA,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  CONSTRAINT FOR 3-PARAMETER PRESSURE HISTORY FITTING BY FITPR
C        F = (PS+C)*EXP(A*TAU+B*TAU**2)-C-P,    TAU=T-TS
C       T=X(1)  $  P=X(2)
C
      LEVEL 2,X,FX,FP,FXX,FXP,FPP
      DIMENSION X(5,100),PAR(10),FX(5),FP(10),FXX(5,5),FXP(5,10),
     1 FPP(10,10)
      COMMON/PSTS/ PS,TS
C
      NBAD=0
      A=PAR(1)+2.*PAR(2)*(X(1,KA)-TS)
      B=X(1,KA)-TS
      ARG=(PAR(2)*B+PAR(1))*B
      IF(ARG.LT.700.) GOTO 15
      NBAD=1  $  RETURN
15    IF(ARG.LT.-650.) EXPQ=0.
      IF(ARG.GE.-650.)EXPQ=EXP(ARG)
C  THIS AVOIDS OVERFLOW OR UNDERFLOW IN THE EXP ROUTINE
      PC=PS+PAR(3)
      PEX=PC*EXPQ
      F=PEX-PAR(3)-X(2,KA)
      FX(1)=A*PEX
      FX(2)=-1.
      FP(1)=B*PEX
      FP(2)=B**2*PEX
      FP(3)=EXPQ-1.
C   SECOND DERIVATIVES
      FXX(1,1)=PEX*(2.*PAR(2)+A**2)
      FXX(1,2)=0.0
      FXX(2,1)=0.0
      FXX(2,2)=0.0
      FXP(1,1)=PEX*(1.+B*A)
      FXP(2,1)=0.0
      FXP(1,2)=PEX*(2.*B+B**2*A)
      FXP(2,2)=0.0
      FXP(1,3)=EXPQ*A
      FXP(2,3)=0.0
      FPP(1,1)=PEX*B**2
      FPP(1,2)=PEX*B**3
      FPP(2,1)=FPP(1,2)
      FPP(1,3)=EXPQ*B  $ FPP(3,1)=FPP(1,3)
      FPP(2,2)=PEX*B**4
      FPP(2,3)=FPP(1,3)*B  $  FPP(3,2)=FPP(2,3)
      FPP(3,3)=0
      RETURN
      END
```

```
      SUBROUTINE PRTPNTS(X,R,ALAB,XC,C,NR,TS,PS,DS,TITLE,
     A SCDIS,SCPRES,SCTIME)
C  THIS IS CALLED FROM FITPR TO PRINT THE SINGLE HISTORY
C  ADJUSTMENT RESULTS
C
      DIMENSION X(5,100),R(5,5,100),ALAB(2,100),XC(5,100)
      DIMENSION C(5,100),TITLE(3)

      LEVEL 2,X,R,ALAB,XC,C

      NSI=1
      HTXD=3HM  $ HTXP=3HPA $ HTXT=3HS
      TXT=5H (S) $ TXP=5H(PA)
      IF(SCDIS.EQ.1..AND.SCPRES.EQ.1..AND.SCTIME.EQ.1.) GOTO 5
      NSI=0
C  NSI=0 INDICATES THAT COMPUTATION IS NOT IN SI UNITS
      HTXD=3HSCD $ HTXP=3HSCP $ HTXT=3HSCT
      TXT=5H(SCT) $ TXP=5H(SCP)
    5 DO 100 J=1,NR
      IF(MOD(J,40).NE.1) GOTO 45

      PRINT 10,(TITLE(K),K=1,3),DS,HTXD,PS,HTXP,TS,HTXT
   10 FORMAT(1H1,5X,5HEVENT,5X,3A10,45X,21HHISTORY DISTANCE    = ,
     A 1PE10.3,2X,A3,/,1H ,5X,5(1H-),80X,21HSHOCK OVERPRESSURE = ,
     B 1PE10.3,2X,A3,/,1H ,90X,21HSHOCK ARRIVAL TIME = ,
     C 1PE10.3,2X,A3,/)
      PRINT 20
   20 FORMAT(1H ,24X,43HADJUSTMENT OF A SINGLE OVERPRESSURE HISTORY,/)
      PRINT 30,TXT,TXT,TXT,TXT,TXP,TXP,TXP,TXP
   30 FORMAT(1H ,8X,6HLABELS,14X,4HTIME,7X,9HSTD.ERROR,3X,
     A 10HCORRECTION,4X,9HCORR.TIME,2X,12HOVERPRESSURE,3X,
     B 9HSTD.ERROR,3X,10HCORRECTION,4X,10HCORR.OVPR.,/,
     C 1H ,22X,8(6X,A5,2X),/)
   40 FORMAT(1H )

   45 R1=SQRT(R(1,1,J))
      R2=SQRT(R(2,2,J))
      PRINT 50,   ALAB(1,J),ALAB(2,J),X(1,J),R1,C(1,J),XC(1,J),X(2,J),
     1 R2,C(2,J),XC(2,J)
   50 FORMAT(1H ,2X,2A10,1P,8(3X,E10.3))
   75 IF((J/5)*5.EQ.J) PRINT 40

      IF(J.NE.NR.AND.MOD(J,40).NE.0.)GOTO 100
      IF(NSI.EQ.1) GOTO 100

C  PRINT SCALES IF SI-SCALES WERE NOT USED
      PRINT 115,SCDIS,SCPRES,SCTIME
  115 FORMAT(1H ,/,1H ,21X,31HTHE DATA ARE SCALED AS FOLLOWS:,5X,
     A 16HDISTANCE  SCD = ,1PE12.5,3H  M,/,1H ,57X,
     B 16HPRESSURE  SCP = ,1PE12.5,4H  PA,/,1H ,57X,
     C 16HTIME      SCT = ,1PE12.5,3H  S)

100   CONTINUE

      IF(MOD(NR,40).GT.30) PRINT 55
   55 FORMAT(1H1)
      RETURN
      END
```

```
      SUBROUTINE DIMPAR(SCDIS,SCPRES,SCTIME,P,VP,ERZ,PDIM,VPDIM)
C  THIS COMPUTES DIMENSIONAL VALUES OF PRESSURE PROFILE PARAMETERS
C  IT IS CALLED FROM MAIN AFTER A PROFILE ADJUSTMANT BY FITPR
C
      DIMENSION P(10),VP(10,10),PDIM(10),VPDIM(10,10)
      DIMENSION SCMAT(10,10)
      DO 15 KA=1,10  $  DO 15 KB=1,10
15    SCMAT(KA,KB)=0
      SCMAT(1,1)=1./SCTIME  $  SCMAT(2,2)=1./SCTIME**2
      SCMAT(3,3)=SCPRES
C
      DO 45 KA=1,3  $  PDIM(KA)=0
      DO 35 KB=1,3  $  VPDIM(KA,KB)=0
      DO 25 KC=1,3  $  DO 25 KD=1,3
25    VPDIM(KA,KB)=VPDIM(KA,KB)+SCMAT(KA,KC)*VP(KC,KD)*SCMAT(KB,KD)
35    PDIM(KA)=PDIM(KA)+SCMAT(KA,KB)*P(KB)
45    CONTINUE
C
      PRINT 55
55    FORMAT(1H0,///,1H ,10X,32HDIMENSIONAL VALUES OF PARAMETERS,/)
      PRINT 65
65    FORMAT(1H0,10X,10HPARAMETERS,5X,8HSTANDARD,7X,8HSTANDARD,
     A 5X,9HDIMENSION,/,1H ,26X,6HERRORS,7X,10HERRORS*ERZ,/)
      PER=SQRT(VPDIM(1,1))  $  PERZ=PER*ERZ
      PRINT 75,PDIM(1),PER,PERZ
75    FORMAT(1H ,9X,1PE12.5,3X,1P E10.3,4X,1PE10.3,6X,3H1/S)
      PER=SQRT(VPDIM(2,2))  $  PERZ=PER*ERZ
      PRINT 85,PDIM(2),PER,PERZ
85    FORMAT(1H ,9X,1PE12.5,3X,1PE10.3,4X,1PE10.3,6X,6H1/S**2)
      PER=SQRT(VPDIM(3,3))  $  PERZ=PER*ERZ
      PRINT 95,PDIM(3),PER,PERZ
95    FORMAT(1H ,9X,1PE12.5,3X,1PE10.3,4X,1PE10.3,6X,2HPA)

      PRINT 105
105   FORMAT(1H ,////,1H ,20X,24HTHE OVERPRESSURE HISTORY,
     A 19H IS APPROXIMATED BY,///,
     B 1H ,30X,7HP(T) = ,35H-C + (PSHOCK+C)*EXP( A*(T-TSHOCK) +,
     C 19H B*(T-TSHOCK)**2 ),,//,
     D 1H ,30X,42HWHERE A, B AND C ARE THE THREE PARAMETERS.)

      RETURN
      END
```

```
      SUBROUTINE PLTPNTS(X,R,ALAB,NR,PSH,TSH,SCP,SCT,PAR,V,TITLE)
C  THIS ROUTINE PLOTS FITTED PRESSURE HISTORY AND CORRESPONDING  OBSERVAT
C  THE PLOTTING IS DONE IN SI UNITS

C  X(5,NR)        = TIME X(1, ) AND PRESSURE X(2, ) OBSERVED
C  R(5,5,NR)      = VARIANCE-COVARIANCE MATRIX OF OBSERVATIONS X
C  ALAB(2,NR)     = LABELS OF OBSERVATIONS
C  NR             = NUMBER OF OBSERVATIKNS
C  PSH          = SHOCK OVERPRESSURE  AT HISTORY GAGE LOCATION
C  TSH            = SHOCK ARRIVAL TIME AT HISTORY GAGE LOCATION
C  SCP,  SCT      = PRESSURE AND TIME SCALES, RESPECTIVELY, OF THE ABOVE
C  PAR(10)        = HISTORY FITTING PARAMETERS IN SI UNITS
C  V(10,10)       = VARIANCE-COVARIANCE MATRIX OF PAR
C  TITLE(3)       = NAME OF THE EVENT
C
      DIMENSION X(5,100),R(5,5,100),PAR(10),V(10,10),ALAB(2,100)
      DIMENSION Q(2,2)
      DIMENSION TEMP(8),TITLE(3),X1(200),Y1(200),Y2(200)
      DIMENSION X3(201),Y3(201),X4(201),Y4(201)
      LEVEL 2,FX,FP,FXX,FXP,FPP
      COMMON/SCRCHA/XP(5,1),FX(5),FP(10),FXX(5,5),FXP(5,10),FPP(10,10)
      COMMON/PSTS/PS,TS
      LEVEL 2,X,R,ALAB,XP
      COMMON/PLOT/ERF,D(5),PLABL(4)
C
      PS=PSH*SCP
      TS=TSH*SCT
      XMIN=X(1,1)*SCT   $ XMAX=XMIN
      DO 15 KA=2,NR
      XMIN=AMIN1(XMIN,X(1,KA)*SCT)
      XMAX=AMAX1(XMAX,X(1,KA)*SCT)
15    CONTINUE
      DELX=(XMAX-XMIN)/200.
      IF(ERF.EQ.0.0) ERF=2.0
C
C  NEXT COMPUTE 200 POINTS OF FITTED CURVE WITH CONFIDENCE LIMITS
      DO 200 I=1,200
      ES=0.0
      XP(1,1)=XMIN+DELX*I
      XP(2,1)=0.0
      CALL EXPON(XP,1,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  F IS OVERPRESSURE
      IF(NBAD.EQ.0) GOTO 139
      PRINT 134,NBAD
      PRINT 135,XP(1,1),(PAR(J),J=1,5)
      PRINT 138
      RETURN
134   FORMAT(1H ,10X,*ERROR RETURN FROM EXPON WITH NBAD=*,I5)
135   FORMAT(1H ,10X,*THE ARGUMENTS WERE XP(1,1)=*,1PE12.5,/
     A 1H ,10X,*PAR(J)=*,5(2X,1PE12.5))
138   FORMAT(1H ,10X,*ERROR RETURN FROM PLTPNTS*)
139   DO 150 KA=1,3
      DO 150 KB=1,3
      ES=ES+FP(KA)*V(KA,KB)*FP(KB)
  150 CONTINUE
      E=SQRT(ES)
C  E IS THE STANDARD ERRROR OF COMPUTED F (OVERPRESSURE)
```

```
C
      X1(I)=XP(1,1)
      Y1(I)=F+ERF*E
      Y2(I)=F-ERF*E
      X3(I+1)=XP(1,1)
      Y3(I+1)=F
  200 CONTINUE
      CALL PLTBEG(8.7,11.2,1.0,13,PLABL)
C
C  NEXT FIX SCALES AND PLOT AXES WITH LABELS
C
      XSIZE=5.0
      YSIZE=4.0
      X3(1)=X3(2)
      CALL FIXSCA(X1,200,XSIZE,XS,XMIN,XMAX,DX)
      CALL FIXSCA(Y1,200,YSIZE,YS,YMIN,YMAX,DY)
      CALL CONSCA(Y2,200,YSIZE,YS,YMIN,YMAX,DY)
      Q(1,1)=R(1,1,1)*SCT**2
      Q(1,2)=R(1,2,1)*SCT*SCP     $  Q(2,1)=Q(1,2)
      Q(2,2)=R(2,2,1)*SCP**2
      CALL ERELCM(X(1,1)*SCT,X(2,1)*SCP,Q,ERF,X4,Y4)
      CALL CONSCA(X4,201,XSIZE,XS,XMIN,XMAX,DX)
      CALL CONSCA(Y4,201,YSIZE,YS,YMIN,YMAX,DY)
      Q(1,1)=R(1,1,NR)*SCT**2
      Q(1,2)=R(1,2,NR)*SCT*SCP   $  Q(2,1)=Q(1,2)
      Q(2,2)=R(2,2,NR)*SCP**2
      CALL ERELCM(X(1,NR)*SCT,X(2,NR)*SCP,Q,ERF,X4,Y4)
      CALL CONSCA(X4,201,XSIZE,XS,XMIN,XMAX,DX)
      CALL CONSCA(Y4,201,YSIZE,YS,YMIN,YMAX,DY)
      Y3(1)=YMIN
      CALL PLTSCA(2.5,4.0,XMIN,YMIN,XS,YS)
      CALL PLTAXS(DX,DY,XMIN,XMAX,YMIN,YMAX,4)
      CALL LABAX(DX,2.0*DY,XMIN,XMAX,YMIN,YMAX)
      HT=0.1
      ENCODE(80,160,TEMP) ERF
  160 FORMAT(*FITTED CURVE WITH *,F3.1,* STANDARD ERRORS>*)
      TX=(XMAX+XMIN)*0.5-17.5*HT*XS
      TY=YMAX+0.5*YS
      CALL PLTSYM(HT,TEMP,0.0,TX,TY)
      ENCODE(80,110,TEMP)
  110 FORMAT(9HTIME (S)>)
      TX=(XMAX+XMIN)*0.5-4.0*HT*XS
      TY=YMIN-0.5*YS
      CALL PLTSYM(HT,TEMP,0.0,TX,TY)
      ENCODE(80,120,TEMP)
  120 FORMAT(18HOVERPRESSURE (PA)>)
      TX=XMIN-0.7*XS
      TY=(YMAX+YMIN)*0.5-9.0*HT*YS
      CALL PLTSYM(HT,TEMP,90.0,TX,TY)
C
C  NEXT PLOT CURVE WITH CONFIDENCE LIMITS
C
      CALL PLTDTS(1,0,X1,Y1,200,0)
      CALL PLTDTS(1,0,X1,Y2,200,0)
      CALL PLTDTS(1,0,X3,Y3,201,0)
C
C  NEXT PLOT ERROR ELLIPSES OF OBSERVATIONS
```

```
      C
            DO 250 I=1,NR
            X1(I)=X(1,I)*SCT
            Y1(I)=X(2,I)*SCP
            Q(1,1)=R(1,1,I)*SCT**2
            Q(1,2)=R(1,2,I)*SCT*SCP  $  Q(2,1)=Q(1,2)
            Q(2,2)=R(2,2,I)*SCP**2
            CALL ERELCM(X1(I),Y1(I),Q,ERF,X3,Y3)
            CALL PLTDTS(1,0,X3,Y3,201,0)
        250 CONTINUE
      C  THIS PLOTS OBSERVATIONS
            CALL PLTDTS(3,1,X1,Y1,NR,0)
            ENCODE(80,130,TEMP) ALAB(1,1)
        130 FORMAT(A10,1H>)
            TX=(XMAX+XMIN)*0.5-5.0*HT*XS
            TY=YMAX+0.75*YS
            CALL PLTSYM(HT,TEMP,0.0,TX,TY)
            ENCODE(80,140,TEMP)TITLE
        140 FORMAT(3A10,1H>)
            TX=(XMAX+XMIN)*0.5-15.0*HT*XS
            TY=YMAX+0.95*YS
            CALL PLTSYM(HT,TEMP,0.0,TX,TY)
            CALL PLTPGE
            RETURN
            END
```

```
      SUBROUTINE ERELCM(X,Y,R,ERP,XE,YE)
C  THIS COMPUTES ERROR ELLIPSE FOR GIVEN VARIANCE-COVARIANCE MATRIX R
C  THE ELLIPSE CORRESPONDS TO ERP STANDARD ERRORS
      DIMENSION R(2,2),XE(201),YE(201)
      C=0.  $  IF(R(1,1).LE.0..OR.R(2,2).LE.0.) GOTO 15
      C=R(1,2)/SQRT(R(1,1)*R(2,2))
15    A=0.  $ IF(C.GT.-1.) A=SQRT(1.+C)
      B=0.  $  IF(C.LT.1.) B=SQRT(1.-C)
      FX=0.  $  IF(R(1,1).GT.0.) FX=ERP*SQRT(R(1,1)*0.5)
      FY=0.  $  IF(R(2,2).GT.0.) FY=ERP*SQRT(R(2,2)*0.5)
      DO 25 KA=1,201
      FI=FLOAT(KA-1)*0.031415927
      XE(KA)=X+FX*(A*COS(FI)-B*SIN(FI))
      YE(KA)=Y+FY*(A*COS(FI)+B*SIN(FI))
25    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINPAR(PLAB,DIST,TIM,PIN,P,VP,DISTD,
     A TIMD,PIND,PD,VPD,NR,PNU,EXNU,TITLE)
C  SUBROUTINE PRINTS SUMMARY OF PRESSURE HISTORY FITTINGS
C  IT IS CALLED FROM MAIN AFTER ALL PRESSURE HISTORIES HAVE BEEN FITTED
C  IT ALSO COMPUTES INITIAL PARAMETER APPROXIMATIONS PNU AND EXPONENTS
C  EXNU FOR THE PRESSURE FIELD FUNCTION
C
      DIMENSION PLAB(50),DIST(50),TIM(50),PIN(50),P(4,50),VP(4,4,50),
     ADISTD(50),TIMD(50),PIND(50),PD(4,50),VPD(4,4,50),ER(4)
     B,PNU(10),EXNU(3),TITLE(3)

      LEVEL 2,P,VP,PD,VPD
C
      PRINT 12,(TITLE(J),J=1,3)
12    FORMAT(1H1,/,1H ,10X,5HEVENT,5X,3A10,/,1H ,10X,5(1H-))
      PRINT 15 $ PRINT 25
15    FORMAT(1H ,///,1H ,10X,20HSCALED PARAMETERS OF,
     A30H INDIVIDUAL PRESSURE HISTORIES,/)
   25 FORMAT(1H ,3X,3HNR.,5X,5HLABEL,6X,8HDISTANCE,2X,
     A 12HARRIVAL TIME,1X,9HOVERPRES.,5X,6HPAR(1),3X,9HSTD.ERROR,
     B 5X,6HPAR(2),3X,9HSTD.ERROR,5X,6HPAR(3),3X,9HSTD.ERROR,/)
      PRINT 16
16    FORMAT(1H+,23X,5H(SCD),6X,5H(SCT),8X,5H(SCP),
     A 6X,7H(1/SCT),4X,7H(1/SCT),4X,10H(1/SCT**2),1X,
     B 10H(1/SCT**2),5X,5H(SCP),5X,5H(SCP),/)

      DO 65 KA=1,NR
      DO 55 KB=1,3
   55 ER(KB)=SQRT(VP(KB,KB,KA))
      PRINT 35,KA,PLAB(KA),DIST(KA),TIM(KA),PIN(KA),
     A ((P(J,KA) ,ER(J)),J=1,3)
   65 CONTINUE

      PRINT 45  $  PRINT 25
   45 FORMAT(1H ,///,1H ,10X,25HDIMENSIONAL PARAMETERS OF,
     A30H INDIVIDUAL PRESSURE HISTORIES,/)
      PRINT 46
   46 FORMAT(1H+,24X,3H(M),8X,3H(S),9X,4H(PA),8X,5H(1/S),
     A 6X,5H(1/S),6X,8H(1/S**2),3X,8H(1/S**2),6X,4H(PA),
     B 6X,4H(PA),/)

      DO 85 KA=1,NR
      DO 75 KB=1,3
   75 ER(KB)=SQRT(VPD(KB,KB,KA))
      PRINT 35,KA,PLAB(KA),DISTD(KA),TIMD(KA),PIND(KA),
     A ((PD(J,KA),ER(J)),J=1,3)
   35 FORMAT(1H ,2X,I4,2X,A10,3(2X,1PE10.3),3(2X,1PE11.4,1X,1PE9.2))
   85 CONTINUE
C
C  NEXT COMPUTE INITIAL APPROXIMATIONS OF PRESSURE FIELD PARAMETERS
C  AND EXPONENTS FOR THE PRESSURE FIELD FUNCTION
C  BY STRAIGHT LINE LG,LG FIT OF PARAMETER(DISTANCE)
C
      DO 135 KB=1,3
      C11=0  $  C12=0  $  C22=0  $  RS1=0  $  RS2=0
      KK=0
      DO 105 KC=1,NR
```

```
      IF(DIST(KC).LE.0.) GOTO 105
      IF(ABS(P(KB,KC)).LT.1.E-30) GOTO 105
      KK=KK+1
      IF(KK.EQ.1) KM=KC
      IF(DIST(KC).LT.DIST(KM)) KM=KC
      ALD=ALOG(DIST(KC))
      PSQ=P(KB,KC)**2  $  ALP=0.5*ALOG(PSQ)
      C11=C11+PSQ  $  C12=C12+PSQ*ALD  $  C22=C22+PSQ*ALD**2
      RS1=RS1+PSQ*ALP  $  RS2=RS2+PSQ*ALP*ALD
      SIG=SIGN(1.,P(KB,KM))
C  USE THE SIGN OF PARAMETER CORRESPONDING TO SMALLEST DISTANCE
105   CONTINUE

      IF(KK.GE.2) GOTO 125
      PRINT 115,KB
      STOP
115   FORMAT(1H ,//,1H ,10X,15HSTOP BY PRINPAR,
     A 37H BECAUSE LESS THAN TWO HISTORIES HAVE,/,
     B 1H ,10X,19HNON-ZERO PARAMETER(,I1,1H))

125   C=(RS1*C22-RS2*C12)/(C11*C22-C12**2)
      EN=(RS2*C11-RS1*C12)/(C11*C22-C12**2)
      PNU(2*KB-1)=EXP(C)*SIG
      PNU(2*KB)=0.
      NEN=EN*10.  $  EXNU(KB)=-FLOAT(NEN)/10.
135   CONTINUE

      PNU(5)=-PNU(5)
      PRINT 145
  145 FORMAT(1H ,////,1H ,10X,22HINITIAL APPROXIMATIONS,
     A 36H OF SCALED PRESSURE FIELD PARAMETERS,//)

      DO 165 KB=1,3
      KC=2*KB-1  $  KD=2*KB
      PRINT 155,KC,PNU(KC),KD,PNU(KD),KB,EXNU(KB)
155   FORMAT(1H ,10X,4HPNU(,I1,2H)=,1PE12.5,5X,4HPNU(,I1,2H)=,1PE8.1,
     A 5X,5HEXNU(,I1,2H)=,0PF5.2)
165   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLTPAR(NRPROF,PRPD,PRDSD,TITLE)
C  THIS ROUTINE PLOTS HISTORY PARAMETERS VERSUS DISTANCE IN LOG-SCALES
C
C  NRPROF       = NUMBER OF HISTORIES OBSERVED
C  PRPD(4,50)   = HISTORY PARAMETERS
C  PRDSD(50)    = HISTORY DISTANCES
C  TITLE(3)     = DESIGNATION OF EVENT
C
      LEVEL 2,PRPD
      DIMENSION PRPD(4,50),PRDSD(50)
      DIMENSION TITLE(3)
      DIMENSION X(50),Y(50),TEMP(4)
      DIMENSION XA(50),NS(50),DIM(3)
      COMMON/PLOT/D(6),PLABL(4)
C
      DIM(1)=10H(1/S)>
      DIM(2)=10H(1/S**2)>
      DIM(3)=10H(PA)>
      CALL PLTBEG(8.7,11.2,1.0,13,PLABL)
      DO 1000 KA=1,3
      DO 100 KB=1,NRPROF
      X(KB)=ALOG10(PRDSD(KB))
      Y(KB)=ALOG10(ABS(PRPD(KA,KB)))
      XA(KB)=X(KB)
      NS(KB)=0
      IF(PRPD(KA,KB).LT.0.0) NS(KB)=1
C  USE SYMBOL NS=0 OR 1 FOR POSITIVE OR NEGATIVE PARAMETERS, RESPECTIVELY
  100 CONTINUE
C
      CALL SORTXY(X,Y,NRPROF)
      CALL SORTXY(XA,NS,NRPROF)
      CALL FLOGSC(X,NRPROF,4.0,XS,XMIN,XMAX,DX)
      CALL FLOGSC(Y,NRPROF,6.0,YS,YMIN,YMAX,DY)
      XS=AMAX1(XS,YS)
      YS=XS
      CALL PLTSCA(3.0,4.0,XMIN,YMIN,XS,YS)
      CALL PLTAXS(DX,DY,XMIN,XMAX,YMIN,YMAX,7)
      CALL LABLOG(DX,DY,XMIN,XMAX,YMIN,YMAX,0.0,0.0)
      CALL PLTDTS(1,0,X,Y,NRPROF,0)
      DO 120 KB=1,NRPROF
      CALL PLTDTS(3,NS(KB),X(KB),Y(KB),1,0)
  120 CONTINUE
      ENCODE(40,150,TEMP)
  150 FORMAT(*DISTANCE (M)>*)
      TX=(XMIN+XMAX)*0.5-6.0*0.1*XS
      TY=YMIN-0.5*YS
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)
      ENCODE(40,160,TEMP)KA,DIM(KA)
  160 FORMAT(*PARAMETER(*,I1,*)  *,A10)
      TX=XMIN-0.7*XS
      TY=(YMIN+YMAX)*0.5-9.0*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.0,TX,TY)
  900 ENCODE(40,370,TEMP) TITLE
  370 FORMAT(3A10,1H>)
      TX=(XMAX+XMIN)*0.5-15.0*0.1*XS
      TY=YMAX+0.5*YS
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)
```

```
      CALL PLTPGE
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FTPFLD(SCDIS,SCPRE,SCTIM,TITLE,PRLAB,PRDSD,
     A TARD,PIND,NRPROF,EXNU,PAR,VPAR,ERZ,NP,NBAD)
C
C CALLED FROM MAIN THIS FITS AN OVERPRESSURE FIELD MODEL TO ALL
C OVERPRESSURE DATA
C INITIAL VALUES OF PARAMETERS PAR ARE ASSUMED TO BE SPECIFIED BY
C THE CALLING PROGRAM
C
C   SCDIS,SCPRE,SCTIM = SCALES USED FOR THE PARAMETERS
C   TITLE  =  ALPHANUMERIC TITLE OF THIS RUN
C   PRLAB  =  ALPHANUMERIC LABELS OF HISTORIES
C   PRDSD  =  DISTANCES OF HISTORIES IN METRES
C   TARD   =  SHOCK ARRIVAL TIMES IN SECONDS
C   PIND   =  INCIDENTAL SHOCK OVERPRESSURES IN PASCALS
C   NRPROF =  NUMBER OF PROFILES (HISTORIES)
C   EXNU   =  EXPONENTS IN OVERPRESSURE MODEL FUNCTION
C
C THE FOLLOWING WILL BE COMPUTED BY THIS ROUTINE
C
C   PAR    =  PARAMETERS OF THE OVERPRESSURE FIELD MODEL
C   VPAR   =  VARIANCE-COVARIANCE MATRIX OF PAR, NOT INCLUDING ERZ**2
C   ERZ    =  STANDARD ERROR OF WEIGHT ONE
C   NP     =  NUMBER OF OVERPRESSURE FIELD FUNCTION PARAMETERS.
C             NP.NE.5  ONLY IN CASE OF ERROR RETURN
C
      DIMENSION TITLE(3),TARD(50),PIND(50),EXNU(3),PAR(10),VPAR(10,10)

      DIMENSION PST(6),VPF(10,10),ERPAR(10),PARG(10)
C
      EXTERNAL PFIELD,PFIELDC,PLDAUX
C
      COMMON/COMPR/TP(2,5000),ERTP(2,5000),ALB(2,5000),NSET(50),
     1 DIST(50),ERDIST(50)
      LEVEL2,TP,ERTP,ALB,NSET,DIST,ERDIST
      COMMON/CFLDEX/EXA,EXB,EXC
      COMMON/CSCALE/SCDI,SCPR,SCTI
      COMMON/SCRCH2/ X(3,5000),R(3,3,5000),LSTX(5000),XC(3,5000),
     1 C(3,5000),WORK(14307),LSTN(5000)
      LEVEL 2,X,R,LSTX,XC,C,WORK,LSTN
C
      COMMON/TPINDX/ITC,IPC
C TIME AND PRESSURE INDEX IN X-ARRAY
      DATA (IT=2),(IP=1)
      ITC=IT  $  IPC=IP
C X(IT)=TIME  ,  X(IP)=OVERPRESSURE, X(3)=DISTANCE
C
      SCDI=SCDIS  $  SCPR=SCPRE  $  SCTI=SCTIM
C THE SCALES ARE NEEDED IN QFUNCT WHICH IS CALLED FROM PFIELD
C
      EXA=EXNU(1)  $  EXB=EXNU(2)  $  EXC=EXNU(3)
C STORE EXPONENTS TO BE USED BY THE PRESSURE FIELD AUXILIARY FUNCTIONS
C ACOEF, BCOEF AND CCOEF
C
      NXD=3  $  NPD=10   $ NWORK=14307
      NBAD=0
      IF(SCDIS.GT.0.0.AND.SCPRE.GT.0.0.AND.SCTIM.GT.0.0)GOTO 15
      NBAD=1$ PRINT 20,NBAD$ RETURN
```

```
   15 IF(NRPROF.GT.1)GOTO 23
      NBAD=2
      PRINT 20,NBAD$ RETURN
   20 FORMAT(1H0,10X,29HRETURN FROM FTPFLD WITH NBAD=,I3)
   23 KCS=0 $ KC=0
      DO 35 KA=1,NRPROF
      KBM=NSET(KA)   $   IF(KBM.LE.0) GOTO 35
      DO 25 KB=1,KBM
      KC=KCS+KB
      X(IT,KC)=TP(1,KC)/SCTIM  $  R(IT,IT,KC)=(ERTP(1,KC)/SCTIM)**2
      X(IP,KC)=TP(2,KC)/SCPRE  $  R(IP,IP,KC)=(ERTP(2,KC)/SCPRE)**2
      X(3,KC)=DIST(KA)/SCDIS$ R(3,3,KC)=(ERDIST(KA)/SCDIS)**2
      R(1,2,KC)=0$ R(1,3,KC)=0$ R(2,3,KC)=0
      R(2,1,KC)=0$ R(3,1,KC)=0$ R(3,2,KC)=0
      LSTX(KC)=0
      XC(2,KC)=X(2,KC)  $  XC(3,KC)=X(3,KC)
      C(2,KC)=0.0  $  C(3,KC)=0.0
      WORK(KC)=PIND(KA)/SCPRE $ WORK(6000+KC)=TARD(KA)/SCTIM
C  STORE SHOCK OVERPRESSURE AND ARRIVAL TIME FOR FLDGES
   25 CONTINUE
      KCS=KC
   35 CONTINUE
      NR=KC
C
      PARG(5)=PAR(5)
      CALL FLDGES(X,R,WORK(1),WORK(6001),NR,EXNU,PARG,NBAD)
C  THIS COMPUTES BETTER INITIAL APPROXIMATIONS OF PARG
      IF(NBAD.NE.0)GOTO 39
C  BRANCH AND TRY APPROXIMATIONS PROVIDED BY CALLING PROGRAM
      DO 38 KA=1,6
   38 PAR(KA)=PARG(KA)

   39 CONTINUE
      DO 47 KA=1,6
47    PST(KA)=PAR(KA)
C
      NX=1  $  NP=5  $  ITYPE=0
      CALLCOLSACA(X,R,ALB,LSTX,NX,NR,PAR,NP,PFIELDC,ITYPE,
     AXC,C,LSTN,NRGD,ERZ,VPAR,ERPAR,NBAD,NXD,NPD,WORK,NWORK)
      NX=2  $  NP=5  $  ITYPE=1
      IF(NBAD.EQ.0) GOTO 52
C
49    PAR(1)=PST(1) $ PAR(2)=PST(3) $ PAR(3)=PST(5)
      NX=1  $  NP=3  $  ITYPE=0
      CALL COLSACA(X,R,ALB,LSTX,NX,NR,PAR,NP,PLDAUX,ITYPE,
     1 XC,C,LSTN,NRGD,ERZ,VPF,ERPAR,NBAD,NXD,NPD,WORK,NWORK)
      IF(NBAD.NE.0) RETURN
      NX=2  $  NP=3  $  ITYPE=1
      CALLCOLSACA(X,R,ALB,LSTX,NX,NR,PAR,NP,PLDAUX,ITYPE,
     1 XC,C,LSTN,NRGD,ERZ,VPF,ERPAR,NBAD,NXD,NPD,WORK,NWORK)
      IF(NBAD.NE.0) RETURN
      NX=3  $  NP=3  $  ITYPE=1
      CALL COLSACA(X,R,ALB,LSTX,NX,NR,PAR,NP,PLDAUX,ITYPE,
     1 XC,C,LSTN,NRGD,ERZ,VPF,ERPAR,NBAD,NXD,NPD,WORK,NWORK)
      IF(NBAD.NE.0) RETURN
      PAR(5)=PAR(3)  $  PAR(3)=PAR(2)
      PAR(2)=PST(2)  $  PAR(4)=PST(4)
```

```
          NX=3  $  NP=5  $  ITYPE=1
          GOTO 54
C
52        CONTINUE
          CALLCOLSACA(X,R,ALB,LSTX,NX,NR,PAR,NP,PFIELDC,ITYPE,
         AXC,C,LSTN,NRGD,ERZ,VPAR,ERPAR,NBAD,NXD,NPD,WORK,NWORK)
          NX=3  $  NP=5  $  ITYPE=1
          IF(NBAD.EQ.0) GOTO 54
          DO 53 KA=1,NR  $  XC(2,KA)=X(2,KA)
53        C(2,KA)=0.
          GOTO 49
C
54        CONTINUE
          CALLCOLSACA(X,R,ALB,LSTX,NX,NR,PAR,NP,PFIELDC,ITYPE,
         AXC,C,LSTN,NRGD,ERZ,VPAR,ERPAR,NBAD,NXD,NPD,WORK,NWORK)
          IF(NBAD.EQ.0) GOTO 55
          RETURN

   55 CONTINUE
          CALL PRTFLD(SCDIS,SCPRE,SCTIM,TITLE,PRLAB,PRDSD,TARD,
         A PIND,X,R,ALB,NR,C)
C   PRINT FIELD ADJUSTMENT RESULTS (RESIDUALS)
          CALL PLTFLD(TITLE,TARD,PIND,PAR,VPAR,ERZ,NP,NRPROF)
C   PLOT OVERPRESSURE FIELD HISTORIES
          RETURN
          END
```

. EXCEEDS 131,071 WORDS (LCM=I REQUIRED)

```
      SUBROUTINE FLOGES(X,R,PS,TS,NR,EXNU,PARG,NBAD)
C  THIS COMPUTES INITIAL APPROXIMATIONS OF FIELD PARAMETERS PARG.
C  FLOGES IS CALLED FROM FTPFLD.
C
C  X       = TIME,OVERPRESSURE,DISTANCE
C  R       = VARIANCE-COVARIANCE MATRICES OF X
C  PS      = INCIDENTAL SHOCK OVERPRESSURES
C  TS      = SHOCK ARRIVAL TIMES
C  NR      = NUMBER OF DATA POINTS
C  EXNU    = EXPONENTS IN OVERPRESSURE FIELD FORMULA
C
C  THE FOLLOWING WILL BE PROVIDED BY THIS PROGRAM
C
C  PARG    = FIELD PARAMETERS
C  NBAD    = ERROR INDICATOR.  NBAD.NE.0  IN CASE OF ERROR RETURN

      DIMENSIONX(3,5000),R(3,3,5000),PS(5000),TS(5000),EXNU(3),PARG(10)
      LEVEL2,X,R,PS,TS

      COMMON/TPINOX/ITC,IPC
C  X(ITC,K)= TIME,  X(IPC,K)=OVERPRESSURE
      COMMON/GUECM/AN(3,3),RS(3),W(18)
      DOUBLE PRECISION AN,RS,W,DET
      LEVEL 2,AN,RS,W

      NBAD=0
      FMIN=X(IPC,1)*X(3,1)**EXNU(3) $ FMAX=FMIN
      DO 15 KA=2,NR
      FF=X(3,KA)**EXNU(3)
      FMIN=AMIN1(FMIN,X(IPC,KA)*FF,PS(KA)*FF)
      FMAX=AMAX1(FMAX,X(IPC,KA)*FF,PS(KA)*FF)
   15 CONTINUE

      CMAX=FMIN-ABS(FMAX)*0.001
      CMIN=AMIN1(-0.5*ABS(FMAX),CMAX)
      C=AMIN1(CMAX,AMAX1(PARG(5),CMIN))

   25 KIT=0
C  KIT IS ITERATION COUNTER
      NX=3 $ NA=3 $ KIN=1
      IF(CMIN.EQ.CMAX) NX=2

C  NEXT ESTABLISH NORMAL EQS FOR SIMPLIFIED PROBLEM
   35 DO 55 KA=1,3 $ DO 45 KB=1,3
   45 AN(KA,KB)=0
   55 RS(KA)=0

C  THE FITTED FUNCTION IS OF THE FORM Y=F(A,B),I.E.,
C     ALOG((P-CD)/(PS-CD))=AD*(T-TS)+BD*(T-TS)**2,
C     WHERE  AD=A/D**EXNU(1),  BD=B/D**EXNU(2),
C     CR=C/D**EXNU(3),  AND  D IS DISTANCE
C  THE WEIGHTS ARE   (P-CD)**2/R
C  THE FIRST TERM IS LINEARIZED FOR CORRECTION EPS OF C
C  INITIAL VALUE C=PARG(5) PROVIDED BY CALLING PROGRAM

      DO 65 KA=1,NR
      PC=X(IPC,KA)-C/X(3,KA)**EXNU(3)
```

```
      PSC=PS(KA)-C/X(3,KA)**EXNU(3)
      ERF=PC**2/R(IPC,IPC,KA)
      TAU=(X(ITC,KA)-TS(KA))/X(3,KA)**EXNU(1)
      TAUS=(X(ITC,KA)-TS(KA))**2/X(3,KA)**EXNU(2)
      RO=(PSC-PC)/(PSC*PC*X(3,KA)**EXNU(3))
      AL=ALOG(PC/PSC)
      AN(1,1)=AN(1,1)+ERF*TAU**2
      AN(1,2)=AN(1,2)+ERF*TAU*TAUS
      AN(1,3)=AN(1,3)+ERF*TAU*RO
      RS(1)=RS(1)+ERF*TAU*AL
      AN(2,2)=AN(2,2)+ERF*TAUS**2
      AN(2,3)=AN(2,3)+ERF*TAUS*RO
      RS(2)=RS(2)+ERF*TAUS*AL
      AN(3,3)=AN(3,3)+ERF*RO**2
      RS(3)=RS(3)+ERF*RO*AL
   65 CONTINUE

      AN(2,1)=AN(1,2) $ AN(3,1)=AN(1,3) $ AN(3,2)=AN(2,3)

      CALL MTRINDB(AN,NX,RS,NA,KIN,DET,W)
C  THIS SOLVED THE NORMAL EQUATIONS
      IF(NX.EQ.2.OR.DET.NE.0.)GOTO 75
      NX=2 $ NA=3 $ KIN=1
      GOTO 35

   75 EPS=RS(3) $ IF(NX.EQ.2)EPS=0
      C=AMAX1(CMIN,AMIN1(C+EPS,CMAX))
      IF(CMIN.EQ.CMAX) GOTO 85
C  NO ITERATION FOR C IF C IS FIXED
      KIT=KIT+1
      NX=3 $ NA=3 $ KIN=1
      IF(KIT.LT.4)GOTO 35
C  ITERATE THREE TIMES

85    PARG(1)=RS(1) $ PARG(3)=RS(2) $ PARG(5)=C
      PARG(2)=0 $ PARG(4)=0
      IF(DET.EQ.0.)NBAD=1
      RETURN
      END
```

```
      SUBROUTINE PFIELD(X.KK,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C
C  THIS IS THE OVERPRESSURE FIELD FUNCTION CONSTRAINT ROUTINE
C  THE ARGUMENTS ARE EXPLAINED IN COLSACB AND IN COLSAC MANUAL
C
C  THE FUNCTION F IS DEFINED AS
C     F=(PSHOCK-C)K*EXP(Q(T,R,P(1),...,P(4))+C(R,P(5)) - P
C  THE OBSERVABLES ARE
C       TIME T=X(IT),   OVERPRESSURE P=X(IP),   RADIUS R=X(3)
C  THE INDEXES IT AND IP ARE IN COMMON/TPINDX/
C  THE FUNCTIONS Q,PSHOCK,C WILL BE OBTAINED BY CALLING
C  QFUNCT AND CCOEF.
C
      LEVEL 2,X,FX,FP,FXX,FXP,FPP
      DIMENSION X(3,1),PAR(10),FX(3),FP(10),FXX(3,3),FXP(3,10),FPP(10,10
     1)
      DIMENSION QX(3),QP(10),QXX(3,3),QXP(3,10),QPP(10,10),CX(3),
     ACP(10),CXX(3,3),CXP(3,10),CPP(10,10),PSP(10),PSRP(10),PSPP(10,10)
      DIMENSION PSCX(3),PSCP(10)
C
      COMMON/TPINDX/IT,IP
C  /TPINDX/ IS SET BY FTPFLD
C  TIME=X(IT)  , OVERPRESSURE=X(IP),   DISTANCE=X(3)
C
      NPSHK=4  $  GOTO 10
      ENTRY PFIELDC
      NPSHK=0
10    CONTINUE
C
C  ENTRY PFIELDC IS USED AS CONSTRAINT FOR PRESSURE FIELD ADJUSTMENT
C  IT DOES NOT COMPUTE DERIVATIVES WITH RESPECT TO THE SHOCK
C  PARAMETERS PAR(6) THROUGH PAR(9)
C
C  ENTRY PFIELD IS USED TO COMPUTE THE PRESSURE FIELD AFTER ADJUSTMENT
C  IT COMPUTES DERIVATIVES OF THE OVERPRESSURE WITH RESPECT TO
C  ALL PARAMETERS
C
      DO 12 KB=1,10
      FXP(1,KB)=0  $  FXP(2,KB)=0  $  FXP(3,KB)=0   $ FP(KB)=0
      DO 12 KC=1,10
12    FPP(KC,KB)=0
      NBAD=0
      CALL QFUNCT(X,KK,PAR,Q,QX,QP,QXX,QXP,QPP,
     APS,PSR,PSP,PSRR,PSRP,PSPP,NPSHK,NBAD)
      IF(NBAD.NE.0)RETURN
      CALL CCOEF(X,KK,PAR,C,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.NE.0)RETURN
C
   13 EXPQ=0.0
      PSC=PS-C
      IF(Q.GE.-675.84.AND.Q.LE.741.67) EXPQ=EXP(Q)
      IF(Q.LE.100.) GOTO 14
C  LARGE EXP HAS CAUSED OVERFLOW IN COLSAC
      IF(Q.LE.741.67) GOTO 14
      NBAD=101
      RETURN
   14 CONTINUE
```

```
      FEX=PSC*EXPQ
      F=FEX+C-X(IP,KK)
      DO 15 KB=1,3
      PSCX(KB)=-CX(KB)
15    FX(KB)=EXPQ*(PSC*QX(KB)+PSCX(KB))+CX(KB)
      FX(IP)=FX(IP)-1.
      PSCX(3)=PSCX(3)+PSR
      FX(3)=FX(3)+EXPQ*PSR
      DO 25 KB=1,5
      PSCP(KB)=-CP(KB)
25    FP(KB)=EXPQ*(PSC*QP(KB)+PSCP(KB))+CP(KB)
C
      DO 32 KB=1,3 $ DO 32 KC=1,3
      FXX(KB,KC)=EXPQ*(PSC*(QXX(KB,KC)+QX(KB)*QX(KC))
     A+QX(KB)*PSCX(KC)+PSCX(KB)*QX(KC)-CXX(KB,KC))+CXX(KB,KC)
   32 CONTINUE
      FXX(3,3)=FXX(3,3)+EXPQ*PSRR
C
      DO 35 KB=1,3 $ DO 35 KC=1,5
      FXP(KB,KC)=EXPQ*(PSC*(QXP(KB,KC)+QX(KB)*QP(KC))
     A+QX(KB)*PSCP(KC)+PSCX(KB)*QP(KC)-CXP(KB,KC))+CXP(KB,KC)
   35 CONTINUE
C
      DO 45 KB=1,5 $ DO 45 KC=1,5
      FPP(KB,KC)=EXPQ*(PSC*(QPP(KB,KC)+QP(KB)*QP(KC))
     A+QP(KB)*PSCP(KC)+PSCP(KB)*QP(KC)-CPP(KB,KC))+CPP(KB,KC)
45    CONTINUE
C
      IF(NPSHK.LE.0)GOTO 75
C  NPSHK IS THE NUMBER OF SHOCK PARAMETERS.  NPSHK=0  OR  =4
      KUP=5+4
C  ASSUME THAT PRESSURE FUNCTION HAS 5 PARAMETERS AND SHOCK HAS 4 PAR.
      DO 55 KB=6,KUP
      PSCP(KB)=PSP(KB)
      FP(KB)=EXPQ*(PSC*QP(KB)+PSCP(KB))
      DO 52 KC=1,3
      FXP(KC,KB)=EXPQ*(PSC*(QXP(KC,KB)+QX(KC)*QP(KB))
     A+QX(KC)*PSCP(KB)+PSCX(KC)*QP(KB))
52    CONTINUE
      FXP(3,KB)=FXP(3,KB)+EXPQ*PSRP(KB)
      DO 55 KC=6,KUP
      FPP(KB,KC)=EXPQ*(PSC*(QPP(KB,KC)+QP(KB)*QP(KC))
     A+QP(KB)*PSCP(KC)+PSCP(KB)*QP(KC)+PSPP(KB,KC))
   55 CONTINUE
      DO 65 KB=1,5  $  DO 65 KC=6,KUP
      FPP(KB,KC)=EXPQ*(PSC*(QPP(KB,KC)+QP(KB)*QP(KC))
     A+QP(KB)*PSCP(KC)+PSCP(KB)*QP(KC)+PSPP(KB,KC))
65    FPP(KC,KB)=FPP(KB,KC)
   75 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLDAUX(X,KK,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS CONVERTS THE FIVE PARAMETER PRESSURE FIELD FUNCTION INTO A
C  THREE PARAMETER FUNCTION.  IT IS USED BY FTPFLD IN CASE OF
C  ALGORITHMIC PROBLEMS TO OBTAIN INITIAL APPROXIMATIONS FOR
C  THE FINAL FIVE PARAMETER FITTING
C
      DIMENSION X(3,1),PAR(10),FX(3),FP(10),FXX(3,3),FXP(3,10),
     1 FPP(10,10),P(10)
      LEVEL 2,X,FX,FP,FXX,FXP,FPP
      COMMON/SCRCH4/ GP(10),GXP(3,10),GPP(10,10)
      LEVEL 2,GP,GXP,GPP
C
      P(1)=PAR(1)  $  P(3)=PAR(2)  $  P(5)=PAR(3)  $  P(2)=0 $ P(4)=0
      CALL PFIELDC(X,KK,P,F,FX,GP,FXX,GXP,GPP,NBAD)
      DO 15 KA=1,3  $  FP(KA)=GP(KA*2-1)
      DO 15 KB=1,3  $  FXP(KB,KA)=GXP(KB,KA*2-1)
15    FPP(KB,KA)=GPP(KB*2-1,KA*2-1)
      RETURN  $  END
```

```
      SUBROUTINE QFUNCT(X,KK,PAR,Q,QX,QP,QXX,QXP,QPP,
     APS,PSR,PSP,PSRR,PSRP,PSPP,NPSHK,NBAD)
C  AUXILIARY ROUTINE FOR PFIELD. IT COMPUTES THE EXPONENT Q OF THE
C  PRESSURE FIELD FUNCTION. IT ALSO TR.  MITS THE SHOCK
C  OVERPRESSURE PS(R) WITH DERIVATIVES.
C
C     SUBROUTINES ACOEF,BCOEF AND SHODER ARE NEEDED
C
      LEVEL 2,X
      DIMENSION X(3,1),PAR(10),QX(3),QP(10),QXX(3,3),QXP(3,10),
     AQPP(10,10),AX(3),AP(10),AXX(3,3),AXP(3,10),APP(10,10),
     BTAUX(3)
      DIMENSION TP(10),TRP(10),TPP(10,10),PSP(10),PSRP(10),PSPP(10,10)
C
      COMMON/CSCALE/SCDIS,SCPRE,SCTIM
      COMMON/COMSHK/NPSH ,PARSH(4),VPARSH(4,4),SCDSH,SCPSH,SCTSH
C
      COMMON/TPINDX/IT,IP
C  /TPINDX/ IS SET BY FTPFLD
C  TIME=X(IT)  , OVERPRESSURE=X(IP),  DISTANCE=X(3)
C
      DO 12 KA=1,10 $ QP(KA)=0 $ DO 10 KB=1,3
   10 QXP(KB,KA)=0 $ DO 12 KC=1,10
   12 QPP(KA,KC)=0
      NBAD=0 $ R=X(3,KK)*SCDIS
C
      IF(NPSHK.GT.0) GOTO 13
C  IF NPSHK = NUMBER OF SHOCK PARAMETERS IS ZERO THEN COMPUTE ONLY
C  DERIVATIVES WITH RESPECT TO PRESSURE PARAMETERS PAR(1) THROUGH PAR(5)
      CALL SHOCK2(R,T,TR,TRR,PS,PSR,PSRR,NBAD)
      IF(NBAD.NE.0) RETURN
      GOTO 14
C
13    CONTINUE
      CALL SHODER(    R,T,TR,TP,TRR,TRP,TPP,PS,PSR,PSP,
     APSRR,PSRP,PSPP,NBAD)
      IF(NBAD.NE.0)RETURN
C
14     CONTINUE
C  SHOCK2 OR SHODER COMPUTED EVRYTHING IN SI UNITS. NOW SCALE RESULTS
C  ACCORDING TO THE SCALES IN  /CSCALE/
      T=T/SCTIM  $  TR=TR*SCDIS/SCTIM  $  TRR=TRR*SCDIS**2/SCTIM
      PS=PS/SCPRE  $  PSR=PSR*SCDIS/SCPRE  $ PSRR=PSRR*SCDIS**2/SCPRE
      IF(NPSHK.LE.0) GOTO 16
C
      DO 15 KB=6,8
      TP(KB)=TP(KB)*SCPRE*SCDIS**(KB-5)/SCTIM
      PSP(KB)=PSP(KB)*SCDIS**(KB-5)
      TRP(KB)=TRP(KB)*SCDIS**(KB-4)*SCPRE/SCTIM
      PSRP(KB)=PSRP(KB)*SCDIS**(KB-4)
      TPP(9,KB)=TPP(9,KB)*SCPRE*SCDIS**(KB-5)  $  TPP(KB,9)=TPP(9,KB)
      PSPP(9,KB)=PSPP(9,KB)*SCTIM*SCDIS**(KB-5)  $ PSPP(KB,9)=PSPP(9,KB)
      DO 15 KC=6,8
      TPP(KC,KB)=TPP(KC,KB)*(SCPRE/SCTIM)**2*SCDIS**(KB+KC-10)
      PSPP(KC,KB)=PSPP(KC,KB)*SCDIS**(KB+KC-10)
15    CONTINUE
      PSP(9)=PSP(9)*SCTIM/SCPRE
```

```
      TPP(9,9)=TPP(9,9)*SCTIM
      PSPP(9,9)=PSPP(9,9)*(SCTIM/SCPRE)**2
C
16    CONTINUE
      TAU=X(IT,KK)-T
      TAUX(IT)=1.0  $  TAUX(IP)=0.0  $  TAUX(3)=-TR
C
C  NEXT COMPUTE THE LINEAR TERM IN THE EXPONENT
      CALL ACOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
      IF(NBAD.NE.0)RETURN
      Q=A*TAU
C
      DO 25 KB=1,3
      QX(KB)=AX(KB)*TAU+A*TAUX(KB)
      DO 25 KC=1,3
      QXX(KB,KC)=AXX(KB,KC)*TAU+AX(KB)*TAUX(KC)+AX(KC)*TAUX(KB)
   25 CONTINUE
      QXX(3,3)=QXX(3,3)-A*TRR
C
      DO 35 KB=1,3 $ DO 35 KC=1,5
   35 QXP(KB,KC)=AXP(KB,KC)*TAU+AP(KC)*TAUX(KB)
C
      DO 45 KB=1,5 $ QP(KB)=AP(KB)*TAU
      DO 45 KC=1,5
   45 QPP(KB,KC)=APP(KB,KC)*TAU
      IF(NPSHK.LE.0)GOTO 53
C  NPSHK IS THE NUMBER OF SHOCK PARAMETERS
      KUP=5+NPSHK
C  ASSUME THAT PRESSURE FIELD HAS 5 PARAMETERS
      DO 48 KA=6,KUP
      QP(KA)=-A*TP(KA)
      QXP(3,KA)=-AX(3)*TP(KA)-A*TRP(KA)
      DO 48 KB=6,KUP
   48 QPP(KA,KB)=-A*TPP(KA,KB)
      DO 50 KA=1,5 $ DO 50 KB=6,KUP
      QPP(KA,KB)=-AP(KA)*TP(KB)
   50 QPP(KB,KA)=QPP(KA,KB)
C
C  NEXT COMPUTE QUADRATIC TERM
53    CALL BCOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
      IF(NBAD.NE.0)RETURN
      Q=Q+A*TAU*TAU
C
      DO 55 KB=1,3
      QX(KB)=QX(KB)+TAU*(AX(KB)*TAU+2.*A*TAUX(KB))
      DO 55 KC=1,3
      QXX(KB,KC)=QXX(KB,KC)+TAU*(AXX(KB,KC)*TAU+2.*AX(KB)*TAUX(KC)
     A+2.*AX(KC)*TAUX(KB))+2.*A*TAUX(KB)*TAUX(KC)
   55 CONTINUE
      QXX(3,3)=QXX(3,3)-2.*A*TAU*TRR
C
      DO 65 KB=1,3 $ DO 65 KC=1,5
      QXP(KB,KC)=QXP(KB,KC)+TAU*(AXP(KB,KC)*TAU+2.*
     ATAUX(KB)*AP(KC))
   65 CONTINUE
C
      DO 75 KB=1,5 $ QP(KB)=QP(KB)+AP(KB)*TAU*TAU
```

```
      DO 75 KC=1,5
   75 QPP(KB,KC)=QPP(KB,KC)+APP(KB,KC)*TAU*TAU
      IF(NPSHK.LE.0)GOTO 97
      DO 85 KA=6,KUP
      QP(KA)=QP(KA)-A*2.*TAU*TP(KA)
      QXP( 3,KA)=QXP(3,KA)+2.*(-AX(3)*TAU*TP(KA)+A*TP(KA)*TR
     A-A*TAU*TRP(KA))
      DO 85 KB=6,KUP
      QPP(KA,KB)=QPP(KA,KB)+A*2.*(TP(KA)*TP(KB)-TAU*TPP(KA,KB))
   85 CONTINUE
      DO 95 KA=6,KUP $ DO 95 KB=1,5
      QPP(KB,KA)=QPP(KB,KA)-2.*AP(KB)*TP(KA)*TAU
   95 QPP(KA,KB)=QPP(KB,KA)
   97 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ACOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
C  LINEAR COEFFICIENT IN PRESSURE FIELD EXPONENT
C  AUXILIARY ROUTINE FOR QFUNCT
      DIMENSION X(3,1),PAR(10),AX(3),AP(10),AXX(3,3),AXP(3,10),
     AAPP(10,10),CP(2),CXP(2),CPP(2,2)
      LEVEL 2,X
      COMMON/CFLDEX/EXA,EXB,EXC
      NBAD=0
      R=X(3,KK) $ P1=PAR(1) $   P2=PAR(2)
      EX=EXA
      CALL COEFFI(R,P1,P2,EX,A,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.EQ.0)GOTO 15 $ NBAD=NBAD+100 $ RETURN
C
   15 DO 25 KA=1,5 $ AP(KA)=0 $ IF(KA.LE.3)AX(KA)=0
      DO 25 KB=1,5 $ IF(KA.LE.3)AXP(KA,KB)=0
      IF(KA.LE.3.AND.KB.LE.3)AXX(KA,KB)=0
   25 APP(KA,KB)=0
C
      AX(3)=CX $ AP(1)=CP(1) $ AP(2)=CP(2)
      AXX(3,3)=CXX $ AXP(3,1)=CXP(1) $ AXP(3,2)=CXP(2)
      DO 35 KA=1,2 $ DO 35 KB=1,2
   35 APP(KA,KB)=CPP(KA,KB)
      RETURN $ END
```

```
      SUBROUTINE BCOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
C  QUADRATIC COEFFICIENT IN PRESSURE FIELD EXPONENT
C  AUXILIARY ROUTINE FOR QFUNCT
      DIMENSION X(3,1),PAR(10),AX(3),AP(10),AXX(3,3),
     AAXP(3,10),APP(10,10),CP(2),CXP(2),CPP(2,2)
      LEVEL 2,X
      COMMON/CFLDEX/EXA,EXB,EXC
      NBAD=0
      R=X(3,KK) $ P1=PAR(3) $   P2=PAR(4)
      EX=EXB
      CALL COEFFI(R,P1,P2,EX,A,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.EQ.0)GOTO 15 $ NBAD=200+NBAD $ RETURN
C
   15 DO 25 KA=1,5 $ AP(KA)=0 $ IF(KA.LE.3)AX(KA)=0
      DO 25 KB=1,5 $ IF(KA.LE.3)AXP(KA,KB)=0
      IF(KA.LE.3.AND.KB.LE.3)AXX(KA,KB)=0
   25 APP(KA,KB)=0
C
      AX(3)=CX $ AP(3)=CP(1) $ AP(4)=CP(2)
      AXX(3,3)=CXX $ AXP(3,3)=CXP(1) $ AXP(3,4)=CXP(2)
      DO 35 KA=1,2 $ DO 35 KB=1,2
   35 APP(2+KA,2+KB)=CPP(KA,KB)
      RETURN $ END
```

```
      SUBROUTINE CCOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
C  THIS IS ADDITIVE COEFFICIENT IN PRESSURE FIELD FORMULA
C  AUXILIARY ROUTINE FOR PFIELD
      DIMENSION X(3,1),PAR(10),AX(3),AP(10),AXX(3,3),AXP(3,10),
     AAPP(10,10),CP(2),CXP(2),CPP(2,2)
      LEVEL 2,X
      COMMON/CFLDEX/EXA,EXB,EXC
      NBAD=0
      R=X(3,KK) $ P1=PAR(5) $ P2=0.
      EX=EXC
      CALL COEFFI(R,P1,P2,EX,A,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.EQ.0)GOTO 15 $ NBAD=NBAD+300 $ RETURN
C
   15 DO 25 KA=1,5 $ AP(KA)=0 $ IF(KA.LE.3)AX(KA)=0
      DO 25 KB=1,5 $ IF(KA.LE.3)AXP(KA,KB)=0
      IF(KA.LE.3.AND.KB.LE.3)AXX(KA,KB)=0
   25 APP(KA,KB)=0
C
      AX(3)=CX $ AP(5)=CP(1)
      AXX(3,3)=CXX  $  AXP(3,5)=CXP(1)
      APP(5,5)=CPP(1,1)
      RETURN $ END
```

```
      SUBROUTINE COEFFI(R,P1,P2,EX,A,AX,AP,AXX,AXP,APP,NBAD)
C  THIS COMPUTES TAU COEFFICIENTS TO BE USED IN PRESSURE FIELD
C  FUNCTION EXPONENT AND AS ADDITIVE TERM. THE COEFFICIENTS DEPEND ON R
C
      DIMENSION AP(2),AXP(2),APP(2,2)
C
      NBAD=0
      REX=1./R**EX
      A=REX*(P1+P2*R)
C  A IS THE COEFFICIENT.  NEXT COMPUTE FIRST ORDER DERIVATIVES
      AX=REX*(-P1*EX/R+P2*(1.-EX))
      AP(1)=REX  $  AP(2)=REX*R
C  NEXT COMPUTE SECOND ORDER DERIVATIVES
      AXX=REX*(P1*EX*(EX+1.)/R-P2*(1.-EX)*EX)/R
      AXP(1)=REX*(-EX)/R  $  AXP(2)=REX*(1.-EX)
      APP(1,1)=0. $ APP(1,2)=0. $ APP(2,1)=0. $ APP(2,2)=0.
      RETURN  $ END
```

```
      SUBROUTINE SHOCK(R,T,POV,US,UP,RHO,NBAD)
C  THIS COMPUTES SHOCK VALUES USING PARAMETERS FROM /COMSHCK/
C  ALL ARGUMENTS ARE ASSUMED TO BE  EXPRESSED IN SI UNITS
C  ROUTINE USES ROMBIN AND SHTINT TO COMPUTE SHOCK ARRIVAL TIME
C
C     R      =  SHOCK DISTANCE (GIVEN)
C     T      =  SHOCK ARRIVAL TIME
C     POV    =  INCIDENTAL SHOCK OVERPRESSURE
C     US     =  SHOCK SPEED
C     UP     =  PARTICLE VELOCITY BEHIND SHOCK
C     RHO    =  SHOCK DENSITY
C     NBAD   =  ERROR INDICATOR.  NBAD.NE.0  IN CASE OF ERROR RETURN
C
      EXTERNAL SHTINT
C  INTEGRAND TO COMPUTE SHOCK ARRIVAL TIME
C
      COMMON/COMSHK/NPS,PARSH(4),VPARSH(4,4),SCDIS,SCPRE,SCTIM
      COMMON/AMBCHA/PZ,TZ,GAM,AMOL,CHVOL,CHEN,CHH,CHHER
      COMMON/CF2DER/GAMCAP,SNDSPD,PAR(4),ALOW,SCD,SCP,SCT
C
      GAMCAP=GAMCAP/SCP  $  SNDSPD=SNDSPD*SCD/SCT  $ ALOW=ALOW*SCD
      SCD=1.  $  SCP=1.  $  SCT=1.
      DO 15 KA=1,3
15    PAR(KA)=PARSH(KA)*SCPRE *SCDIS**KA
      PAR(4)=PARSH(4)*SCTIM
C  THIS CHANGED THE CONTENTS OF /CF2DER/ INTO SI UNITS
C
      POV=((PAR(3)/R+PAR(2))/R+PAR(1))/R
      CALL ROMBIN(SHTINT,ALOW,R,F,NBAD)
C  QUADRATURE TO COMPUTE SHOCK ARRIVAL TIME
      IF(NBAD.EQ.0) GO TO 30
      PRINT 20,NBAD
   20 FORMAT(1H ,*RETURN FROM SHOCK WITH NBAD= *,I5)
      RETURN
C
   30 CONTINUE
      T=F/SNDSPD +PAR(4)
      US=SQRT(SNDSPD**2*(1.+GAMCAP*POV))
      RHOZ=(AMOL/8.3143)*(PZ/TZ)
      UP=POV/(RHOZ*US)
      RHO=RHOZ*(1.+GAMCAP*POV)/(1.+(GAM-1.)*POV*0.5/(GAM*PZ))
      RETURN
      END
```

```
      SUBROUTINE SHOCK2(R,T,TR,TRR,P,PR,PRR,NBAD)
C  THIS ROUTINE COMPUTES SHOCK ARRIVAL TIME AND OVERPRESSURE FOR
C  GIVEN DISTANCE
CC
C  R   = SHOCK DISTANCE (GIVEN)
C  T   = SHOCK ARRIVAL TIME
C  TR, TRR  =  DERIVATIVES OF T WITH RESPECT TO R
C  P   = SHOCK OVERPRESSURE
C  PR, PRR  =  DERIVATIVES OF P WITH RESPECT TO R
C
C  ALL QUANTITIES ARE COMPUTED IN SI UNITS
C
      EXTERNAL SHTINT
      COMMON/COMSHK/NPS,PARS(4),VP(4,4),SCDS,SCPS,SCTS
      COMMON/CF2DER/GAMCAP,SNDSPD,CP(4),ALOW,SCD,SCP,SCT
C
      GAMCAP=GAMCAP/SCP  $  SNDSPD=SNDSPD*SCD/SCT  $  ALOW=ALOW*SCP
      SCD=1.  $  SCP=1.  $  SCT=1.
      DO 15 KA=1,3
15    CP(KA)=PARS(KA)*SCPS*SCDS**KA
      CP(4)=PARS(4)*SCTS
C  THIS TRANSFORMED /CF2DER/ INTO SI UNITS
C
      CALL ROMBIN(SHTINT,ALOW,R,T,NBAD)
      IF(NBAD.EQ.0) GO TO 30
      PRINT 20,NBAD
   20 FORMAT(1H ,*RETURN FROM SHOCK2 WITH NBAD= *,I5)
   30 CONTINUE
C
      P=((CP(3)/R+CP(2))/R+CP(1))/R
      PR=-((3.*CP(3)/R+2.*CP(2))/R+CP(1))/R**2
      PRR=((12.*CP(3)/R+6.*CP(2))/R+CP(1))/R**3
      T=T/SNDSPD+CP(4)
      SQ=1.+GAMCAP*P
      TR=1./(SQRT(SQ)*SNDSPD)
      TRR=-0.5*GAMCAP*TR*PR/SQ
      RETURN
      END
```

```
      SUBROUTINE SHTINT(X,F,NBAD)
C  INTEGRAND FOR SHOCK ARRIVAL TIME COMPUTATION
C
      COMMON/CF2DER/GAMCAP,SNDSPD,PAR(4),ALOW,SCD,SCP,SCT
C
      IF(X.GT.1.E-10) GOTO 15  $  NBAD=1  $  RETURN
15    SQ=1.+GAMCAP*((PAR(3)/X+PAR(2))/X+PAR(1))/X
      IF(SQ.GT.1.E-100) GOTO 25  $  NBAD=2  $  RETURN
25    F=1./SQRT(SQ)  $  NBAD=0
      RETURN
      END
```

```
      SUBROUTINE ROMBIN (F,A,B,FINT,NBAD)
C  ROMBERG INTEGRATION SUBROUTINE
C
      DIMENSION T(10,20),CORKM(10)
C
      NBAD=0
      CALL F(A,FA,NBAD) $ IF(NBAD.NE.0)RETURN
      CALL F(B,FB,NBAD) $ IF(NBAD.NE.0)RETURN
      T(1,1)=(FA+FB)*0.5
      KM=1 $ KMA=1
C
   15 DEN=FLOAT(KMA)*2. $ FM=0
      DO 25 KA=1,KMA
      AC=FLOAT(1+2*(KMA-KA))/DEN
      BC=FLOAT(2*KA-1)/DEN
      ARG=AC*A+BC*B
      CALL F(ARG,FN,NBAD) $ IF(NBAD.NE.0)RETURN
      FM=FM+FN
   25 CONTINUE
      FM=FM/FLOAT(KMA)
      T(1,KM+1)=(T(1,KM)+FM)*0.5
C  THIS IS TRAPEZ. NOW COMPUTE ROMBERG
      KM=KM+1 $ KC=1 $ DDEN=1.
   35 KC=KC+1 $ DDEN=DDEN*4.
      CORKM(KC)=(T(KC-1,KM)-T(KC-1,KM-1))/(DDEN-1.)
      T(KC,KM)=T(KC-1,KM)+CORKM(KC)
      IF(KC.LT.KM.AND.KC.LT.10)GOTO 35
      IF(KC.GE.3)GOTO 45
C  AFTER AT LEAST 3 STEPS BRANCH TO 45 AND TEST CONVERGENCE
      KMA=KMA*2  $  GOTO 15
C
   45 DO 55 KA=2,KC
      TEST=ABS(CORKM(KA))
      IF(TEST.LE.ABS(T(KC,KM))*1.E-10)GOTO 65
      IF(TEST.LE.1.E-100)GOTO 65
   55 CONTINUE
      IF(KM.GE.20)GOTO 65
C  COMPUTE NOT MORE THAN 20 ROMBERG CORRECTIONS
      KMA=KMA*2  $  GOTO 15
C
   65 FINT=T(KC,KM)*(B-A)
      RETURN
      END
```

```
      SUBROUTINE PRTFLD(SCDIS,SCPRE,SCTIM,TITLE,PRLAB,PRDSD,
     A TARD,PIND,X,R,ALAB,NR,C)
C  THIS IS CALLED FROM FTPFLD TO PRINT PRESSURE FIELD ADJUSTMENT RESULT

      DIMENSION PRLAB(50),PRDSD(50),TARD(50),PIND(50)
      DIMENSION X(3,1),R(3,3,1),ALAB(2,1),C(3,1),TITLE(3)

      LEVEL 2,X,R,ALAB,C
C
      COMMON/TPINDX/IT,IP
C  /TPINDX/ IS SET BY FTPFLD
C  TIME=X(IT)  , OVERPRESSURE=X(IP),  DISTANCE=X(3)
C
      KH =0 $ KHIS=1
      DO 200 KA=1,NR
      KH=KH+1
C  KH COUNTS OBSERVATION SETS WITHIN THIS HISTORY
      IF(MOD(KH,40).NE.1) GOTO 28

      PRINT 10,(TITLE(J),J=1,3),PRDSD(KHIS),PIND(KHIS),TARD(KHIS)
   10 FORMAT(1H1,/1H ,15X,5HEVENT,5X,3A10,40X,21HHISTORY DISTANCE    = ,
     A 1PE10.3,3H  M,/,1H ,15X,5(1H-),75X,21HSHOCK OVERPRESSURE = ,
     B 1PE10.3,4H  PA,/,1H , 95X,21HSHOCK ARRIVAL TIME = ,1PE10.3,
     C 3H  S)
      PRINT 15
   15 FORMAT(1H ,/,1H ,40X,31HJOINT FITTING OF ALL SPECIFIED ,
     A 22HOVERPRESSURE HISTORIES,/)
      PRINT 20
   20 FORMAT(1H ,26X,3(5X,8HOBSERVED,4X,8HSTANDARD,3X,8HLST. SQ.),/
     A 1H ,2X,2HNR,10X,6HLABELS,13X,4HTIME,7X,5HERROR,4X,
     B 10HCORRECTION,2X,12HOVERPRESSURE,3X,5HERROR,4X,10HCORRECTION,
     C 4X,8HDISTANCE,5X,5HERROR,4X,10HCORRECTION)
      IF(SCTIM.EQ.1.)PRINT 21
   21 FORMAT(1H ,34X,3H(S),8X,3H(S),8X,3H(S))
      IF(SCTIM.NE.1.)PRINT 22
   22 FORMAT(1H ,33X,5H(SCT),6X,5H(SCT),6X,5H(SCT))
      IF(SCPRE.EQ.1.)PRINT 23
   23 FORMAT(1H+,68X,4H(PA),8X,4H(PA),7X,4H(PA))
      IF(SCPRE.NE.1.)PRINT 24
   24 FORMAT(1H+,69X,5H(SCP),6X,5H(SCP),6X,5H(SCP))
      IF(SCDIS.EQ.1.)PRINT 25
   25 FORMAT(1H+,105X,3H(M),9X,3H(M),7X,3H(M))
      IF(SCDIS.NE.1.)PRINT 26
   26 FORMAT(1H+,104X,5H(SCD),7X,5H(SCD),6X,5H(SCD))

C  THIS PRINTED HEADLINE.  NEXT PRINT A DATA LINE
   28 R1=SQRT(R(IT,IT,KA))
      R2=SQRT(R(IP,IP,KA))
      R3=SQRT(R(3,3,KA))
      IF(MOD(KH-1,5).EQ.0) PRINT 30
   30 FORMAT(1H )
      PRINT 40,KA,ALAB(1,KA),ALAB(2,KA),X(IT,KA),R1, C(IT,KA),
     A X(IP,KA),R2, C(IP,KA),X(3,KA),R3, C(3,KA)
   40 FORMAT(1H ,I4,2X,2A10,1P,3(3X,E11.4,1X,E10.3,1X,E10.3))
      IF(KA.EQ.NR) GOTO 55
      IF(ALAB(1,KA).EQ.ALAB(1,KA+1)) GOTO 50
      KHIS=KHIS+1
```

```
C  KHIS COUNTS HISTORIES
      KH=0
      GOTO 55
50    IF(MOD(KH,40).NE.0) GOTO 200
   55 IF(SCTIM.EQ.1..AND.SCPRE.EQ.1..AND.SCDIS.EQ.1.)GOTO 200
      PRINT 65,SCTIM,SCPRE,SCDIS
   65 FORMAT(1H ,//,1H ,31X,31HTHE DATA ARE SCALED AS FOLLOWS:,5X,
     A 16HTIME        SCT = ,1PE12.5,3H  S,/,1H ,67X,
     B 16HPRESSURE  SCP = ,1PE12.5,4H  PA,/,1H ,67X,
     C 16HDISTANCE  SCD = ,1PE12.5,3H  M)
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLTLOC(PRDS,TAR,TEND,NRPROF,PAR,VPAR,NP,
     A SCDIS,SCPRES,SCTIME ,SHOCK,TITLE)
C  THIS ROUTINE PLOTS IN THE X,T-PLANE THE SHOCK TRAJECTORY, THE
C      LOCATIONS OF OBSERVED HISTORIES AND SOME STREAMLINES.
C
C  PRDS(50)         = HISTORY DISTANCES
C  TAR(50)          = SHOCK ARRIVAL TIMES
C  TEND(50)         = HISTORY END TIMES
C  NRPROF           = NUMBER OF HISTORIES OBSERVED
C  PAR(10)          = PRESSURE FIELD PARAMETERS
C  VPAR(10,10)      = VARIANCE-COVARIANCE MATRIX OF PAR
C  NP               = NUMBER OF PRESSURE FIELD PARAMETERS
C  SCDIS, SCPRE, SCTIME  = PRESSURE, DISTANCE AND TIME SCALE
C  SHOCK            = SUBROUTINE THAT COMPUTES SHOCK (IN SI UNUTS)
C  TITLE(3)         = NAME OF EVENT TO BE USED ON PLOTS
C
      COMMON/AMBCHA/ AIRPR,AIRTEM,AIRGAM,AIRMOL,CHARVO,CHAREN
      DIMENSION PRDS(50),TAR(50),TEND(50),TEMP(8),TITLE(3)
      DIMENSION XSH(100),YSH(100),X(3),Y(3)
      DIMENSION PAR(10),VPAR(10,10),SOLIN(6),VSOL(6,6,100)
      DIMENSION STRM(6,100)
      DIMENSION XPP(10),UPP(10),UPTP(10),DPIN(10),TPIN(10)
      COMMON/CSCALE/SCDI,SCPR,SCTI
      EXTERNAL PFIELD
      COMMON/PLOT/DUM(6),PLABL(4)
C
      SCDI=SCDIS $ SCPR=SCPRES $ SCTI=SCTIME
      RIN=PRDS(1) $ R=PRDS(1) $ TMAX=TEND(1) $ TMIN=TAR(1)
      DO 5 KA=2,NRPROF
      RIN=AMIN1(PRDS(KA),RIN)
      R=AMAX1(PRDS(KA),R)
      TMIN=AMIN1(TAR(KA),TMIN)
      TMAX=AMAX1(TEND(KA),TMAX)
    5 CONTINUE
C
C  NEXT COMPUTE SHOCK TRAJECTORY
C
      RMIN=RIN
      RMAX=R
      DELR=(RMAX-RMIN)/99.
      DO 10 KA=1,100
      R1=RMIN+FLOAT(KA-1)*DELR
      RINDIM=R1*SCDIS
      XSH(KA)=R1
      CALL SHOCK(RINDIM,TDIM,POVDIM,USDIM,UPDIM,RHODIM,LBAD)
      IF(LBAD.EQ.0) GO TO 12
      NBAD=LBAD
      PRINT 14,NBAD
   14 FORMAT(1H ,*RETURN FROM PLTLOC 14 WITH NBAD= *,I10)
   12 CONTINUE
      YSH(KA)=TDIM/SCTIME
   10 CONTINUE
C
C  NEXT PLOT SHOCK TRAJECTORY AND LABEL AXES
C
      CALL PLTBEG(8.7,11.2,1.0,13,PLABL)
      CALL FIXSCA(XSH,100,5.0,XS,XMIN,XMAX,DX)
```

```
      X(1)=RMIN-0.02*(RMAX-RMIN)
      X(2)=RMAX+0.02*(RMAX-RMIN)
      X(3)=RMIN*1.01
      CALL CONSCA(X,3,5.0,XS,XMIN,XMAX,DX)
      CALL FIXSCA(YSH,100,4.0,YS,YMIN,YMAX,DY)
      Y(1)=TMIN-0.02*(TMAX-TMIN)
      Y(2)=TMAX+0.02*(TMAX-TMIN)
      CALL CONSCA(Y,2,4.0,YS,YMIN,YMAX,DY)
      CALL PLTSCA(2.5,4.0,XMIN,YMIN,XS,YS)
      CALL PLTAXS(DX,DY,XMIN,XMAX,YMIN,YMAX,4)
      CALL LABAX(DX,2.0*DY,XMIN,XMAX,YMIN,YMAX)
      CALL PLTWND(XMIN,XMAX,YMIN,YMAX)
      TX=(XMAX+XMIN)*0.5-15.0*0.1*XS
      TY=YMAX+0.5*YS
      ENCODE(80,15,TEMP)TITLE
   15 FORMAT(3A10,1H>)
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)
      ENCODE(80,20,TEMP)
   20 FORMAT(13HDISTANCE (M)>)
      IF(SCDIS.NE.1.0)ENCODE(80,21,TEMP)
21    FORMAT(15HDISTANCE (SCD)>)
      TX=(XMAX+XMIN)*0.5-6.0*0.1*XS
      TY=YMIN-0.5*YS
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)
      ENCODE(80,30,TEMP)
   30 FORMAT(9HTIME (S)>)
      IF(SCTIME.NE.1.0) ENCODE(80,31,TEMP)
31    FORMAT(11HTIME (SCT)>)
      TX=XMIN-0.7*XS
      TY=(YMIN+YMAX)*0.5-4.0*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.0,TX,TY)
      CALL PLTDTS(1,0,XSH,YSH,100,0)
C
C  NEXT PLOT HISTORY LOCATIONS
C
      DO 40 KA=1,NRPROF
      X(1)=PRDS(KA)
      X(2)=X(1)     $  X(3)=X(1)
      Y(1)=TAR(KA)
      Y(2)=TEND(KA)    $ Y(3)=Y(1)
      CALL PLTDTS(1,0,X,Y,3,0)
   40 CONTINUE
C
C  NEXT COMPUTE AND PLOT STREAMLINES
C
      AIRPRSC=AIRPR/SCPRES
      DR=0.2*(R-RIN)
      DO 1000 I=1,5
C  IN THIS LOOP COMPUTE 5 STREAMLINES
      D=RIN+DR*(I-1)
      SOLIN(3)=D
      CALL STRBEG(SOLIN,TPIN,XPP,UPP,UPTP,DPIN,LBAD)
      IF(LBAD.EQ.0) GO TO 700
      NBAD=LBAD+100
      PRINT 690,NBAD
  690 FORMAT(1H ,*ERROR RETURN IN PLTLOC 690 WITH NBAD= *,I10)
      GOTO 1000
```

```
  700 CONTINUE
      NSTMAX=100
      DELTST=(TMAX-SOLIN(1))/80.
C  THERE WILL BE AT LEAST NSTMAX/2 NODES. NORMALLY THERE WILL BE 80 TO
C  WITH THIS DELTST
      CALL STRLIN(TMAX,AIRPRSC,AIRGAM,PFIELD,PAR,VPAR,NP,SOLIN,TPIN,
     1 XPP,UPP,UPTP,DPIN,DELTST,STRM,VSOL,NSTMAX,LBAD)
      IF(LBAD.EQ.0) GO TO 900
      NBAD=LBAD+300
      PRINT 290,NBAD
  290 FORMAT(1H ,*ERROR RETURN IN PLTLOC 290 WITH NBAD= *,I10)
      IF(NSTMAX.LE.1) GOTO 1000

900   DO 70 KA=1,NSTMAX
      XSH(KA)=STRM(3,KA)
      YSH(KA)=STRM(1,KA)
      IF(KA.LT.NSTMAX/2) GOTO 70
      SD=(XSH(KA)-XSH(KA-1))/(YSH(KA)-YSH(KA-1))-
     A (XSH(KA-1)-XSH(KA-2))/(YSH(KA-1)-YSH(KA-2))
      IF(SD.GT.0.) GOTO 74
C  THIS TESTS FOR POSITIVE CURVATURE OF STREAMLINE AND PREVENTS
C  THE PLOTTING OF NONSENSICAL TAIL OF STREAMLINE
   70 CONTINUE

      GOTO 75
74    NSTMAX=KA
75    CALL PLTDTS(1,0,XSH,YSH,NSTMAX,0)
 1000 CONTINUE
      CALL PLTPGE
      RETURN
      END
```

```
      SUBROUTINE STRBEG(SOLIN,  TPIN,XPP,UPP,UPTP,DPIN,NBAD)
C
C  THIS COMPUTES THE INITIAL STREAMLINE NODE ON THE SHOCK AND  ITS
C  ACCURACY. THE SOLIN COMPONENTS ARE
C      (T, P, R, U, RHO, U**2*RHO/2)
C  THE GIVEN ARGUMENT IS THE SHOCK DISTANCE R=SOLIN(3).
C  R IS ASSUMED TO BE CONSISTENT WITH THE SCALES IN /CSCALE/
C  TPIN,XPP,UPP,UPTP AND DPIN ARE INITIAL STREAMLINE VARIABLE
C  DERIVATIVES WITH RESPECT TO THE PARAMETERS
C
C  ROUTINR USES F2SHCK
C
      DIMENSION SOLIN(6),TPIN(10)    ,XPP(10),UPP(10),UPTP(10),DPIN(10)
      DIMENSION X(5,1),PAR(10),FX(5),FP(10),FXX(5,5),FXP(5,10),
     A FPP(10,10),SOLMAT(6,4),SCALE(4)
C
      COMMON/CSCALE/SCD,SCP,SCT
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCDC,SCPC,SCTC
      COMMON/AMBCHA/PZ,TZ,GZ,AMZ,VCH,ENCH,HCH,EHCH
      COMMON/COMSHK/NPS,PARS(4),VPARS(4,4),SCDS,SCPS,SCTS
C
      DO  25 KA=1,3
25    SCALE(KA)=(SCPS/SCP)*(SCDS/SCD)**KA
      SCALE(4)=SCTS/SCT
      DO 45 KA=1,4       $  PAR(KA)=SCALE(KA)*PARS(KA)
45    CPAR(KA)=PAR(KA)
C  THE NEW PARAMETERS ARE SCALED ACCORDING TO /CSCALE/
C
      SNDSPD=SNDSPD*(SCT/SCTC)*(SCDC/SCD)
      GAMCAP=GAMCAP*(SCP/SCPC)
      ALOW=ALOW*(SCDC/SCD)
      SCDC=SCD  $  SCPC=SCP  $  SCTC=SCT
C  THIS TRANSFORMED /CF2DER/ INTO /CSCALE/ UNITS
C
      R=SOLIN(3)
C  NEXT COMPUTE SHOCK ARRIVAL TIME
      X(1,1)=0.  $  X(2,1)=R  $  X(3,1)=0.
      CALL F2SHCK(X,1,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
      IF(NBAD.NE.0) RETURN
C
      POV=((PAR(3)/R+PAR(2))/R+PAR(1))/R
      USH=SNDSPD*SQRT(1.+GAMCAP*POV)
C  SHOCK VELOCITY
      ROSI=(AMZ/8.3143)*(PZ/TZ)
C  ROSI IS AMBIENT DENSITY IN SI UNITS
      RAMB=ROSI*(SCD/SCT)**2/SCP
C  AMBIENT DENSITY IN /CSCALE/ UNITS
C
      UPSH=POV/(RAMB*USH)
C  PARTICLE VELOCITY AT THE SHOCK
      GAMTIL=((GZ-1.)/(2.*GZ*PZ))*SCP
      ROSH=RAMB*(1.+GAMCAP*POV)/(1.+GAMTIL*POV)
C  DENSITY AT THE SHOCK
      DPSH=UPSH**2*ROSH*0.5
C  DYNAMIC PRESSURE AT THE SHOCK (=SPECIFIC KINETIC ENERGY)
      SOLIN(1)=F/SNDSPD
      SOLIN(2)=POV
```

```
      SOLIN(4)=UPSH
      SOLIN(5)=ROSH
      SOLIN(6)=DPSH
C
C  NEXT COMPUTE INFLUENCE MATRIX SOLMAT WHICH EXPRESSES THE
C  RELATION BETWEEN SOLIN AND THE PARAMETER VARIANCES VPARS
      DUM=1.+GAMCAP*POV
      UPFACT=UPSH*(1./POV-0.5*GAMCAP/DUM)
      ROFACT=1./(SNDSPD**2*DUM*(1.+GAMTIL*POV))
      DPFACT=(UPSH**2*ROFACT+2.*UPSH*ROSH*UPFACT)*0.5
      DO 65 KA=1,3
65    SOLMAT(2,KA)=1./R**KA
      SOLMAT(2,4)=0.
      DO 75 KA=1,4
      SOLMAT(1,KA)=FP(KA)/SNDSPD
      SOLMAT(3,KA)=0.
      SOLMAT(4,KA)=UPFACT*SOLMAT(2,KA)
      SOLMAT(5,KA)=ROFACT*SOLMAT(2,KA)
75    SOLMAT(6,KA)=DPFACT*SOLMAT(2,KA)
C
      DO 105 KA=1,10 $ XPP(KA)=0 $ UPP(KA)=0 $ TPIN(KA)=0 $ DPIN(KA)=0
105   UPTP(KA)=0
      POVR=-((3.*PAR(3)/R+2.*PAR(2))/R+PAR(1))/R**2
      UPT=-POVR/ROSH
C  DU/DT OF PARTICLE VELOCITY AT SHOCK
      DO 115 KA=1,3
      TPIN(KA+5)=SOLMAT(1,KA)
      DPIN(KA+5)=(ROFACT/ROSH-1./(GZ*(POV+PZ/SCP)))*SOLMAT(2,KA)
      UPP(KA+5)=SOLMAT(4,KA)
115   UPTP(KA+5)=UPT*(-SOLMAT(5,KA)/ROSH+FLOAT(-KA)/(R**(KA+1)*POVR))
      TPIN(9)=SOLMAT(1,4)
      RETURN
      END
```

```
      SUBROUTINE SHODER(      R,T,TR,TP,TRR,TRP,TPP,
     APOV,PR,PP,PRR,PRP,PPP,NBAD)
C  THIS COMPUTES FOR GIVEN DISTANCE R THE CORRESPONDING
C  SHOCK TIME T AND OVERPRESSURE POV, AND DERIVATIVES
C  SUBROUTINE USES F2SHCK TO COMPUTE SHOCK ARRIVAL TIME
C  ALL ARGUMENTS ARE ASSUMED TO BE IN SI UNITS
C
      DIMENSION TP(10),TRP(10),TPP(10,10),PP(10),PRP(10),PPP(10,10),
     A        SPAR(10),X(5,1),FX(5),FP(10),FXX(5,5),FXP(5,10),FPP(10,10
C
      COMMON/COMSHK/NPS,PARSH(4),VPARSH(4,4),SCDIS,SCPRE,SCTIM
      COMMON/CF2DER/GAMCAP,SNDSPD,PRS(4),ALOW,SCD,SCP,SCT
      GAMCAP=GAMCAP/SCP  $  SNDSPD=SNDSPD*SCD/SCT  $  ALOW=ALOW*SCD
      SCD=1. $  SCP=1.  $  SCT=1.
C  THIS CHANGED  /CF2DER/  TO SI UNITS
      IF(NPS.GE.0.AND.NPS.LE.5)GOTO 15
C  THIS IS CODED FOR NPS = NUMBER OF SHOCK PARAMETERS = 4
      NBAD=IABS(NPS) $ RETURN
   25 NBAD=25
      PRINT 27,NBAD
   27 FORMAT(1H ,*RETURN FROM SHODER WITH NBAD= *,I5)
      RETURN
C
15    IF(R.LE.0.)GOTO 25
      NBAD=0
      IF(NPS.EQ.0)GOTO 55
C
C  NOW COMPUTE SHOCK OVERPRESSURE IN PASCALS BY 3-PARAMETER FORMULA
      DO 35 KA=1,3
35    SPAR(KA)=PARSH(KA)*SCPRE*SCDIS**KA
      SPAR(4)=PARSH(4)*SCTIM
C  SPAR IS FOR COMPUTATION OF POV IN PASCALS WHEN R IS IN METRES
C
      POV=((SPAR(3)/R+SPAR(2))/R+SPAR(1))/R
      PR=-((SPAR(3)*3./R+SPAR(2)*2.)/R+SPAR(1))/R**2
      PRR=((SPAR(3)*12./R+SPAR(2)*6.)/R+SPAR(1)*2.)/R**3
C
      DO 37 KA=1,10 $ PP(KA)=0 $ PRP(KA)=0
      TP(KA)=0 $ TRP(KA)=0
      DO 37 KB=1,10 $ TPP(KA,KB)=0
   37 PPP(KA,KB)=0
C
C  ASSUME THAT SHOCK PARAMETERS ARE NR. 6,7,8,9.
      PP(6)=1./R $ PP(7)=PP(6)/R $ PP(8)=PP(7)/R
      PRP(6)=-PP(7) $ PRP(7)=-2.*PP(8) $ PRP(8)=-3.*PP(8)/R
C  NEXT COMPUTE SHOCK ARRIVAL TIME. X(1)=PRESSURE, X(3)=TIME
      X(1,1)=0 $ X(2,1)=R $ X(3,1)=0
      CALL F2SHCK(X,1,SPAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C
      IF(NBAD.EQ.0) GO TO 40
      PRINT 38,NBAD
   38 FORMAT(1H ,*RETURN FROM SHODER AFTER F2SHCK WITH NBAD= *,I5)
      GO TO 55
   40 T=F/SNDSPD $ TR=FX(2)/SNDSPD $ TRR=FXX(2,2)/SNDSPD
C
      DO 45 KA=1,NPS $ TP(5+KA)=FP(KA)/SNDSPD
      TRP(5+KA)=FXP(2,KA)/SNDSPD
```

```
      DO 45 KB=1,NPS
   45 TPP(5+KA,5+KB)=FPP(KA,KB)/SNDSPD
C
   55 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE F2SHCK(XX,KA,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS IS SECOND CONSTRAINT COMPONENT FOR SHOCK FITTING
C
      DIMENSION XX(5,100),PAR(10),FX(5),FP(10),FXX(5,5),FXP(5,10),
     A FPP(10,10),SF(9)
      EXTERNAL F2DER
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCD,SCP,SCT
C  GAMCAP=((1.+GAM)/(2.*GAM))*(SCPR/AMBPR)
C  GAMCAP, SNDSPD AND ALOW ARE SET BY SUBROUTINE SCALSH
C
      DO 15 KB=1,4
   15 CPAR(KB)=PAR(KB)
C  THE PARAMETERS CPAR WILL BE USED BY SUBROUTINE F2DER
      X=XX(2,KA)
      DO 25 KB=1,3 $ DO 25 KC=1,3
   25 FXX(KB,KC)=0
      IF(X.GT.1.E-30) GOTO 35 $ NBAD=1 $ RETURN
C
   35 NBAD=0
      SQ=1.+GAMCAP*((PAR(3)/X+PAR(2))/X+PAR(1))/X
      IF(SQ.GT.1.E-50 ) GOTO 45 $ NBAD=2 $ RETURN
   45 FX(1)=0.  $  FX(2)=1./SQRT(SQ)  $  FX(3)=-SNDSPD
      FXX(2,2)=0.5*GAMCAP*FX(2)*((3.*PAR(3)/X+2.*PAR(2))/X
     A+PAR(1))/(X*X*SQ)
C  COMPUTE PARTS OF F2 AND DERIVATIVES BY MULTIPLE QUADRATURE
      CALL ROMULT(F2DER,ALOW,X,SF,NBAD)
      IF(NBAD.EQ.0) GOTO 55 $ NBAD=NBAD+10 $ RETURN
   55 F=SF(1)+(PAR(4)-XX(3,KA))*SNDSPD
      FP(1)=SF(2)  $  FP(2)=SF(3)  $  FP(3)=SF(4)  $  FP(4)=SNDSPD
      FPP(1,1)=SF(5) $ FPP(1,2)=SF(6) $ FPP(1,3)=SF(7)
      FPP(2,1)=SF(6) $ FPP(2,2)=SF(7) $ FPP(2,3)=SF(8)
      FPP(3,1)=SF(7) $ FPP(3,2)=SF(8) $ FPP(3,3)=SF(9)
      DO 65 KB=1,4 $ FPP(4,KB)=0 $ FPP(KB,4)=0 $ FXP(1,KB)=0
   65 FXP(3,KB)=0
      FXP(2,1)=-0.5*GAMCAP*FX(2)/(X*SQ)
      FXP(2,2)=FXP(2,1)/X $ FXP(2,3)=FXP(2,2)/X $ FXP(2,4)=0
      RETURN
      END
```

```
      SUBROUTINE F2DER(X,F,NBAD)
C  INTEGRAND FOR NINE COMPONENTS OF F2 AND DERIVATIVES
C  USED BY F2SHCK AS ARGUMENT OF ROMULT
C
      DIMENSION F(9)
      COMMON/CF2DER/GAMCAP,SNDSPD, PAR(4),ALOW,SCD,SCP,SCT
C  GAMCAP=((1.+GAM)/(2.*GAM))*(SCP /AMBPR)
C  GAMCAP, SNDSPD, ALOW AND SCALES ARE SET BY SUBROUTINE SCALSH
C
      NBAD=0 $ IF(X.GT.1.E-30) GOTO 15 $ NBAD=1 $ RETURN
C
   15 Y=1./X
      SQ=1.+GAMCAP*((PAR(3)*Y+PAR(2))*Y+PAR(1))*Y
      IF(SQ.GT.1.E-50 ) GOTO 25 $ NBAD=2 $ RETURN
C
C  INTEGRANDS CORRESPOND TO FOLLOWING QUANTITIES
C  F,FP(1),(2),(3),FPP(1,1),(1,2),(1,3)=(2,2),(2,3),(3,3)
   25 F(1)=1./SQRT(SQ)
      F(2)=-0.5*GAMCAP*F(1)*Y/SQ
      F(3)=F(2)*Y $ F(4)=F(3)*Y
      F(5)=-1.5*GAMCAP*F(3)/SQ
      F(6)=F(5)*Y $ F(7)=F(6)*Y $ F(8)=F(7)*Y $ F(9)=F(8)*Y
      RETURN
      END
```

```
      SUBROUTINE ROMULT(F,A,B,SF,NBAD)
C  ROMBERG INTEGRATION OF A 9-DIMENSIONAL VECTOR FUNCTION
C
      DIMENSION SF(9),T(9,10,20),FA(9),FB(9),FN(9),FM(9),CORKM(9,10)
C
      NBAD=0
      CALL F(A,FA,NBAD) $ IF(NBAD.NE.0) RETURN
      CALL F(B,FB,NBAD) $ IF(NBAD.NE.0) RETURN
      DO 14 KD=1,9
   14 T(KD,1,1)=(FA(KD)+FB(KD))*0.5
      KM=1 $ KMA=1
C
   15 DO 16 KD=1,9
   16 FM(KD)=0
      DEN=FLOAT(KMA)*2.
      DO 25 KA=1,KMA
      AC=FLOAT(1+2*(KMA-KA))/DEN $ BC=FLOAT(2*KA-1)/DEN
      ARG=AC*A+BC*B
      CALL F(ARG,FN,NBAD) $ IF(NBAD.NE.0) RETURN
      DO 23 KD=1,9
   23 FM(KD)=FM(KD)+FN(KD)
   25 CONTINUE
      DO 26 KD=1,9 $ FM(KD)=FM(KD)/FLOAT(KMA)
   26 T(KD,1,KM+1)=(T(KD,1,KM)+FM(KD))*0.5
C
C  THIS IS TRAPEZ. NEXT COMPUTE ROMBERG
      KM=KM+1 $ KC=1 $ DDEN=1.
C
   35 KC=KC+1 $ DDEN=DDEN*4.
      DO 37 L=1,9
      CORKM(L,KC)=(T(L,KC-1,KM)-T(L,KC-1,KM-1))/(DDEN-1.)
   37 T(L,KC,KM)=T(L,KC-1,KM)+CORKM(L,KC)
      IF(KC.LT.KM.AND.KC.LT.10) GOTO 35
C
      IF(KM.GE.3) GOTO 45 $ KMA=KMA*2 $ GOTO 15
C  AFTER THREE STEPS TEST CONVERGENCE
C
   45 IF(KM.GE.20) GOTO 56
C  MAXIMUM OF 20 STEPS ALLOWED
C
      DO 53 L=1,9
      TEST=ABS(CORKM(L,KC))
C  KC=MIN(KM,10)
      IF(TEST.LE.1.E-100) GOTO 53
      IF(TEST.LE.ABS(T(L,KC,KM))*1.E-10) GOTO 53
      KMA=KMA*2 $ GOTO 15
   53 CONTINUE
C
   56 DO 58 L=1,9
   58 SF(L)=T(L,KC,KM)*(B-A)
      RETURN
      END
```

```
      SUBROUTINE STRLIN(TMAX,AIRPR,AIRGAM,PFIELD,PAR,VPAR,NPAR,SOLIN,
     A TPIN,XPP,UPP,UPTP,DPIN,DT,SLINA,VSLINA,NMAXA,NBAD)
C  THIS COMPUTES A STREAMLINE STARTING WITH SPECIFIED INITIAL
C  VALUES AND ENDING AT TMAX
C
C  TMAX               = TIME AT END POINT OF STREAMLINE. THE ACTUAL TIME
C                       CAN BE BY DT LARGER THAN TMAX
C  AIRPR              = AMBIENT PRESSURE
C  AIRGAM             = RATIO OF SPECIFIC HEATS
C  PFIELD             = PRESSURE FIELD SUBROUTINE
C  PAR,VPAR,NPAR      = PARAMETERS, THEIR VARIANCE AND NUMBER FOR PFIELD
C  SOLIN(6)           = INITIAL VALUES ON STREAMLINE, VIZ.
C                       TIME, PRESSURE, DISTANCE, VELOCITY, DENSITY,
C                       DYNAMIC PRESSURE (= KINETIC ENERGY DENSITY)
C  TPIN(10)           = D/DPAR OF THE INITIAL TIME
C  XPP(10)            = D/DPAR OF INITIAL POSITION
C  UPP(10)            = D/DPAR OF INITIAL PARTICLE VELOCITY
C  UPTP(10)           = D/DPAR OF INITIALL PARTICLE ACCELERATION
C  DPIN(10)           = D/DPAR EXPRESSION NEEEDED FOR INTEGRATION OF UPP
C  DT                 = TIME INTERVAL FOR INTEGRATION
C
C  THE FOLLOWING WILL BE COMPUTED BY THIS ROUTINE
C
C  SLINA(6,NMAXA)   = FLOW VARIABLES ALONG THE STREAMLINE (T,P,R,U,RHO,U**2*
C  VSLINA(6,6,NMAXA)= VARIANCE-COVARIANCE MATRIX OF SLINA
C  NMAXA              = MAXIMUM NUMBER OF NODES IN SLINE
C                       WILL BE REPLACED BY ACTUAL NUMBER COMPUTED
C  NBAD               = ERROR INDICATOR
C
      DIMENSION PAR(10),VPAR(10,10),SOLIN(6),TPIN(10),XPP(10),UPP(10),
     A UPTP(10),DPIN(10),SLINA(6,100),VSLINA(6,6,100)
C
      COMMON/SCRCH3/ X(3,1),FX(3),FP(10),FXP(3,10),FXX(3,3),FPP(10,10)
      LEVEL 2,X,FX,FP,FXP,FXX,FPP
      COMMON/TPINDX/IT,IP
C  /TPINDX/ IS SET BY FTPFLD
C  TIME=X(IT)   , OVERPRESSURE=X(IP),   DISTANCE=X(3)
C
      DIMENSION UT(2),XP(2,10),UTP(2,10),UP(2,10),SOLMAT(6,10)
     A,U(2),UTT(2),SLINE(6,51),VSLINE(6,6,51)
C  SLINE AND VSLINE ARE WORKING AREAS WITH LENGTH NMAX
      DATA (NMAX=51)
C
      NBAD=0
      DO 9 KA=1,6
      SLINE(KA,1)=SOLIN(KA)
    9 SLINA(KA,1)=SOLIN(KA)
      IF(NMAXA.GT.2)GOTO 12
      NMAXA=0
      NBAD=11 $ PRINT 11, NBAD $ RETURN
   11 FORMAT(1H0,10X,30HRETURN FROM STRLIN WITH NBAD =,I4)
   12 IF(DT.GT.0.) GOTO 15
      IF(SLINA(1,1).GE.TMAX) GOTO 15
      NMAXA=0
      NBAD=12 $ PRINT 11,NBAD $ RETURN
C  DT IS PERMITTED TO BE ZERO FOR ONE POINT STREAMLINE
   15 IF(SOLIN(3).GT.0.) GOTO 25
```

```
C  CHECK FOR NEGATIVE INITIAL DISTANCE
      NMAXA=0
      NBAD=15 $ PRINT 11, NBAD $ RETURN
   25 CONTINUE
      ROZ=SOLIN(5) $ GEXP=1./AIRGAM $ PRZ=SOLIN(2)+AIRPR
      DO 31 I=1,2
      DO 30 KA=1,NPAR $ XP(I,KA)=XPP(KA) $ UP(I,KA)=UPP(KA)
   30 UTP(I,KA)=UPTP(KA)
   31 CONTINUE
C
      X(IT,1)=SLINE(1,1)  $  X(IP,1)=0.0  $  X(3,1)=SLINE(3,1)
C     TIME           PRESSURE          DISTANCE
      CALL PFIELD(X,1,PAR,F,FX,FP,FXX,FXP,FPP,LBAD)
 3500 IF(LBAD.EQ.0) GOTO 39
      NMAXA=0
      NBAD=3500+LBAD $ PRINT 11, NBAD $ RETURN
C
   39 UT(1)=-FX(3)*(PRZ/(F+AIRPR))**GEXP/ROZ
C  DU/DT=-(DP/DR)*(PO/P)**(1/GAMMA)/RHOZERO
      U(1)=SLINE(4,1)
      UTT(1)=UT(1)*(-GEXP*(FX(IT)+U(1)*FX(3))/(F+AIRPR)
     A +(FXX(IT,3)+U(1)*FXX(3,3))/FX(3) )
      DTSTOR=DT $ TSTOR=SLINA(1,1)+DTSTOR $ KT=1
C  COMPUTATION RESULTS WILL BE STORED APROXIMATELY FOR TSTOR
C  KT COUNTS STORAGE IN SLINA AND VSLINA
C  THIS IS ACTUAL INTEGRATION INTERVAL. WITH DTS=0 GET FIRST NODE
      DTS=0.
      KA=1
C
C     NEXT STATEMENT IS BEGINNING OF INTEGRATION LOOP
   45 SLINE(3,KA+1)=SLINE(3,KA)+DTS*(U(1)+0.5*DTS*(UT(1)+DTS*UTT(1)/3.))
C  NEW DISTANCE BY FOURTH ORDER FORMULA IN DTS
      SLINE(1,KA+1)=SLINE(1,KA)+DTS
C  NEW TIME
      DO 47 KB=1,NPAR
   47 XP(2,KB)=XP(1,KB)+DTS*(UP(1,KB)+0.5*DTS*UTP(1,KB))
C  NEW DX/DPARAMETER. THIRD ORDER ERROR IN DTS
C
      X(IT ,1)=SLINE(1,KA+1)  $  X(IP,1)=0.0  $  X(3,1)=SLINE(3,KA+1)
      CALL PFIELD(X,1,PAR,F,FX,FP,FXX,FXP,FPP,LBAD)
      IF(LBAD.EQ.0) GOTO 55
 5100 NBAD=5100+LBAD $ PRINT 11, NBAD
      KT=KT-1  $  GOTO 155
C
   55 SLINE(2,KA+1)=F
C  NEW PRESSURE
      UT(2)=-FX(3)*(PRZ/(F+AIRPR))**GEXP/ROZ
      U(2)=U(1)+0.5*DTS*(UT(1)+UT(2))
C  FIRST APPROXIMATION OF NEW VELOCITY. THIRD ORDER ERROR IN DTS
      UTT(2)=UT(2)*(-GEXP*(FX(IT)+U(2)*FX(3))/(F+AIRPR)
     A +(FXX(IT,3)+U(2)*FXX(3,3))/FX(3) )
      U(2)=U(2)+(UTT(1)-UTT(2))*DTS**2/12.
C  NEW VELOCITY. FIFTH ORDER ERROR IN DTS
      SLINE(4,KA+1)=U(2)
      DO 65 KB=1,NPAR
      UTP(2,KB)=UT(2)*(-DPIN(KB)
     A -(FP(KB)+FX(3)*XP(2,KB))*GEXP/(F+AIRPR)
```

```
      B +(FXP(3,KB)+FXX(3,3)*XP(2,KB))/FX(3))
       UP(2,KB)=UP(1,KB)+0.5*DTS*(UTP(1,KB)+UTP(2,KB))
   65 CONTINUE
C  NEW DU/DPARAMETER. THIRD ORDER ERROR IN DTS
       SLINE(5,KA+1)=ROZ*((F+AIRPR)/PRZ)**GEXP
C  NEW DENSITY
       SLINE(6,KA+1)=0.5*SLINE(5,KA+1)*SLINE(4,KA+1)**2
C  NEW DYNAMIC PRESSURE
C
C  NEXT COMPUTE VARIANCE ESTIMATES OF SOLUTION
       DO 75 KB=1,NPAR
       SOLMAT(1,KB)=TPIN(KB)
       SOLMAT(2,KB)=FP(KB)+FX(3)*XP(2,KB)
       SOLMAT(3,KB)=XP(2,KB)
       SOLMAT(4,KB)=UP(2,KB)
       SOLMAT(5,KB)=SLINE(5,KA+1)*(DPIN(KB)
     A +GEXP*(FP(KB)+FX(3)+XP(2,KB)+FX(IT)*SOLMAT(1,KB))/(F+AIRPR) )
       SOLMAT(6,KB)=0.5*SLINE(4,KA+1)*(SLINE(5,KA+1)*SOLMAT(4,KB)*2.
     A +SLINE(4,KA+1)*SOLMAT(5,KB))
   75 CONTINUE
C  SOLMAT IS THE JACOBIAN MATRIX DSLINE/DPARAMETER
       DO 95 KB=1,6 $ DO 95 KC=1,6
       VSLINE(KB,KC,KA+1)=0.
       DO 85 KD=1,NPAR  $  DO 85 KE=1,NPAR
       VSLINE(KB,KC,KA+1)=VSLINE(KB,KC,KA+1)+
     A SOLMAT(KB,KD)*VPAR(KD,KE)*SOLMAT(KC,KE)
   85 CONTINUE
   95 CONTINUE
C
C  NOW STORE RESULTS IF TSTOR REACHED
       KA=KA+1
       IF(KT.EQ.1)GOTO 97
       IF(SLINE(1,KA).LT.TSTOR-DTS*0.2)GOTO 125
   97 DO 99 KB=1,6 $ DO 98 KC=1,6
   98 VSLINA(KB,KC,KT)=VSLINE(KB,KC,KA)
   99 SLINA(KB,KT)=SLINE(KB,KA)
C
       IF(SLINA(1,KT).GE.TMAX)GOTO 155
C  BRANCH TO 155 WHEN END OF STREAMLINE REACHED
       TSTOR=SLINA(1,KT)+DTSTOR
C  TIME VALUE FOR NEXT NODE TO BE STORED IN SLINA
       KT=KT+1 $ DTS=DT*0.2
C  AFTER FIRST NODE CONTINUE WITH DTS.GT.0.
C
       IF(KT.LT.NMAXA)GOTO 115
C
C  THIS IS PROGRAMMING ERROR. WITH GIVEN DT END TIME CANNOT
C  BE REACHED IN NMAXA STEPS. CORRECT BY INCREASING DT
       DTSTOR=DTSTOR*2.
C  ELIMINATE HALF OF STORED RESULTS
       KC=2 $ KB=3
  102 DO 104 KD=1,6 $ DO 103 KE=1,6
  103 VSLINA(KD,KE,KC)=VSLINA(KD,KE,KB)
  104 SLINA(KD,KC)=SLINA(KD,KB)
       KC=KC+1 $ KB=KB+2
       IF(KB.LE.NMAXA)GOTO 102
       KT=KC-1 $ TSTOR=SLINA(1,KT)+DTSTOR
```

```
      GOTO 125
C
  115 IF(KT.LE.2)KA=1
C
  125 IF(KA.LT.NMAX)GOTO 145
C
C  NOW WORK AREA IS OVERFLOWING. ELIMINATE OLD STUFF
      KC=2 $ KB=3
  131 DO 133 KD=1,6 $ DO 132 KE=1,6
  132 VSLINE(KE,KD,KC)=VSLINE(KE,KD,KB)
  133 SLINE(KD,KC)=SLINE(KD,KB)
      KC=KC+1 $ KB=KB+2
      IF(KB.LE.NMAX)GOTO 131
      KA=KC-1 $ IF(KB.EQ.NMAX+1) GOTO 45
C
C  PREPARE FOR NEXT INTEGRATION STEP
  145 U(1)=U(2) $ UT(1)=UT(2) $ UTT(1)=UTT(2)
      DO 148 KB=1,NPAR $ XP(1,KB)=XP(2,KB) $ UP(1,KB)=UP(2,KB)
  148 UTP(1,KB)=UTP(2,KB)
      GOTO 45
C
  155 NMAXA=KT
      RETURN
      END
```

```
      SUBROUTINE DIMFLD(SCD,SCP,SCT,EXNU,P,VP,ERZ,NP,
     A PDIM,VPDIM,TITLE)
C  THIS COMPUTES THE VALUES OF PRESSURE FIELD PARAMETERS
C  AND OF CORRESPONDING VARIANCES IN SI UNITS.
C  IT IS CALLED FROM MAIN AFTER PRESSURE FIELD ADJUSTMENT BY FTPFLD
C
      DIMENSION P(10),VP(10,10),PDIM(10),VPDIM(10,10),TITLE(3)
      DIMENSION EXNU(3)
      DIMENSION SCMAT(10,10),DIM(10),COR(10,10)
C
      EXA=EXNU(1) $ EXB=EXNU(2) $ EXC=EXNU(3)
      DO 15 KA=1,10 $ DO 15 KB=1,10
   15 SCMAT(KA,KB)=0
17    FORMAT(F4.1,A5)
      TX=5H/S     $ ENCODE( 9,17,DIM(1))EXA,TX
      EXA2=EXA-1.  $  ENCODE( 9,17,DIM(2)) EXA2,TX
      TX=5H/S**2 $ ENCODE( 9,17,DIM(3)) EXB,TX
      EXB2=EXB-1.  $  ENCODE( 9,17,DIM(4)) EXB2,TX
      TX=5H*PA    $ ENCODE( 9,17,DIM(5)) EXC,TX
      EXC2=EXC-1. $  ENCODE( 9,17,DIM(6)) EXC2,TX
C
      SCMAT(1,1)=SCD**EXA/SCT
      SCMAT(2,2)=SCD**(EXA-1.)/SCT
      SCMAT(3,3)=SCD**EXB/SCT**2
      SCMAT(4,4)=SCD**(EXB-1.)/SCT**2
      SCMAT(5,5)=SCD**EXC*SCP
      SCMAT(6,6)=SCD**(EXC-1.)*SCP
C
      DO 45 KA=1,NP $ PDIM(KA)=0
      DO 35 KB=1,NP $ VPDIM(KA,KB)=0
      DO 25 KC=1,NP $ DO 25 KD=1,NP
   25 VPDIM(KA,KB)=VPDIM(KA,KB)+SCMAT(KA,KC)*VP(KC,KD)*SCMAT(KB,KD)
   35 PDIM(KA)=PDIM(KA)+SCMAT(KA,KB)*P(KB)
   45 CONTINUE
C
      PRINT 50,(TITLE(J),J=1,3)
50    FORMAT(1H1,//,1H ,10X,5HEVENT,5X,3A10,/,1H ,10X,5(1H-))
      PRINT 55
   55 FORMAT(1H ,///,1H ,10X,30HDIMENSIONAL VALUES OF PRESSURE,
     A 17H FIELD PARAMETERS,/)
      PRINT 65
   65 FORMAT(1H0,10X,10HPARAMETERS,5X,8HSTANDARD,7X,8HSTANDARD,
     A5X,9HDIMENSION,/,1H ,26X,6HERRORS,7X,10HERRORS*ERZ,/)
      DO 85 KA=1,NP
      PER=SQRT(VPDIM(KA,KA)) $ PERZ=PER*ERZ
      PRINT 75,PDIM(KA),PER,PERZ,DIM(KA)
75    FORMAT(1H ,9X,1PE12.5,3X,1PE10.3,4X,1PE10.3,5X,3HM**,A9)
   85 CONTINUE
      PRINT 87,EXA,EXB,EXC
   87 FORMAT (1H ,//,1H ,10X,29HTHE EXPONENTS IN THE PRESSURE,
     A 18H FIELD FORMULA ARE,//,1H ,10X,4HNA =,0PF5.2,/,
     B 1H ,10X,4HNB =,0PF5.2,/,1H ,10X,4HNC =,0PF5.2)

C  NEXT COMPUTE AND PRINT CORRELATION MATRIX
      DO 88 KA=1,NP $ DO 88 KB=1,NP
   88 COR(KA,KB)=VP(KA,KB)/SQRT(VP(KA,KA)*VP(KB,KB))
      PRINT 89
```

```
    89 FORMAT(1H ,//,1H ,10X,26HCORRELATION MATRIX OF THE ,
      A10HPARAMETERS,//)
       DO 91 KA=1,NP
       PRINT 90,(COR(KA,J),J=1,NP)
    90 FORMAT(1H ,8X,6(2X,OPF11.8))
    91 CONTINUE

       PRINT 95,EXC,EXA,EXB,EXC
    95 FORMAT(1HO,///,1H ,10X,34HTHE OVERPRESSURE MODEL IS GIVEN BY,//,
      A1H ,20X,21HP = (PSHOCK(R)-P5/R**,F4.1,1H),
      B 25H * EXP( TAU*(P1+P2*R)/R**,F4.1,3H + ,
      C 20HTAU**2*(P3+P4*R)/R**,F4.1,2H ),9H + P5/R**,F4.1,1H,//,
      D 1H ,20X,5HWHERE,5X,17HTAU = T-TSHOCK(R),//)
C
       PRINT 105
   105 FORMAT(1H ,//,1H ,10X,27HVARIANCE-COVARIANCE MATRIX ,
      A33H(NOT INCLUDING THE FACTOR ERZ**2),//)
       DO 125 KA=1,NP
       PRINT 115,(VPDIM(KA,J),J=1,NP)
   115 FORMAT(1H ,10X,6(3X,1PE12.5))
   125 CONTINUE
       PRINT 127
127     FORMAT(1H ,///)
       RETURN $ END
```

```
      SUBROUTINE PLTFLD(TITLE,TARD,PIND,PAR,VPAR,ERZ,NP,NRPROF)
C  THIS ROUTINE PLOTS INDIVIDUAL OVERPRESSURE HISTORY OBSERVATIONS
C  AND CORRESPONDING PRESSURE FIELD FUNCTION
C  IT IS CALLED FROM FTPFLD AFTER ADJUSTMENT

C  TITLE(3)      = NAME OF THE EVENT
C  TARD(50)      = SHOCK ARRIVAL TIMES AT THE HISTORY LOCATIONS (S)
C  PIND(50)      = INITIAL SHOCK OVERPRESSURE AT HISTORY LOCATIONS (PA)
C  PAR(10)       = PRESSURE FIELD PARAMETERS
C  VPAR(10,10)   = VARIANCE-COVARIANCE MATRIX OF PAR
C  ERZ           = STANDARD ERROR WITH WEIGHT ONE
C  NP            = NUMBER OF FIELD PARAMETERS PAR
C  NRPROF        = NUMBER OF HISTORIES

C  THE ROUTINE USES PFIELD TO COMPUTE THE FITTED PRESSURE

      DIMENSION TITLE(3),TARD(50),PIND(50),PAR(10),VPAR(10,10)

      COMMON/COMPR/TP(2,5000),ERTP(2,5000),ALB(2,5000),NSET(50),
     A DIST(50),ERDIST(50)
      LEVEL 2,TP,ERTP,ALB,NSET,DIST,ERDIST
C  /COMPR/ CONTAINS INPUT TIMES AND OVERPRESSURES,
C  NSET GIVES THE NUMBER OF SETS IN EACH HISTORY,
C  DIST CONTAINS HISTORY DISTANCES

      COMMON/SCRCH2/X(3,5000),R(3,3,5000),LSTX(5000),XC(3,5000),
     A C(3,5000),WORK(14307),LSTN(5000)
      LEVEL 2,X,R,LSTX,XC,C,WORK,LSTN
C  FROM/SCRCH2/ONLY XC IS NEEDED TO PLOT
C  THE CORRECTED OVERPRESSURES AND TIMES

      COMMON/TPINDX/ITC,IPC
      COMMON/CSCALE/SCDI,SCPR,SCTI
C  /TPINDX/ AND /CSCALE/ ARE USED BY PFIELD

      COMMON/CPARG/XF(3,1),FX(3),FP(10),FXX(3,3),FXP(3,10),FPP(10,10)
      LEVEL 2,XF,FX,FP,FXX,FXP,FPP
C  THESE ARE ARGUMENTS OF PFIELD

      COMMON/PLOT/ERF,D(5),PLABL(4)
C  /PLOT/ CONTAINS CONFIDENCE FACTOR ERF AND PLOTLABEL

      DIMENSION XP(201),YP(201),RE(2,2),TEXT(10),EP(201)

      IF(ERF.LE.0.)ERF=2.0

      CALL PLTBEG(22.0,28.5,0.3937,13,PLABL)
C  PLOTTING SCALES ARE IN CENTIMETRES

      KCS=0
   15 DO 155 KH=1,NRPROF
      KSET=NSET(KH) $ IF(KSET.LE.0)GOTO 155

C  NEXT FIND EXTREMA FOR A HISTORY AND FIX SCALES
      KINT=KCS+1
      XP(1)=TP(1,KINT) $ XP(2)=XP(1)
      XP(1)=AMIN1(XP(1),TARD(KH))
```

```
      YP(1)=TP(2,KINT) $ YP(2)=AMAX1(YP(1),PIND(KH))
      DO 25 KA=1,KSET
      KC=KCS+KA
      XP(1)=AMIN1(XP(1),TP(1,KC)-ERTP(1,KC)*ERF)
      XP(2)=AMAX1(XP(2),TP(1,KC)+ERTP(1,KC)*ERF)
      YP(1)=AMIN1(YP(1),TP(2,KC)-ERTP(2,KC)*ERF)
      YP(2)=AMAX1(YP(2),TP(2,KC)+ERTP(2,KC)*ERF)
   25 CONTINUE

C  NEXT FIX SCALES
      XSIZE=12.0 $ YSIZE=10.0
      AUGX=AMAX1(XP(2)-XP(1),0.001)*0.05
      XP(3)=XP(1)-AUGX
      XP(4)=XP(2)+AUGX
      CALL FIXSCA(XP,4,XSIZE,XS,XMIN,XMAX,DX)
      AUGY=AMAX1(YP(2)-YP(1),1.E3)*0.05
      YP(3)=YP(1)-AUGY
      YP(4)=YP(2)+AUGY
      CALL FIXSCA(YP,4,YSIZE,YS,YMIN,YMAX,DY)

      CALL PLTSCA(6.0,10.0,XMIN,YMIN,XS,YS)
      CALL PLTAXS(DX,DY,XMIN,XMAX,YMIN,YMAX,4)
      CALL LABAX(DX,2.0*DY,XMIN,XMAX,YMIN,YMAX)

C  NEXT PLOT HEADLINE ETC.
      HT=0.25
      ENCODE(80,31,TEXT)
   31 FORMAT(9HTIME (S)>)
      XT=(XMIN+XMAX)*0.5-4.0*HT*XS
      YT=YMIN-YS*1.4
      CALL PLTSYM(HT,TEXT,0.0,XT,YT)
      ENCODE(80,32,TEXT)
   32 FORMAT(18HOVERPRESSURE (PA)>)
      XT=XMIN-XS*1.8
      YT=(YMIN+YMAX)*0.5-8.5*HT*YS
      CALL PLTSYM(HT,TEXT,90.0,XT,YT)
      ENCODE(80,33,TEXT)(TITLE(J),J=1,3)
   33 FORMAT(3A10,1H>)
      XT=(XMIN+XMAX)*0.5-15.0*HT*XS
      YT=YMAX+YS*2.3
      CALL PLTSYM(HT,TEXT,0.0,XT,YT)
      YT=YMAX+YS*1.5
      ENCODE(80,34,TEXT)ALB(1,KINT)
   34 FORMAT(A10,1H>)
      CALL PLTSYM(HT,TEXT,0.0,XT,YT)
      ENCODE(80,35,TEXT)
   35 FORMAT(26HFITTED OVERPRESSURE FIELD>)
      XT=(XMIN+XMAX)*0.5-12.5*HT*XS
      YT=YMIN-YS*2.5
      CALL PLTSYM(HT,TEXT,0.0,XT,YT)
      ENCODE(80,36,TEXT)
   36 FORMAT(37HCONFIDENCE LIMITS AND ERROR ELLIPSES>)
      XT=XMIN
      YT=YMIN-YS*4.0
      CALL PLTSYM(HT,TEXT,0.0,XT,YT)
      ENCODE(80,37,TEXT)ERF
   37 FORMAT(14HCORRESPOND TO ,0PF5.2,17H STANDARD ERRORS>)
```

```
         YT=YT-2.0*HT*YS
         CALL PLTSYM(HT,TEXT,0.0,XT,YT)
         ENCODE(80,38,TEXT)ERZ
      38 FORMAT(16HTHE FACTOR ERZ =,1PE9.2,17H IS NOT INCLUDED>)
         YT=YT-4.0*HT*YS
         CALL PLTSYM(HT,TEXT,0.0,XT,YT)
         ENCODE(80,39,TEXT)
      39 FORMAT(23HIN THE ERROR ESTIMATES>)
         YT=YT-2.0*HT*YS
         CALL PLTSYM(HT,TEXT,0.0,XT,YT)

         CALL PLTWND(XMIN,XMAX,YMIN,YMAX)

C     NEXT PLOT ALL OBSERVATIONS WITH ERROR ELLIPSES
         DO 45 KA=1,KSET
         KC=KCS+KA
         TC=TP(1,KC)    $    PC=TP(2,KC)
         CALL PLTDTS(3,1,TC,PC,1,0)
C     THIS PLOTTED DATA POINT
         XP(1)=TP(1,KC) $ XP(2)=XC(ITC,KC)*SCTI
         YP(1)=TP(2,KC) $ YP(2)=XC(IPC,KC)*SCPR
         CALL PLTDTS(1,0,XP,YP,2,0)
C     THIS PLOTTED CONNECTION TO CORRECTED DATUM
         RE(1,1)=ERTP(1,KC)**2
         RE(2,2)=ERTP(2,KC)**2
         RE(1,2)=0. $ RE(2,1)=0.
         CALL ERELCM(TC,PC,RE,ERF,XP,YP)
C     THIS COMPUTED THE ERROR ELLIPSE
         CALL PLTDTS(1,0,XP,YP,201,0)
C     THIS PLOTTED THE ERROR ELLIPSE
      45 CONTINUE

         XP(1)=XMIN $ XP(2)=TARD(KH) $ XP(3)=XP(2)
         YP(1)=0.0 $ YP(2)=0.0 $ YP(3)=PIND(KH)
         CALL PLTDTS(1,0,XP,YP,3,0)
C     THIS PLOTTED PRESSURE AHEAD OF SHOCK AND INITIAL PRESSURE

C     NEXT COMPUTE FITTED CURVE
         DO 75 KA=1,201
         XP(KA)=TARD(KH)+(XMAX-TARD(KH))*FLOAT(KA-1)/200.
         XF(ITC,1)=XP(KA)/SCTI
         XF(IPC,1)=0.
         XF(3,1)=DIST(KH)/SCDI
C     PFIELD PARAMETERS ARE SET FOR SCALED CALCULATIONS,
C     THEREFORE INPUT MUST BE SCALED, TOO
         CALL PFIELD(XF,1,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
         IF(NBAD.EQ.0)GOTO 63 $ F=0.
         DO 61 KB=1,NP
      61 FP(KB)=0.

      63 YP(KA)=F*SCPR
C     YP IS OVERPRESSURE IN PASCALS
         EP(KA)=0
         DO 65 KPA=1,NP $ DO 65 KPB=1,NP
      65 EP(KA)=EP(KA)+FP(KPA)*VPAR(KPA,KPB)*FP(KPB)
         EP(KA)=SQRT(ABS(EP(KA)))*SCPR
      75 CONTINUE
```

```
      CALL PLTDTS(1,0,XP,YP,201,0)
C  THIS PLOTTED THE FITTED FIELD

      DO 85 KA=1,201
   85 YP(KA)=YP(KA)+EP(KA)*ERF
      CALL PLTDTS(1,0,XP,YP,201,0)
      DO 95 KA=1,201
   95 YP(KA)=YP(KA)-2.*EP(KA)*ERF
      CALL PLTDTS(1,0,XP,YP,201,0)
C  THIS PLOTTED CONFIDENCE LIMITS

      CALL PLTPGE
C  PLOTTING COMPLETED.  REPEAT FOR NEXT HISTORY
      KCS=KCS+KSET

  155 CONTINUE
C  END OF LOOP 15-155 OVER ALL HISTORIES

      RETURN
      END

:L EXCEEDS 131,071 WORDS (LCM=I REQUIRED)
```

```
      SUBROUTINE COLSACA(X,R,ALABEL,LSTX,NX,NSET,PAR,NP,FU,ITYPE,
     A XC,C,LSTN,NRGD,ERZ,V,ERP,LBAD,NXD,NPD,W,NW)
C  LEAST SQUARES ROUTINE FOR CORRELATED DATA AND SCALAR CONSTRAINTS
C
C IN THIS ROUTINE COMPUTE ADDRESSES IN THE WORK AREA W(NW) AND THEN
C  CALL COLSACB WITH CORRESPONDING ARGUMENTS
C  THE DIMENSION NW OF THE WORK AREA W MUST BE LARGER OR EQUAL TO
C                           NX*(1+NX)*2 + NX*NP*4 + NP*(1+NP)*8 +
C                         + NXD*(1+NXD) + NXD*NPD + NPD*(1+NPD) +
C                         + NR*(1+NX+NX*NX) + MAXO(NX*(3+NX),NP*(3+NP))
C  FOR DOUBLE PRECISION CALCULATIONS THE REQUIRED WORK AREA IS
C                           NX*(1+NX)*4 + NX*NP*8 + NP*(15+18*NP) +
C                         + NXD*(1+NXD) + NXD*NPD + NPD*(1+NPD) +
C                         + NR*(1+NX+2*NX*NX) + MAXO(NX*(3+NX),NP*(3+NP))*2
C
C  THE MEANINGS OF ALL OTHER ARGUMENTS ARE GIVEN IN COLSACB
      DIMENSION W(1)
      LEVEL 2,X,R,ALABEL,LSTX,XC,C,W,LSTN
      EXTERNAL MTRINDB,MTRINVB
      DATA(I=2)
C  I=1 FOR SINGLE PRECISION COMPUTING
C  I=2 FOR DOUBLE PRECISION COMPUTING
      KFP=NXD+1 $ KFXX=KFP+NPD  $  KFXP=KFXX+NXD*NXD
      KFPP=KFXP+NXD*NPD$ KRINV=KFPP+NPD*NPD
C  ASSUME THAT CONSTRAINT SUBROUTINE IS CODED FOR
C  MAXIMUM X-DIMENSION NXD AND PAR-DIMENSION NPD
C  THE FOLLOWING ARRAYS ARE USED ONLY WITHIN COLSACB, AND
C  THEREFORE ONLY ACTUAL DIMENSIONS NX AND  NP ARE NEEDED
      KRL=KRINV+NX*NX*NSET*I $ KA=KRL+NX*I $ KGG=KA+NX*NX*I
      KB=KGG+NX*NX*I $ KD=KB+NP*NX*I
      KE=KD+NP*NP*I $ KBG=KE+NX*NP*I $ KH=KBG+NP*NX*I
      KFF=KH+NP*NX*I $ KAM=KFF+NP*I $ KAN=KAM+NP*NP*I
      KRS=KAN+NP*NP*I $ KTAU=KRS+NP*I
      KEPS=KTAU+NP*I $ KCOR=KEPS+NX*I $ KGGFACT=KCOR+NP*NP*I
      KDUM=KGGFACT+NSET $ KANN=KDUM+NP*I $ KTTAU=KANN+NP*NP*I
      KPLAST=KTTAU+NP*I $ KCLAST=KPLAST+NP
      KANGAUS=KCLAST+NX*NSET $ KRSGAUS=KANGAUS+NP*NP*I
      KANIN=KRSGAUS+NP*I
      KANLAST=KANIN+NP*NP*I $ KRSLAST=KANLAST+NP*NP*I
      KVD=KRSLAST+NP*I $ KWMAT=KVD+NP*NP*I
      KEND=KWMAT+MAXO(NX*(3+NX),NP*(3+NP))*I-1
      IF(KEND.LE.NW)GOTO 25
      LBAD=NW
      PRINT 15,LBAD,KEND,NW
      RETURN
   15 FORMAT(1H0,10X,30HRETURN FROM COLSACA WITH LBAD=,I6,
     A34H BECAUSE STORAGE REQUIREMENT KEND=,I6,
     B24H EXCEEDS W-DIMENSION NW=,I6)
C
25    PRINT 27,KEND
27    FORMAT(1H1,10X,34HENTERING THE LEAST SQUARES ROUTINE,
     A 8H COLSACA,/,1H ,10X,25HTHE PRESENT RUN REQUIRES ,
     B32HA WORK ARRAY WITH THE DIMENSION ,I5,1H.,/)
      IF(I.EQ.2) GOTO 35
C
      CALL COLSACB(X,R,ALABEL,LSTX,NX,NSET,PAR,NP,FU,ITYPE,
     A XC,C,LSTN,NRGD,ERZ,V,ERP,LBAD,NXD,NPD,
```

```
     B W(1),W(KFP),W(KFXX),W(KFXP),W(KFPP),W(KRINV),
     C W(KRL),W(KA),W(KGG),W(KB),W(KD),W(KE),W(KBG),
     D W(KH),W(KFF),W(KAM),W(KAN),W(KRS),W(KTAU),
     E W(KEPS),W(KCOR),W(KGGFACT),W(KDUM),W(KANN),W(KTTAU),W(KPLAST),
     F W(KCLAST),W(KANGAUS),W(KRSGAUS),W(KANIN),
     G W(KANLAST),W(KRSLAST),W(KVD),W(KWMAT),MTRINVB)
      RETURN
   35 CONTINUE
      CALL COLSACB(X,R,ALABEL,LSTX,NX,NSET,PAR,NP,FU,ITYPE,
     A XC,C,LSTN,NRGD,ERZ,V,ERP,LBAD,NXD,NPD,
     B W(1),W(KFP),W(KFXX),W(KFXP),W(KFPP),W(KRINV),
     C W(KRL),W(KA),W(KGG),W(KB),W(KD),W(KE),W(KBG),
     D W(KH),W(KFF),W(KAM),W(KAN),W(KRS),W(KTAU),
     E W(KEPS),W(KCOR),W(KGGFACT),W(KDUM),W(KANN),W(KTTAU),W(KPLAST),
     F W(KCLAST),W(KANGAUS),W(KRSGAUS),W(KANIN),
     G W(KANLAST),W(KRSLAST),W(KVD),W(KWMAT),MTRINDB)
      RETURN
      END
```

```
      SUBROUTINE COLSACB(X,R,ALABEL,LSTX,NX,NR,PAR,NP,FU,IC,
     A XC,C,LSTN,NRGD,ERZ,V,ERP,LBAD,NXD,NPD,
     B FX,FP,FXX,FXP,FPP,
     C RINV,RL,A,GG,B,D,E,BG,H,FF,AM,AN,RS,TAU,
     D EPS,COR,GGFACT,DUM,ANN,TTAU,PLAST,CLAST,ANGAUS,RSGAUS,
     F ANIN,ANLAST,RSLAST,VD,WMAT,MTRINDB)
C  LEAST SQUARES ROUTINE FOR CORRELATED DATA AND SCALAR CONSTRAINTS
      DOUBLE PRECISION RL,G,AK,A,DGG,GG,RINV,DB,B,DE,E,D,BG,H,FF,
     A RSGAUS,RS,AN,ANGAUS,HRH,AM,WP,TTAU,TAU,ANLAST,RSLAST,WPLAST,
     B ANN,ANIN,DUM,VD,WC,COR,W,DET,WLAST,BGE
C
C  X(NXD,NR)          = NR SETS WITH NX.LE.NXD OBSERVA4IONS EACH
C  R(NXD,NXD,NR)      = VARIANCE-COVARIANCE MATRICES OF OBSERVATIONS X
C  ALABEL(2,NR)       = ALPHANUMERIC LABELS OF OBSERVATION SETS
C  LSTX(NR)           = ONLY SETS WITH ZERO LSTX WILL BE USED
C  NX                 = NUMBER OF OBSERVATIONS IN EACH SET.  (NX.LE.NXD)
C  NR                 = NUMBER OF X-SETS, INCLUDING SETS WITH LSTX.NE.0
C  PAR(NPD)           = PARAMETERS. WILL BE REPLACED BY L.SQ. SOLUTION
C  NP                 = NUMBER OF PARAMETERS.   (0.LE.NP.LE.NPD)
C  FU                 = NAME OF CONSTRAINT SUBROUTINE
C  IC                = ITERATION TYPE IN BINARY CODE
C                     = 0 - NORMAL. SET C=0 AT START, BEGIN WITH PARAMETER
C                             ITERATION, USE NEWTON-RAPHSON FORMULAS.
C                     = 1 - DO NOT SET C=0 AT START
C                     = 2 - START ITERATION WITH RESIDUAL UPDATING
C                     = 4 - START ITERATION USING GAUSS-NEWTON FORMULAS
C  XC(NXD,NR)         = CORRECTED (ADJUSTED) OBSERVATIONS  = X+C
C  C(NXD,NR)          = RESIDUALS (CORRECTIONS OF X)
C  LSTN(NR)           = LSTN.NE.0 IF THE SET WAS NOT USED FOR ADJUSTMENT
C  V(NPD,NPD)         = VARIANCE-COVARIANCE MATRIX OF THE PARAMETERS
C  ERP(NPD)           = STANDARD ERRORS OF THE PARAMETERS
C  NXD                = FIRST DIMENSION OF X, XC AND C, AND FIRST TWO
C                       DIMENSIONS OF R DECLARED BY DIMENSION STATEMENT
C  NPD                = DIMENSIONS OF PAR, V AND ERP AS DECLARED BY
C                       DIMENSION STATEMENTS
C  LBAD               = LBAD.NE.0 IF ADJUSTMENT CANNOT BE DONE PROPERLY
C
C  THE REMAINING ARGUMENTS FX THROUGH WMAT ARE STORAGE ALLOCATIONS
C  SPECIFIED BY THE SUBROUTINE COLSACA. A FORMULA FOR THE
C  REQUIRED STORAGE AREA FOR THESE ALLOCATIONS IS GIVEN IN COLSACA
C
C  MTRINDB            = NAME OF SUBROUTINE FOR MATRIX INVERSION,
C                       ALSO SPECIFIED BY THE SUBROUTINE COLSACA
C
      DIMENSION X(NXD,1),R(NXD,NXD,1),A 4BEL(2,1),LSTX(1),
     APAR(NPD),XC(NXD,1),C(NXD,1),V(NPD,NPD),ERP(NPD),LSTN(1),
     B FX(NXD),FP(NPD),FXX(NXD,NXD),FXP(NXD,NPD),FPP(NPD,NPD),
     C RINV(NX,NX,1),RL(NX),A(NX,NX),GG(NX,NX),B(NP,NX),D(NP,NP),
     D E(NX,NP),BG(NP,NX),H(NP,NX),FF(NP),AM(NP,NP),AN(NP,NP),RS(NP),
     E TAU(NP),EPS(NX),COR(NP,NP),GGFACT(1),DUM(NX),ANN(NP,NP),
     F TTAU(NP),PLAST(NP),CLAST(NX,1),ANGAUS(NP,NP),RSGAUS(NP),
     G ANIN(NP,NP),ANLAST(NP,NP),RSLAST(NP),VD(NP,NP)
      LEVEL 2,FX,FP,FXX,FXP,FPP,RINV,RL,A,GG,B,D,E,BG,H,FF,AM,AN,RS,TAU,
     1 EPS,COR,GGFACT,DUM,ANN,TTAU,PLAST,CLAST,ANGAUS,RSGAUS,ANIN,ANLAST
     2,RSLAST,VD,WMAT
      LEVEL 2,X,R,ALABEL,LSTX,XC,C,LSTN
      NXMX=NXD$ NPMX=NPD
```

```
C  MAXIMUM DIMENSIONS AS DECLARED BY THE CALLING PROGRAM
      DATA(SUBNAM=9H COLSACB )
C  NAME OF THE SUBROUTINE FOR ERROR MESSAGES AND OUTPUT
      DATA(ITMAX=25),(ERMAX=2.)
C  ITMAX IS THE MAXIMUM NUMBER OF ITERATIONS
C  ERMAX IS FACTOR IN LOOP 1056 TO CHECK FOR LARGE RESIDUALS
      PRINT 11,SUBNAM
11    FORMAT(1H0,10X,37HENTERING THE LEAST SQUARES SUBROUTINE,
     A A9,42HFOR CORRELATED DATA AND SCALAR CONSTRAINTS,/
     A 1H ,10X,19HROUTINE USES DOUBLE,
     B43H PRECISION ARITHMETIC FOR MOST CALCULATIONS,//)
      IF(NX.GE.1.AND.NX.LE.NXMX) GOTO 45
      LBAD=1 $ PRINT 15,SUBNAM
   15 FORMAT(15H0   RETURN FROM,A9,30H15 BECAUSE NX IS OUTSIDE RANGE)
   25  FORMAT(3X,3HNX=,I8,30H IS THE NUMBER OF OBSERVATIONS
     1 9H IN A SET,/,3X,3HNR=,I8,22H IS THE NUMBER OF SETS,/,
     2 3X,3HNP=,I8,28H IS THE NUMBER OF PARAMETERS)
   30 PRINT 25,NX,NR,NP
      PRINT 35,LBAD
      RETURN
   35 FORMAT(3X,5HLBAD=,I6)
   45 IF(NR.GE.1)GOTO 65
      LBAD=2 $ PRINT 55,SUBNAM $ GOTO 30
   55 FORMAT(15H0   RETURN FROM,A9,30H45 BECAUSE NR IS OUTSIDE RANGE)
65    IF(NP.GE.0.AND.NP.LE.NPMX.AND.NP.LE.NR) GOTO 85
      LBAD=3 $ PRINT 75,SUBNAM $ GOTO 30
   75 FORMAT(15H0   RETURN FROM,A9,30H65 BECAUSE NP IS OUTSIDE RANGE)
   85 LBAD=0 $ NRGD=0
      IF(IC.LT.0.OR.IC.GT.7)IC=0
C     IC IS MEANINGFULL ONLY BETWEEN ZERO AND 7
      GAUS=0. $ IF(IC.GE.4)GAUS=1. $ MODI=0
C  GAUS=1.  INDICATES THAT GAUSSIAN ITERATION WILL BE USED
C
      DO 135 KA=1,NR
      LSTN(KA)=1
      IF(LSTX(KA).NE.0)GOTO 135
      DO 95 KB=1,NX $ DO 95 KC=1,NX
   95 A(KB,KC)=R(KB,KC,KA)
C
      CALL MTRINDB(A,NX,DUM,NX,0,DET,WMAT)
C  INVERT MATRIX
C
      IF(DET.GT.0.)GOTO 105
C  ONLY DATA WITH POSITIVE DEFINITE R WILL BE ACCEPTED
      PRINT 100,KA,ALABEL(1,KA),ALABEL(2,KA)
      GOTO 135
  100 FORMAT(3X,47HVARIANCE MATRIX R NOT POSITIVE DEFINITE FOR SET,
     A I5,21H WITH LABELS ALABEL= ,2A10)
  105 DO 115 KB=1,NX $ DO 115 KC=1,NX
  115 RINV(KB,KC,KA)=A(KB,KC)
C  RINV IS THE INVERSE TO R AND IS NEEDED TO COMPUTE W
      LSTN(KA)=0 $ NRGD=NRGD+1
      DO 125 KB=1,NX
      IF((IC/2)*2.EQ.IC) C(KB,KA)=0.
  125 XC(KB,KA)=X(KB,KA)+C(KB,KA)
  135 CONTINUE
C
```

```
          IF(NRGD.LE.0) GOTO 145
          IF(NP-NRGD)185,165,145
  145 LBAD=145
          PRINT 150,SUBNAM $ PRINT 155,NRGD $ GOTO 30
  150 FORMAT(15H0   RETURN FROM,A9,22H145 BECAUSE NP.GT.NRGD)
  155 FORMAT(3X,5HNRGD=,I6,27H IS THE NUMBER OF GOOD SETS)
  165 PRINT 175,SUBNAM $ PRINT 155,NRGD $ PRINT 25,NX,NR,NP
  175 FORMAT(14H0   WARNING AT,A9,19H175 BECAUSE NP=NRGD)
  185 ITERNR=0 $ IWTEST=0
C  COUNTER OF ITERATIONS AND CONVERGENCE INDICATOR FOR W
          KPCT=0 $ IPTEST=0
C  COUNTER OF PARAMETER SUBITERATIONS AND CONVERGENCE INDICATOR
          KCCT=0 $ ICTEST=0
C  COUNTER OF RESIDUAL SUBITERATIONS AND CONVERGENCE INDICATOR
          ERZ=1. $ W= FLOAT(NRGD-NP) $ WP=W
          PRINT 190,SUBNAM,IC
  190 FORMAT(1H ,10X,20HITERATION RESULTS BY,A9,10X,16H(ITERATION TYPE ,
     A3HIC=,I3,1H),///,1H ,2X,9HITERATION,8X,1HW,35X,10HPARAMETERS,//)
C
C  ITERATION STARTS AT 195
  195 WLAST=W $ WPLAST=WP  $ KPCT=0
          IF(NP.GT.0)GOTO 196
          PRINT 198,ITERNR,W$ GOTO 569
196   DO 197 KA=1,NP
  197 PLAST(KA)=PAR(KA)
          KP=MIN0(NP,5) $ PRINT 198,ITERNR,W,(PAR(J),J=1,KP)
          IF(KP.EQ.NP)GOTO 200
          KPP=KP+1 $ PRINT 199,(PAR(J),J=KPP,NP)
  198 FORMAT(4X,I5,1PE19.12,5X,5(2X,1PE16.9))
  199 FORMAT(33X,5(2X,1PE16.9))
  200 IF(ITERNR.GT.0) GOTO 204
          IF(IC-4.GE.2) GOTO 575  $  IF(IC.EQ.2.OR.IC.EQ.3) GOTO 575
C  START WITH RESIDUAL ITERATION AT 575 IF IC=2
  204 MARQ=0
C  MARQ INDICATES NUMBER OF MARQUARDT CORRECTIONS.  SEE 435.
  205           NRGDP=0 $ WP=0
208   DO 217 KA=1,NP$ RS(KA)=0.$ RSGAUS(KA)=0.
          DO 217 KB=1,NP
          AM(KA,KB)=0$ AN(KA,KB)=0.$ ANGAUS(KA,KB)=0.
  217 CONTINUE
C
  225 DO 405 KA=1,NR
C  THIS LOOP ESTABLISHES EQUATIONS FOR PARAMETER CORRECTIONS
          IF(LSTN(KA).EQ.1)GOTO 405
C
          CALL FU(XC,KA,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS IS THE CONSTRAINT SUBROUTINE.  ITS ARGUMENTS ARE
C  XC(NXD,NR)       = OBSERVATIONS
C  KA               = NUMBER OF SET WHICH WILL BE USED FOR CALCULATIONS
C  PAR(NPD)         = PARAMETER VECTOR
C
C  THE FOLLOWING WILL BE CALCULATED BY FU
C  F                       = CONSTRAINT FUNCTIONAL
C  FX(NXD) AND FP(NXP) = FIRST ORDER DERIVATIVES OF F
C  FXX(NXD,NXD), FXP(NXD,NPD), FPP(NPD,NPD) = SECOND ORDER DERIVATIVES
C  NBAD        = NBAD.NE.0 IF F CANNOT BE COMPUTED FOR GIVEN XC AND PAR
C
```

```
  235 IF(NBAD.EQ.0)GOTO 245
      LSTN(KA)=235000+IABS(NBAD) $ GOTO 405
  245 DO 255 KB=1,NX
      RL(KB)=0 $ DO 255 KC=1,NX
  255 RL(KB)=RL(KB)+R(KB,KC,KA)*FX(KC)
      G=0 $ DO 265 KB=1,NX
  265 G=G+FX(KB)*RL(KB)
  275 IF(G.GT.1.E-100)GOTO 285
      LSTN(KA)=275
      PRINT 277,KPCT  $  PRINT 278,KA,ALABEL(1,KA),ALABEL(2,KA)
      GOTO 405
  277 FORMAT(3X,29HWEIGHT G NOT POSITIVE AT 275.,9H    KPCT=,I4)
  278 FORMAT(5X,3HKA=,I5,3X,7HALABEL=,2A10)
  285 G=1./G
      AK=-F
      DO 305 KB=1,NX $ DO 295 KC=1,NX
      A(KB,KC)=FX(KB)*RL(KC)*G
      IF(KB.EQ.KC)A(KB,KC)=A(KB,KC)-1.
  295 CONTINUE
  305 AK=AK+FX(KB)*C(KB,KA)
      AK=AK*G
      GGFACT(KA)=1.
  311 DO 325 KB=1,NX $ DO 325 KC=1,NX
      DGG=0
      DO 315 KD=1,NX $ DO 315 KE=1,NX
  315 DGG=DGG+GGFACT(KA)*AK*R(KB,KD,KA)*A(KD,KE)*FXX(KE,KC)
      IF(KB.EQ.KC)DGG=DGG+1.
  325 GG(KB,KC)=DGG
      CALL MTRINDB(GG,NX,DUM,NX,0,DET,WMAT)
      IF(DET.GT.1.E-100)GOTO 335
      GGFACT(KA)=GGFACT(KA)*0.5 $ IF(GGFACT(KA).LT.1.E-3)GGFACT(KA)=0.
C  FXX IN FORMULA FOR GG IS REDUCED FOR NUMERICAL STABILITY
      GOTO 311
  335 DO 345 KB=1,NX
      DB=0 $ DO 337 KC=1,NX
  337 DB=DB+RL(KC)*FXX(KC,KB)
      DO 345 KC=1,NP
      B(KC,KB)=AK*(FXP(KB,KC)-G*FP(KC)*DB)
      DE=0 $ DO 339 KD=1,NX $ DO 339 KE=1,NX
  339 DE=DE+R(KB,KD,KA)*A(KD,KE)*FXP(KE,KC)
      E(KB,KC)=G*RL(KB)*FP(KC)+AK*DE
  345 CONTINUE
      DO 355 KB=1,NP
      DB=0 $ DO 347 KD=1,NX
  347 DB=DB+RL(KD)*FXP(KD,KB)
      DO 355 KC=1,NP
355   D(KC,KB)=G*FP(KB)*FP(KC)-AK*(FPP(KB,KC)-G*FP(KC)*DB)
      DO 365 KB=1,NP $ DO 365 KC=1,NX
      BG(KB,KC)=0 $ DO 357 KD=1,NX
  357 BG(KB,KC)=BG(KB,KC)+B(KB,KD)*GG(KD,KC)
  365 CONTINUE
      DO 385 KB=1,NP
      DO 375 KC=1,NX
      DE=0 $ DO 367 KD=1,NX
  367 DE=DE+BG(KB,KD)*A(KC,KD)
  375 H(KB,KC)=G*FP(KB)*FX(KC)+DE
      DE=0.$ DO 377 KD=1,NX
```

```
  377 DE=DE+BG(KB,KD)*(AK*RL(KD)-C(KD,KA))
  384 FF(KB)=AK*FP(KB)+DE
  385 CONTINUE
C  THIS COMPLETES CALCULATIONS FOR SET KA. NOW ADD UP MATRICES
      DO 395 KB=1,NP
      RSGAUS(KB)=RSGAUS(KB)+AK*FP(KB)
      RS(KB)=RS(KB)+FF(KB)
C  THESE ARE RIGHT HAND SIDES FOR TAU EQS.
      DO 395 KC=1,NP
      BGE=0.$ DO 389 KD=1,NX
  389 BGE=BGE+BG(KB,KD)*E(KD,KC)
  390 AN(KB,KC)=AN(KB,KC)+D(KB,KC)+BGE
C  THIS IS MATRIX OF EQS. FOR TAU
      ANGAUS(KB,KC)=ANGAUS(KB,KC)+G*FP(KB)*FP(KC)
      HRH=0 $ DO 391 KD=1,NX $ DO 391 KE=1,NX
  391 HRH=HRH+H(KB,KD)*R(KD,KE,KA)*H(KC,KE)
      AM(KB,KC)=AM(KB,KC)+HRH
C  THIS IS THE INFLUENCE MATRIX OF SET KA
  395 CONTINUE
      WP=WP+AK**2/G
      NRGDP=NRGDP+1
C  COUNT GOOD SETS IN COMPUTATION LOOP FOR PARAMETERS
  405 CONTINUE
C  END OF LOOP 225-405 OVER ALL SETS OF OBSERVATIONS
C
  415 IF(NP.LE.NRGDP.AND.NRGDP.GT.0) GOTO 425
      LBAD=415 $ PRINT 417,SUBNAM
      PRINT 419,NRGDP $ PRINT 25,NX,NR,NP $ PRINT 35,LBAD $ GOTO 1057
  417 FORMAT(15H0   RETURN FROM,A9,23H415 BECAUSE NP.GT.NRGDP)
  419 FORMAT(3X,6HNRGDP=,I5,26H IS THE NUMBER OF SETS FOR,
     A52H WHICH CALCULATIONS CAN BE PERFORMED IN LOOP 225-405)
  425 IF(KPCT.EQ.0)GOTO 485
C  AFTER FIRST PARAMETER ITERATION CHECK IF WP DECREASES
      IF(WP.LT.WPLAST*1.10)GOTO 475
      IF(MARQ.GT.10) GOTO 475
C  APPLY MARQUARDT IF WP HAS INCREASED TOO MUCH
  435 MARQ=MARQ+1  $ ALAM=10.**(MARQ-4)
      DO 445 KA=1,NP     $  TTAU(KA)=RSLAST(KA)
      DO 445 KB=1,NP    $  AN(KA,KB)=ANLAST(KA,KB)
      IF(KA.EQ.KB)AN(KA,KB)=AN(KA,KB)*(ALAM+1.)
  445 CONTINUE
C
      CALL MTRINDB(AN,NP,TTAU,NP,1,DET,WMAT)
C  INVERT MATRIX AND SOLVE LINEAR EQUATIONS
C
      IF(DET.NE.0.)GOTO 455
      GOTO 435
  455 DO 465 KA=1,NP
      PAR(KA)=PAR(KA)-TAU(KA)+TTAU(KA)
  465 TAU(KA)=TTAU(KA)
      GOTO 205
C  NOW REPEAT AT 205 LAST ITERATION WITH DIFFERENT PAR
C
  475 IF(MARQ.EQ.0)GOTO 485
      PRINT 477,MARQ,KPCT,WP
  477 FORMAT(2X,29HMARQUARDT CORRECTION APPLIED ,I4,
     A15H TIMES AT KPCT=,I4,5X,3HWP=,1PE19.12)
```

```
485   WPLAST=WP$ INDTAU=0
      IF(GAUS.NE.0.)GOTO 491
487   DO 489 KA=1,NP$ TAU(KA)=RS(KA)$ RSLAST(KA)=RS(KA)
      DO 489 KB=1,NP  $  ANLAST(KA,KB)=AN(KA,KB)
489   ANN(KA,KB)=AN(KA,KB)
      GOTO 495
491   DO 493 KA=1,NP$ TAU(KA)=RSGAUS(KA)$ RSLAST(KA)=RSGAUS(KA)
      DO 493 KB=1,NP  $  ANLAST(KA,KB)=ANGAUS(KA,KB)
493   ANN(KA,KB)=ANGAUS(KA,KB)
  495 CALL MTRINDB(ANN,NP,TAU,NP,1,DET,WMAT)
      IF(DET.NE.0.)GOTO 511
      IF(INDTAU.EQ.0)GOTO 509
      LBAD=495 $ PRINT 497,SUBNAM,LBAD
  497 FORMAT(15H0   RETURN FROM,A9,14H495 WITH LBAD=,I4,
     A52H BECAUSE MATRIX ANN OF EQUATIONS FOR TAU IS SINGULAR)
      PRINT 498
498   FORMAT(31H0  THE SINGULAR GAUSS MATRIX IS,/)
      DO 499 KA=1,NP
      PRINT 500,(ANGAUS(KA,J),J=1,NP)
499   CONTINUE
500   FORMAT(1H ,10(1X,1PE12.5))
      PRINT 501
501   FORMAT(32H0  THE SINGULAR NEWTON MATRIX IS,/)
      DO 502 KA=1,NP
      PRINT 500,(AN(KA,J),J=1,NP)
502   CONTINUE
      RETURN
509   INDTAU=1$ IF(GAUS.NE.0.)GOTO 487
      GOTO 491
511   INDVAR=0
      IF(INDTAU.EQ.0.AND.GAUS.EQ.0.)GOTO 515
      IF(INDTAU.NE.0.AND.GAUS.NE.0.)GOTO 515
C  BRANCH TO 515 IF ANN CONTAINS THE INVERSE OF NEWTON MATRIX AN
      IF(GAUS.EQ.0..AND.INDTAU.NE.0) GOTO 514
C  BRANCH TO 514 IF NEWTON MATRIX AN WAS SINGULAR
      DO 512 KA=1,NP  $  DO 512 KB=1,NP
512   ANIN(KA,KB)=AN(KA,KB)
      CALL MTRINDB(ANIN,NP,DUM,NP,0,DET,WMAT)
      IF(DET.EQ.0.) GOTO 514
      DO 513 KA=1,NP  $  DO 513 KB=1,NP
513   ANN(KA,KB)=ANIN(KA,KB)
      GOTO 515
514   INDVAR=1
C  INDVAR=1 INDICATES THAT GAUSS MATRIX USED FOR VARIANCES
  515 DO 525 KA=1,NP
      PAR(KA)=PAR(KA)+TAU(KA)
      DO 525 KB=1,NP
      VD(KA,KB)=0$ DO 517 KC=1,NP $ DO 517 KD=1,NP
517   VD(KA,KB)=VD(KA,KB)+ANN(KA,KC)*AM(KC,KD)*ANN(KB,KD)
  525 CONTINUE
      KPCT=KPCT+1
      IF(MARQ.NE.0)GOTO 555
C  APPLY CONVERGENCE TESTS ONLY IF MARQUART WAS NOT USED
C
      DE=0. $ DO 535 KA=1,NP $ DO 535 KB=1,NP
535   DE=DE+TAU(KA)*AN(KA,KB)*TAU(KB)
      FTEST=10.**(-MIN0(10,ITERNR+2))*(1.+99.*GAUS)
```

```
      SDE=DE  $  IF(ABS(SDE).GT.WP*FTEST) GOTO 555
      FTEST=AMAX1(ERZ,0.01)*10.**(-MINO(8,ITERNR+2))*(1.+99.*GAUS)
      IPITER=0
      DO 545 KA=1,NP
      STAU=TAU(KA)  $   SVD=VD(KA,KA)
      IF(ABS(STAU).LT.SQRT(SVD)*FTEST) IPITER=IPITER+1
  545 CONTINUE
      IF(IPITER.EQ.NP)GOTO 565
  555 IF(KPCT.LE.11)GOTO 204
  565 PRINT 567,KPCT
567   FORMAT(1H ,10X,5HKPCT=,I4,24H =  PARAMETER ITERATIONS)
      PTEST=AMAX1(ERZ,0.01)*1.E-8*(1.+99.*GAUS)
      DO 568 KA=1,NP
      SVD=VD(KA,KA)
      IF(ABS(PAR(KA)-PLAST(KA)).GT.SQRT(SVD)*PTEST) IPTEST=0
  568 CONTINUE
569   IPTEST=IPTEST+1
C  IPTEST COUNTS CONSECUTIVE PASSES OF TESTS FOR PAR
C  ENTER 569 FROM 195 IN PROBLEMS WITHOUT PARAMETERS
C
  570 IF(IPTEST.GT.2.AND.IWTEST.GT.2.AND.ICTEST.GT.2)GOTO 785
C  THIS IS TEST AND BRANCH FOR REGULAR RETURN
575   IF(ITERNR.GT.ITMAX+MODI)GOTO 775
      KCCT=0  $   IEPTE=1
C  COUNTER OF RESIDUAL ITERATIONS AND RESIDUAL CONVERGENCE INDICATOR
      DO 577 KA=1,NR $ DO 577 KB=1,NX
  577 CLAST(KB,KA)=C(KB,KA)
      EPTEST=AMAX1(ERZ,0.01)*10.**(-MINO(8,ITERNR+2))*(1.+99.*GAUS)
C
C  RESIDUAL ITERATION STARTS AT 578
  578 W=0 $ NRGDC=0
      DO 745 KA=1,NR
      IF(LSTN(KA).EQ.1)GOTO 745
      LSTN(KA)=0
      CALL FU(XC,KA,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
  585 IF(NBAD.EQ.0)GOTO 595
      LSTN(KA)=585000+IABS(NBAD) $ GOTO 745
  595 DO 605 KB=1,NX
      RL(KB)=0 $ DO 605 KC=1,NX
  605 RL(KB)=RL(KB)+R(KB,KC,KA)*FX(KC)
      G=0 $ DO 615 KB=1,NX
  615 G=G+FX(KB)*RL(KB)
  625 IF(G.GT.1.E-100)GOTO 635
      LSTN(KA)=625
      PRINT 627,KCCT  $  PRINT 278,KA,ALABEL(1,KA),ALABEL(2,KA)
      GOTO 745
  627 FORMAT(3X,29HWEIGHT G NOT POSITIVE AT 625.,9H    KPCT=,I4)
  635 G=1./G
      AK=-F
      DO 655 KB=1,NX $ DO 645 KC=1,NX
      A(KB,KC)=FX(KB)*RL(KC)*G
      IF(KB.EQ.KC)A(KB,KC)=A(KB,KC)-1.
  645 CONTINUE
  655 AK=AK+FX(KB)*C(KB,KA)
      AK=AK*G
      GGFACT(KA)=1.
  665 DO 685 KB=1,NX $ DO 685 KC=1,NX
```

```
      DGG=0.  $  IF(GAUS.NE.0.) GOTO 681
               DO 675 KD=1,NX $ DO 675 KE=1,NX
  675 DGG=DGG+GGFACT(KA)*AK*R(KB,KD,KA)*A(KD,KE)*FXX(KE,KC)
  681 IF(KB.EQ.KC)DGG=DGG+1.
  685 GG(KB,KC)=DGG
      CALL MTRINDB(GG,NX,DUM,NX,0,DET,WMAT)
      IF(DET.GT.1.E-100)GOTO 695
      GGFACT(KA)=GGFACT(KA)*0.5 $ IF(GGFACT(KA).LT.1.E-3)GGFACT(KA)=0.
C  REDUCE INFLUENCE OF FXX IN GG TO IMPROVE STABILITY
      GOTO 665
  695 DO 715 KB=1,NX
      EPS(KB)=0
      DO 715 KC=1,NX
  715 EPS(KB)=EPS(KB)+GG(KB,KC)*(AK*RL(KC)-C(KC,KA))
      DO 725 KB=1,NX
      IF( ABS(EPS(KB)).GT.EPTEST* SQRT(R(KB,KB,KA)))IEPTE=0
      C(KB,KA)=C(KB,KA)+EPS(KB)
  725 XC(KB,KA)=X(KB,KA)+C(KB,KA)
      WC=0 $ DO 735 KB=1,NX $ DO 735 KC=1,NX
  735 WC=C(KB,KA)*RINV(KB,KC,KA)*C(KC,KA)+WC
      W=W+WC
      NRGDC=NRGDC+1
  745 CONTINUE
C  END OF LOOP 575-745 FOR UPDATING OF RESIDUALS
C
      IF(NP.GT.NRGDC.OR.NRGDC.LE.0) GOTO 765
      KCCT=KCCT+1
      IF(KCCT.GT.11)GOTO 746
      IEPTE=IEPTE+1 $ IF(IEPTE.LE.1)GOTO 578
  746 PRINT 747,KCCT
747   FORMAT(1H ,10X,5HKCCT=,I4,23H =  RESIDUAL ITERATIONS)
      SW=W $ WTEST=AMAX1(SW,FLOAT(NRGD-NP)*0.01)*1.0E-10*(1.+99.*GAUS)
C  THIS TAKES CARE OF EXACT DATA FOR WHICH W=0.
      SWWL=W-WLAST  $  IF( ABS(SWWL).GT.WTEST) IWTEST=0
      EPF  =AMAX1(ERZ,0.01)*1.E-8*(1.+99.*GAUS)
      DO 755 KA=1,NR $ IF(LSTN(KA).NE.0)GOTO 755
      DO 754 KB=1,NX
      IF( ABS(C(KB,KA)-CLAST(KB,KA)).GT.EPF* SQRT(R(KB,KB,KA)))ICTEST=0
754   CONTINUE
  755 CONTINUE
      IWTEST=IWTEST+1 $ ICTEST=ICTEST+1
      ITERNR=ITERNR+1
      ERZSQ=1.
      IF(NP.GT.NRGDC)ERZSQ=W/ FLOAT(NRGDC-NP)
      ERZ=SQRT(ERZSQ)
      GOTO 195
C  BRANCH TO 195 FOR NEXT ITERATION
C
  765 LBAD=745 $ PRINT 767,SUBNAM $ PRINT 747,KCCT $ PRINT 768,NRGDC
      PRINT 25,NX,NR,NP $ PRINT 35,LBAD $ GOTO 1057
  767 FORMAT(15H0   RETURN FROM,A9,23H745 BECAUSE NP.GT.NRGDC)
  768 FORMAT(3X,6HNRGDC=,I5,26H IS THE NUMBER OF SETS FOR,
     A52H WHICH CALCULATIONS CAN BE PERFORMED IN LOOP 575-745)
  775 LBAD=ITMAX
C  ENTER 775 FROM 575 IF TOO MANY ITERATIONS
  776 PRINT 777,SUBNAM
  777 FORMAT(1H1,10X,34HRESULTS OF ADJUSTMENT BY THE LEAST,
```

```
      A19H SQUARES SUBROUTINE,A9,//)
       PRINT 779,ITMAX,LBAD
  779 FORMAT(43H0   WARNING.   THIS IS NOT A REGULAR RETURN,/,4H    ,
      A39(1H-),/,49H    COMPUTATION INTERRUPTED BECAUSE THE NUMBER OF,
      B27H ITERATIONS EXCEEDED ITMAX=,I5, 9H    LBAD=,I5,  1H.,/,
      C45H    TO CONTINUE ITERATION RESTART WITH ODD IC,//)
       GOTO 795
C  ENTER 785 FROM 570 FOR A REGULAR RETURN
785    IF(GAUS.EQ.0.)GOTO 788
       PRINT 786 $ MODI=3
       IPTEST=0 $ IWTEST=0 $ ICTEST=0 $ GAUS=0. $ GOTO 195
786    FORMAT(1H0,5X,35HSWITCH ITERATIONS TO NEWTON-RAPHSON,/)
C  BRANCH TO 195 FOR ADDITIONAL NEWTON ITERATIONS AFTER GAUSS ITERATIONS
  788 PRINT 777,SUBNAM
  795 IF(NRGDC.EQ.NRGD)GOTO 815
       PRINT 805
805    FORMAT(41H0  WARNING.   SOME OBSERVATION SETS COULD,
      A30H NOT BE USED FOR COMPUTATIONS.,//)
  815 IF(NP.LT.NRGDC)GOTO 835
       PRINT 825
825    FORMAT(41H0  WARNING.   THE NUMBER OF PARAMETERS IS,
      A47H EQUAL TO THE NUMBER OF USABLE OBSERVATON SETS.,//)
  835 PRINT 845,NP,NRGD,NRGDP,NX,ITERNR
845    FORMAT(10X,20HNUMBER OF PARAMETERS,10X,I5,/,
      A10X,26HNUMBER OF OBSERVATION SETS,4X,I5,/,
      B10X,19HNUMBER OF SETS USED,11X,I5,/,
      C10X,21HDIMENSION OF EACH SET,9X,I5,//
      D10X,20HNUMBER OF ITERATIONS,10X,I5,//)
       PRINT 855,W
855    FORMAT(10X,34HWEIGHTED SUM OF CORRECTION SQUARES,8X,
      A 7HW      =,1PE16.9,//)
       IF(NP.LT.NRGDC)GOTO 885
       ERZ=0.  $  VARZ=0.
       PRINT 875
875    FORMAT(10X,40HVARIANCE OF WEIGHT ONE AND CORRESPONDING,/,
      A10X,41HSTANDARD ERROR NOT COMPUTABLE BECAUSE THE,
      B10X,47HNUMBER OF PARAMETERS EQUALS THE NUMBER OF SETS.,//)
       GOTO 894
  885 VARZ=W/ FLOAT(NRGDC-NP)
       ERZ=0
       IF(VARZ.GT.0.)ERZ= SQRT(VARZ)
  894 PRINT 895,VARZ,ERZ
895    FORMAT(10X,22HVARIANCE OF WEIGHT ONE,20X,7HERZ**2=,1PE16.9,/
      A10X,39HSTANDARD ERROR OF A SET WITH WEIGHT ONE,3X,7HERZ    =,
      B1PE16.9,//)
       IF(NP.EQ.0)GOTO 1028
C
905    PRINT 915
915    FORMAT(1H ,13X,10HPARAMETERS,8X,16HLAST CORRECTIONS,6X,
      A15HSTANDARD ERRORS,6X,15HSTANDARD ERRORS,/,1H ,77X,
      B9HTIMES ERZ,//)
       DO 910 KA=1,NP
       SVD=VD(KA,KA)  $  ERP(KA)=SQRT(SVD)
       ERPZ=ERP(KA)*ERZ $ DIFP=PLAST(KA)-PAR(KA)
       PRINT 925,PAR(KA),DIFP,ERP(KA),ERPZ
910    CONTINUE
925    FORMAT(1H ,5X,4(5X,1PE16.9))
```

```
      IF(INDVAR.NE.0) PRINT 928
  928 FORMAT(42H0  WARNING.  SECOND ORDER DERIVATIVES WERE,
     A43H NOT USED FOR VARIANCE CALCULATIONS BECAUSE,/,
     B1H ,12X,29HTHE NEWTON MATRIX IS SINGULAR)
      PRINT 935
  935 FORMAT(1H ,//,1H ,10X,24HTHE FACTOR ERZ**2 IS NOT,
     A34H INCLUDED IN THE VARIANCE MATRIX V)
  965 DO 975 KA=1,NP$ DO 975 KB=1,NP
      V(KA,KB)=VD(KA,KB) $ SVD=VD(KA,KA)*VD(KB,KB)
  975 COR(KA,KB)=V(KA,KB)/ SQRT(SVD)
  995 PRINT 1005
 1005 FORMAT(1H ,///,10X,25HCORRELATION MATRIX OF THE,
     A11H PARAMETERS,//)
      DO 1015 KA=1,NP
      PRINT 1025,(COR(KA,J),J=1,NP)
 1015 CONTINUE
 1025 FORMAT(1X,10(2X,F11.8))
C
 1028 KPR=0
      DO 1045 KA=1,NR
      IF(LSTN(KA).NE.0)GOTO 1045
      IF(GGFACT(KA).EQ.1.)GOTO 1045
      IF(KPR.EQ.0)PRINT 1035
 1035 FORMAT(1H ,//,3X,33HFOR THE FOLLOWING SETS THE SECOND,
     A55H DERIVATIVES FXX HAVE BEEN REDUCED BY THE SHOWN FACTORS,
     B//,5X,10HSET NUMBER,5X,6HFACTOR,9X,10HSET LABELS,/)
      KPR=1
      PRINT 1037,KA,GGFACT(KA),ALABEL(1,KA),ALABEL(2,KA)
 1037 FORMAT(8X,I4,6X,1PE12.5,5X,2A10)
 1045 CONTINUE
      IF(ERZ.EQ.0.) GOTO 1057
C
      SQ=ERMAX*ERZ $ DUMSS=SQ**2
      KPR=0 $ DO 1056 KA=1,NR
      IF(LSTN(KA).NE.0) GOTO 1056
      DUMS=0. $ DO 1050 KB=1,NX $ DO 1050 KC=1,NX
 1050 DUMS=DUMS+C(KB,KA)*RINV(KB,KC,KA)*C(KC,KA)
      IF(DUMS.LT.DUMSS) GOTO 1056
      IF(KPR.EQ.0)PRINT 1052,ERMAX,SQ  $  KPR=1
 1052 FORMAT(1H ,//,1H ,3X,35HTHE FOLLOWING SETS HAVE CORRECTIONS,
     A24H LARGER THAN ERMAX*ERZ =,F4.1,8H * ERZ =,1PE12.5,//,1H ,4X,
     B7HSET NR.,10X,6HLABELS,11X,14HSQRT(C*RINV*C),/)
      DUMS=SQRT(DUMS)
      PRINT 1054,KA,ALABEL(1,KA),ALABEL(2,KA),DUMS
 1054 FORMAT(1H ,5X,I4,5X,2A10, 5X,1PE12.5)
 1056 CONTINUE
C
 1057 KPR=0 $ DO 1065 KA=1,NR
      IF(LSTX(KA).NE.0) GOTO 1065 $ IF(LSTN(KA).EQ.0) GOTO 1065
      IF(KPR.EQ.0) PRINT 1059 $ KPR=1
 1059 FORMAT(1H ,//,1H ,32HTHE FOLLOWING SETS HAVE NOT BEEN,
     A25H USED IN THE CALCULATIONS,//,1H ,3X,7HSET NR.,11X,
     B6HLABELS,12X,4HLSTN,/)
      PRINT 1062,KA,ALABEL(1,KA),ALABEL(2,KA),LSTN(KA)
 1062 FORMAT(1H ,5X,I4,5X,2A10, 3X,I7)
 1065 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MTRINDB(A,NX,RS,NA,KIN,DET,W)
      DOUBLE PRECISION A,RS,DET,D1,D2,W
C  MATRIX INVERSION ROUTINE
C  NX = ACTUAL DIMENSION OF A
C  NA = DIMENSION OF A(NA,NA) AS DECLARED BY DIMENSION STATEMENT
C  W MUST HAVE THE LENGTH  NA*(3+NA) OR MORE
C  KIN=0 - COMPUTE INVERSE.  KIN=1 - SOLVE ALSO  A*X=RS.
C  AT RETURN A IS REPLACED BY ITS INVERSE AND RS IS REPLACED BY THE
C  SOLUTION X (THE LATTER IF KIN=1)
C  USES SUBROUTINES LUDATD AND LUELMD
      DIMENSION A(NA,1),RS(1),W(NA,1)
      LEVEL 2,A,RS,W
      DET=0
      IF(NX.LE.0.OR.NX.GT.NA)GOTO 55
      IF(KIN.LT.0.OR.KIN.GT.1) GOTO 55
      DO 15 KA=1,NX  $  DO 15 KB=1,NX
   15 W(KA,KB)=A(KA,KB)
      CALL LUDATD(W,W,NX,NA,D1,D2,W(1,NA+1),W(1,NA+2),NBAD)
      IF(NBAD.NE.0) RETURN
      DET=D1*2.**D2
      DO 35 KA=1,NX
      DO 25 KB=1,NX
   25 W(KB,NA+3)=0
      W(KA,NA+3)=1.
      CALL LUELMD(W,W(1,NA+3),W(1,NA+1),NX,NA,A(1,KA))
 35   CONTINUE
      IF(KIN.EQ.1)CALL LUELMD(W,RS,W(1,NA+1),NX,NA,RS)
      RETURN
   55 PRINT 65,NX,NA,KIN
      RETURN
   65 FORMAT(1H ,10X,26HERROR CALLING MTRINDB. NX=,I4,
     A7H,   NA=,I4,7H,   KIN=,I4)
      END
```

```
      SUBROUTINE LUDATD (A,UL,N,IA,D1,D2,IPVT,EQUIL,IER)
      DOUBLE PRECISION A,UL,D1,D2,EQUIL,P,Q,SUM,BIG,RN
C
C   FUNCTION            - L-U DECOMPOSITION BY THE CROUT ALGORITHM
C   USAGE               - CALL LUDATD(A,UL,N,IA,D1,D2,IPVT,EQUIL,IER)
C   PARAMETERS   A      - INPUT MATRIX OF DIMENSION N BY N CONTAINING
C                           THE MATRIX TO BE DECOMPOSED
C                UL     - REAL OUTPUT MATRIX OF DIMENSION N BY N
C                           CONTAINING THE L-U DECOMPOSITION OF A
C                           ROWWISE PERMUTATION OF THE INPUT MATRIX.
C                N      - INPUT SCALAR CONTAINING THE ORDER OF THE
C                           MATRIX A.
C                IA     - INPUT SCALAR CONTAINING THE ROW DIMENSION OF
C                           MATRICES A AND LU IN THE CALLING PROGRAM.
C                D1     - OUTPUT SCALAR CONTAINING ONE OF THE TWO
C                           COMPONENTS OF THE DETERMINANT. SEE
C                           DESCRIPTION OF PARAMETER D2, BELOW.
C                D2     - OUTPUT SCALAR CONTAINING ONE OF THE
C                           TWO COMPONENTS OF THE DETERMINANT. THE
C                           DETERMINANT MAY BE EVALUATED AS (D1)(2**D2)
C                IPVT   - OUTPUT VECTOR OF LENGTH N CONTAINING THE
C                           PERMUTATION INDICES. SEE DOCUMENT
C                           (ALGORITHM).
C                EQUIL  - OUTPUT VECTOR OF LENGTH N CONTAINING
C                           RECIPROCALS OF THE ABSOLUTE VALUES OF
C                           THE LARGEST (IN ABSOLUTE VALUE) ELEMENT
C                           IN EACH ROW.
C                IER    - ERROR PARAMETER
C                         = 0  MEANS NO ERROR
C                         = 129  MEANS THAT MATRIX  A  IS
C                           ALGORITHMICALLY SINGULAR
C   PRECISION           - DOUBLE
C   LANGUAGE            - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION     - AUGUST 15, 1973
C    CHANGE TO DOUBLE PRECISION AT BRL   -  12 APRIL 1979
C
      DIMENSION          A(IA,1),UL(IA,1),IPVT(1),EQUIL(1)
      LEVEL 2,A,UL,IPVT,EQUIL
C                                  INITIALIZATION
      IER = 0
      RN = N  $ D1=1.0  $  D2=0.0
      DO 10 I=1,N    $ BIG=0.0
         DO 5 J=1,N
            P = A(I,J)
            UL(I,J) = P
      IF(P.LT.0.0) P=-P
            IF (P .GT. BIG) BIG = P
    5    CONTINUE
         IF (BIG .EQ. 0.0 ) GO TO 110
         EQUIL(I) = 1.0/BIG
   10 CONTINUE
      DO 105 J=1,N
         JM1 = J-1
         IF (JM1 .LT. 1) GO TO 40
C                                  COMPUTE U(I,J), I=1,...,J-1
         DO 35 I=1,JM1
```

```
               SUM = UL(I,J)
               IM1 = I-1
   25          IF (IM1 .LT. 1) GO TO 35
               DO 30 K=1,IM1
                  SUM = SUM-UL(I,K)*UL(K,J)
   30          CONTINUE
               UL(I,J) = SUM
   35       CONTINUE
   40       P = 0.0
C                                              COMPUTE U(J,J) AND L(I,J), I=J+1,...,
            DO 70 I=J,N
               SUM = UL(I,J)
   55          IF (JM1 .LT. 1) GO TO 65
               DO 60 K=1,JM1
                  SUM = SUM-UL(I,K)*UL(K,J)
   60          CONTINUE
               UL(I,J) = SUM
65       Q=EQUIL(I)*SUM  $  IF(Q.LT.0.0) Q=-Q
               IF (P .GE. Q) GO TO 70
               P = Q
               IMAX = I
   70       CONTINUE
C                                              TEST FOR ALGORITHMIC SINGULARITY
            IF (RN+P .EQ. RN) GO TO 110
            IF (J .EQ. IMAX) GO TO 80
C                                              INTERCHANGE ROWS J AND IMAX
            D1 = -D1
            DO 75 K=1,N
               P = UL(IMAX,K)
               UL(IMAX,K) = UL(J,K)
               UL(J,K) = P
   75       CONTINUE
            EQUIL(IMAX) = EQUIL(J)
   80       IPVT(J) = IMAX
            D1 = D1*UL(J,J)
85       IF(D1*D1.LE.1.0) GOTO 90
            D1 = D1/16.0      $  D2=D2+4.0
            GO TO 85
90       IF(D1.GE.0.0625 .OR. D1.LE.-0.0625) GOTO 95
            D1 = D1*16.0      $  D2=D2-4.0
            GO TO 90
   95       CONTINUE
            JP1 = J+1
            IF (JP1 .GT. N) GO TO 105
C                                              DIVIDE BY PIVOT ELEMENT U(J,J)
            P = UL(J,J)
            DO 100 I=JP1,N
               UL(I,J) = UL(I,J)/P
  100       CONTINUE
  105 CONTINUE
      RETURN
C                                              ALGORITHMIC SINGULARITY
  110 IER = 129
      D1=0.0  $  D2=0.0
 9005 RETURN
      END
```

```
      SUBROUTINE LUELMD (A,B,IPVT,N,IA,X)
      DOUBLE PRECISION A,B,X,SUM
C
C   FUNCTION            - ELIMINATION PART OF SOLUTION OF AX=B -
C                           FULL STORAGE MODE
C   USAGE               - CALL LUELMD (A,B,IPVT,N,IA,X)
C   PARAMETERS   A      - THE RESULT, LU, COMPUTED IN THE SUBROUTINE
C                           *LUDATD*, WHERE L IS A LOWER TRIANGULAR
C                           MATRIX WITH ONES ON THE MAIN DIAGONAL. U IS
C                           UPPER TRIANGULAR, L AND U ARE STORED AS A
C                           SINGLE MATRIX A, AND THE UNIT DIAGONAL OF
C                           L IS NOT STORED
C                B      - B IS A VECTOR OF LENGTH N ON THE RIGHT HAND
C                           SIDE OF THE EQUATION AX=B
C                IPVT   - THE PERMUTATION MATRIX RETURNED FROM THE
C                           SUBROUTINE *LUDATD*, STORED AS AN N LENGTH
C                           VECTOR
C                N      - ORDER OF A AND NUMBER OF ROWS IN B
C                IA     - NUMBER OF ROWS IN THE DIMENSION STATEMENT
C                           FOR A IN THE CALLING PROGRAM.
C                X      - THE RESULT X
C   PRECISION           - DOUBLE
C   LANGUAGE            - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION     - APRIL 11,1975
C    CHANGE TO DOUBLE PRECISION AT BRL   -  12 APRIL 1979
C
      DIMENSION          A(IA,1),B(1),IPVT(1),X(1)
      LEVEL 2,A,B,IPVT,X
C                                  SOLVE LY = B FOR Y
      DO 5 I=1,N
    5 X(I) = B(I)
      IW = 0
      DO 20 I=1,N
         IP = IPVT(I)
         SUM = X(IP)
         X(IP) = X(I)
         IF (IW .EQ. 0) GO TO 15
         IM1 = I-1
         DO 10 J=IW,IM1
            SUM = SUM-A(I,J)*X(J)
   10    CONTINUE
         GO TO 20
   15    IF (SUM .NE. 0.) IW = I
   20 X(I) = SUM
C                                  SOLVE UX = Y FOR X
      DO 30 IB=1,N
         I = N+1-IB
         IP1 = I+1
         SUM = X(I)
         IF (IP1 .GT. N) GO TO 30
         DO 25 J=IP1,N
            SUM = SUM-A(I,J)*X(J)
   25    CONTINUE
   30 X(I) = SUM/A(I,I)
      RETURN
      END
```

APPENDIX C
BLAST FIELD HISTORY COMPUTATION PROGRAM BLAFHI

APPENDIX C (continued)

```
      PROGRAM HISTORY(INPUT,OUTPUT,TAPE6=OUTPUT,TAPE13)
C  THIS PROGRAM COMPUTES FLOW HISTORIES AT SPECIFIED LOCATIONS

      COMMON/COMFLD/FPAR(5),VFPAR(5,5),SCD,SCP,SCT,RMIN,RMAX
      COMMON/CFLDEX/EXNU(3)
C  /COMFLD/ AND /CFLDEX/ ARE SHARED WITH READFP

      DIMENSION X(5,100),R(5,5,100),UTEST(100)
      DIMENSION PAR(10),VPAR(10,10),TITLE(3),SCV(10)
C
      CALL READAM(SD,SP,ST,TITLE,NBAD)
C         READ AMBIENT DATA
      IF(NBAD.NE.0.AND.NBAD.NE.3) STOP
C
      CALL READSP(NBAD)
C     THIS READS SHOCK FITTING RESULTS.THE PARAMETERS AND THEIR
C     ACCURACIES WILL BE STORED IN THE PROPER COMMON STORAGES
      IF(NBAD.EQ.0) GO TO 5
      PRINT 2,NBAD
2     FORMAT(1H0,10X,*ERROR RETURN FROM READSP WITH NBAD=*,I10)
      STOP
C
    5 CONTINUE
      CALL READFP(NBAD)
C  READ IN PARAMETERS OF THE OVERPRESSURE FIELD FUNCTION
C  THE RESULTS ARE IN /COMFLD/ AND /CFLDEX/
      IF (NBAD.EQ.0) GO TO 10
      PRINT 7,NBAD
7     FORMAT(1H0,10X,*ERROR RETURN FROM READFP WITH NBAD=*,I10)
      STOP
C
   10 CONTINUE
C  NEXT EXPRESS FIELD PARAMETERS IN SCALES SPECIFIED BY READAM
      SCV(1)=(SCD/SD)**EXNU(1)/(SCT/ST)
      SCV(2)=(SCD/SD)**(EXNU(1)-1.)/(SCT/ST)
      SCV(3)=(SCD/SD)**EXNU(2)/(SCT/ST)**2
      SCV(4)=(SCD/SD)**(EXNU(2)-1.)/(SCT/ST)**2
      SCV(5)=(SCD/SD)**EXNU(3)*(SCP/SP)
      DO 20 KA=1,5  $  DO 15 KB=1,5
15    VPAR(KA,KB)=VFPAR(KA,KB)*SCV(KA)*SCV(KB)
20    PAR(KA)=FPAR(KA)*SCV(KA)

      NP=9
C  NP IS THE TOTAL NUMBER OF PARAMETERS.  PAR WILL BE SUPPLEMENTED
C  IN FLOFLD WITH SHOCK PARAMETERS
C
25    READ 35,TA,TB,DHIST,TMAX,ANR
C  READ AN INSTRUCTION CARD FOR HISTORY COMPUTATION
35    FORMAT(2A10,6E10.3)
      PRINT 36,TA,TB,DHIST,TMAX,ANR
36    FORMAT(1H1,//,1H ,10X,*INPUT READ BY HISTORYMAIN*,/,1H0,5X,2A10,
     A 6(2X,1PE14.7))
      IF(TA.NE.10H          ) GOTO 55
      PRINT 45  $  STOP
45    FORMAT(1H0,10X,*STOP BECAUSE FIRST FIELD OF INPUT CARD IS BLANK*)

55    PRINT 65
```

```
65    FORMAT(1HO,5X,*THE CARD CONTAINS DISTANCE, MAXIMUM TIME AND*,
     A * THE DESIRED NUMBER OF NODES*,/,1H ,5X,*FLOW HISTORY WILL*,
     B * BE CALCULATED AT THE GIVEN DISTANCE AND UP TO THE MAXIMUM*,
     C * TIME.*,/,1H ,5X,*COMPUTING SCALES ARE SPECIFIED BY *
     D ,*AMBIENT DATA INPUT*)
      PRINT 75
75    FORMAT(1HO,10X,*THE PRESENT INPUT IS ASSUMED TO BE IN SI UNITS*)

      RMINS=RMIN*SCD/SD  $  RMAXS=RMAX*SCD/SD
      DHISTS=DHIST/SD  $  TMAXS=TMAX/ST
      NRHIST=ANR

      CALL FLOFLD(SD,SP,ST,RMINS,RMAXS,DHISTS,TMAXS,PAR,VPAR,NP,
     A X,R,NRHIST,UTEST,NUTEST,NBAD)
      IF(NBAD.NE.0) PRINT 85,NBAD
85    FORMAT(1HO,10X,*ERRROR RETURN FROM FLOFLD WITH NBAD=*,I10,/
     A/,1HO,10X,*NEXT TRY TO PLOT THE RESULT*)
C  THIS COMPUTED AND PRINTED THE FLOW FIELD AT DHIST
      CALL PLFFLD(SD,SP,ST,DHISTS,  X,R,NRHIST,UTEST,NUTEST,TITLE)
C  THIS PLOTTED THE RESULTS OF FLOFLD
      GOTO 25
C
      END
```

```
      SUBROUTINE READAM(SCDIST,SCPRES,SCTIME,TITLE,NBAD)
C THIS ROUTINE READS TITLE, PLOTLABEL AND DATA CARDS DESCRIBING
C AMBIENT CONDITIONS AND THE CHARGE
C FIRST TWO CARDS ARE MANDATORY AND ALPHANUMERIC  (TITLE AND PLOTLABEL
C THE REST OF THE CARDS HAVE THE FORMAT (2A10,6E10.3)
C CHARGE CARD IS MANDATORY
C IF AMBIENT DATA ARE NOT PROVIDED THEN STANDARD AIR WILL BE ASSUMED
C
C SEQUENCE OF MANDATORY INPUT CARDS
C     TITLE CARD  (ALPHANUMERIC)
C     PLOTLABEL CARD   (ALPHANUMERIC)
C     CHARGE CARD = VOLUME, ENERGY, HIGHT, ERROR OF HIGHT
C
C THE FOLLOWING ARE OPTIONAL INPUT CARDS IN ARBITRARY SEQUENCE
C     AMBIENT = P,TEMPERATURE, GAMMA, MOLAR MASS
C         DEFAULT VALUES CORRESPOND TO A STANDARD AIR
C     SCALES = SCALES OF R,P,T  TO BE USED IN COMPUTATIONS
C          DEFAULT VALUES ARE COMPUTED AFTER STATEMENT 1110
C     PLOTTING DATA = ERROR FACTORS FOR THE PLOTTING OF CONFIDENCE
C                     LIMITS IN HISTORY PLOTS
C          DEFAULT VALUES ARE FACTORS 2.0 FOR ALL PLOTS
C
C END OF INPUT IS INDICATED BY A BLANK CARD
C
      DIMENSION TITLE(3)
      DIMENSION D(8),AMSTAR(4)
      COMMON/AMBCHA/AIRPR,AIRTEM,AIRGAM,AIRMOL,CHARVO,CHAREN,
     ACHARHI,CHARHER
      COMMON/PLOT/PD(6),PLABL(4)
      DATA(TITL =10HTITLE     ), (PLAB=10HPLOTLABEL )
      DATA (BLANK=10H          ),(AMB=10HAMBIENT   )
      DATA (CHA=10HCHARGE    )
      DATA(PLT=10HPLOTTING D),(SCAL=10HSCALES R,P)
   15 FORMAT(1H1,10X,20HINPUT READ BY READAM,/,1H ,10X,20(1H-),/)
   25 FORMAT(8A10)
   26 FORMAT(1H ,10X,8A10)
   35 FORMAT(2A10,6E10.3)
   36 FORMAT(1H , 5X,2A10,6(2X,1PE14.7))
C
      PD(1)=2.0
C DEFAULT VALUE FOR PLOTTING ERROR LIMITS IN PRESSURE HISTORIES
      PD(2)=2.0
C DEFAULT VALUE FOR PLOTTING FIELD HISTORIES (P,V,RHO,V**2*RHO/2.)
      AIRPR=101325.0  $  AIRTEM=293.0  $  AIRGAM=1.4
      AIRMOL=0.02896  $  AIRDEN=(AIRMOL/8.3143)*(AIRPR/AIRTEM)
C THESE ARE STANDARD AIR DEFAULT VALUES FOR AMBIENT CONDITIONS
C
      NSCAL=0   $ NAMSTAR=0
      NAMB=0 $ NCHA=0
      DO 37 J=1,4
   37 AMSTAR(J)=1H
      PRINT 15
      DO 46 KK=1,2
      READ 25,(D(J),J=1,8)
      PRINT 26,(D(J),J=1,8)
      IF(D(1).EQ.TITL ) GOTO 42
      IF(D(1).EQ.PLAB) GOTO 44
```

```
      PRINT 48 $ NBAD=1 $ RETURN
C
42    DO 43 KA=1,3
43    TITLE(KA)=D(KA+1)
      GOTO 46
44    DO 45 KA=1,4
45    PLABL(KA)=D(KA+1)
46    CONTINUE
C
   47 READ 35,(D(J),J=1,8)
      PRINT 36,(D(J),J=1,8)
      IF(D(1).EQ.AMB)GOTO 55
      IF(D(1).EQ.CHA)GOTO 65
      IF(D(1).EQ.PLT) GOTO 66
      IF(D(1).EQ.SCAL) GOTO 68
      IF(D(1).EQ.BLANK) GOTO 69
  475 PRINT 48 $ NBAD=2 $ RETURN
   48 FORMAT(1H0,10X,13HINVALID INPUT)
C
   55 IF(NAMB.EQ.1)GOTO 475
C  ONLY ONE AMBIENT DATA CARD WILL BE CONSIDERED
      NAMB=1
      IF(D(3).GT.0.)AIRPR=D(3)  $  IF(D(4).GT.0.)AIRTEM=D(4)
      IF(D(5).GT.0.)AIRGAM=D(5)  $  IF(D(6).GT.0.)AIRMOL=D(6)
C  IF INPUT IS ZERO THEN USE AIR DEFAULT VALUES
      DO 57 KA=1,4  $  AMSTAR(KA)=1H
      IF(D(KA+2).GT.0.) GOTO 57
      AMSTAR(KA)=1H*  $  NAMSTAR=1
57    CONTINUE
      AIRDEN=(AIRMOL/8.3143)*(AIRPR/AIRTEM)
      GOTO 47
C
   65 IF(NCHA.EQ.1)GOTO 475
      CHARVO=D(3) $ CHAREN=D(4)
      CHARHI=D(5) $ CHARHER=D(6)
      NCHA=1
      GOTO 47
C
66    DO 67 KA=1,6
67    PD(KA)=D(KA+2)
      GOTO 47
C  PLOTTING DATA CARD SPECIFIES PLOTTED OUTPUT
C  PD(1)= ERROR FACTOR FOR PRESSURE HISTORIES
C  PD(2)= ERROR FACTOR FOR OTHER FLOW HISTORIES
C
68    NSCAL=1
      SCD=D(3)  $  SCP=D(4)  $  SCT=D(5)
C  SCALE CARD OVERRIDES SCALES COMPUTED FROM AMBIENT AND CHARGE DATA
      IF(SCD.GT.0..AND.SCP.GT.0..AND.SCT.GT.0.) GOTO 47
      NSCAL=0  $  PRINT 681
681    FORMAT(1H ,10X,36HNON-POSITIVE SCALES ARE NOT ACCEPTED)
      GOTO 47
C
69    IF(NCHA.EQ.0.OR.NAMB.EQ.0) PRINT 70
70    FORMAT(1H0,10X,16HINCOMPLETE INPUT)
   75 PRINT106,(TITLE(J),J=1,3)
106   FORMAT(1H1,/,1H ,10X,5HEVENT,/,1H ,10X, 5(1H-),/,1H0,15X,3A10,//)
```

```
      PRINT 107
  107 FORMAT(1H0,10X,18HAMBIENT CONDITIONS,/,1H ,10X,18(1H-),/)
      IF(NAMB.EQ.0) PRINT 1071
 1071  FORMAT(1H0,10X,36HTHE FOLLOWING AMBIENT CONDITIONS ARE,
     A /,1H ,10X,27HSTANDARD AIR DEFAULT VALUES,/)
      PRINT 108,AMSTAR(1),AIRPR,AMSTAR(2),AIRTEM,AMSTAR(3),AIRGAM,
     A AMSTAR(4),AIRMOL
  108 FORMAT(1H ,13X,A1,1X,8HPRESSURE,11X,7HAIRPR =,1PE12.5,4H  PA,/,
     A 1H ,13X,A1,1X,11HTEMPERATURE,8X,7HAIRTEM=,1PE12.5,3H  K,/,
     B 1H ,13X,A1,1X,16HSPEC. HEAT RATIO,3X,7HAIRGAM=,1PE12.5,/,
     C 1H ,13X,A1,1X,10HMOLAR MASS,9X,7HAIRMOL=,1PE12.5,9H  KG/MOLE,/)
      AIRSND=SQRT(AIRGAM*AIRPR/AIRDEN)
      PRINT 109,AIRSND,AIRDEN
  109 FORMAT(1H ,15X,11HSOUND SPEED,8X,7HAIRSND=,1PE12.5,5H  M/S,/,
     A 1H ,15X,7HDENSITY,12X,7HAIRDEN=,1PE12.5,9H  KG/M**3,/)
      IF(NAMSTAR.EQ.1) PRINT 1081
 1081  FORMAT(1H ,13X,35H* THE STARRED DATA ARE STANDARD AIR,
     A 15H DEFAULT VALUES,/)

      IF(NCHA.EQ.1) GOTO 1100
      NBAD=4  $  PRINT 1101,NBAD  $  RETURN
 1101  FORMAT(1H0,10X,29HRETURN FROM READAM WITH NBAD=,I2,
     A 33H, BECAUSE CHARGE DATA ARE MISSING)
C
 1100  PRINT 110
  110 FORMAT(1H0,10X,18HCHARGE DESCRIPTION,/,1H ,10X,18(1H-),/)
      PRINT 111, CHARVO,CHAREN
  111 FORMAT(1H ,15X,13HCHARGE VOLUME,6X,7HCHARVO=,1PE12.5,6H  M**3,/,
     A 1H ,15X,13HCHARGE ENERGY,6X,7HCHAREN=,1PE12.5,3H  J,/)
      SCDIST=CHARVO**(1./3.)
      PRINT 1110,CHARHI,CHARHER
 1110 FORMAT(1H ,15X,16HCHARGE ELEVATION,3X,7HCHARHI=,1PE12.5,4H +- ,
     A 1PE12.5,3H  M,/)
      SCTIME=SCDIST/AIRSND
      SCPRES=AIRPR
      SCEVEN=CHAREN/(CHARVO*AIRPR)
      PRINT 112
  112 FORMAT(1H0,10X,7HSCALING,/,1H ,10X,7(1H-),/)
      PRINT 113,SCDIST,SCTIME,SCPRES,SCEVEN
  113 FORMAT(1H ,15X,12HLENGTH SCALE,4X,20HSCDIST=CHARVO**(1/3),
     A 2X,1H=,1PE12.5,3H  M,/,
     B 1H ,15X,10HTIME SCALE,6X,20HSCTIME=SCDIST/AIRSND,
     C 2X,1H=,1PE12.5,3H  S,/,
     D 1H ,15X,14HPRESSURE SCALE,2X,13HSCPRES=AIRPR ,
     E 9X,1H=,1PE12.5,4H  PA,/,
     F 1H ,15X,14HSCALE OF EVENT,2X,21HCHAREN/(CHARVO*AIRPR),
     G 1X,1H=,1PE12.5,/)
      IF(SCEVEN.EQ.0.0)PRINT 114
  114 FORMAT(1H ,15X,30HEVENT CANNOT BE SCALED BECAUSE,
     A29H CHAREN IS NOT GIVEN BY INPUT,/)

      IF(NSCAL.EQ.0) GOTO 115
C  USE SCALES FROM SCALE CARD IF SUCH A CARD WAS READ
      SCDIST=SCD  $  SCPRES=SCP  $  SCTIME=SCT

  115  PRINT 116,SCDIST,SCTIME,SCPRES
  116 FORMAT(1H ,////,1H ,10X,27HSCALES USED IN THIS PROGRAM,/,
```

```
A 1H ,10X,27(1H-),//,1H ,20X,16HLENGTH SCALE    =,1PE12.5,3H  M,/,
B 1H ,20X,16HTIME SCALE      =,1PE12.5,3H  S,/,
C 1H ,20X,16HPRESSURE SCALE =,1PE12.5,4H  PA)
  NBAD=0
  RETURN
  END
```

```
      SUBROUTINE READSP(NBAD)
C
C  THIS ROUTINE READS SHOCK PARAMETERS NAD THEIR ACCURACIES
C
      COMMON/COMSHK/NPS,PAR(4),VPAR(4,4),SCD,SCP,SCT
      COMMON/CF2DER/GAMCAP,SNDSPD,CFPAR(4),ALOW,CFSCD,CFSCP,CFSCT
      COMMON/AMBCHA/AMP,AMT,AMG,AMM,      AMCHV,AMCHE,AMCHH,AMCHHE
C
      DIMENSION DAT(8),ER(4),COR(4,4)
      DIMENSION DSI(4),DSC(4),DPR(4)
C
      DATA(PL=10HSHOCKPAR  ),(EL=10HSHOCKPARER),(CL=10HSHOCKPARCO),
     A (SC=10HSHOCKSCALE),(BL=10H          )
C
      DATA DSI/10HPA*M      ,10HPA*M**2   ,10HPA*M**3   ,
     A 10HS         /
      DATA DSC/10HSCP*SCD   ,10HSCP*SCD**2,10HSCP*SCD**3,
     A 10HSCT       /

      KPL=1  $  KEL=1  $  KCL=1  $  KSC=1
      PRINT 12
12    FORMAT(1H1,10X,20HINPUT READ BY READSP,/)
15    FORMAT(2A10,6E10.3)
25    FORMAT(1H ,5X,2A10,6(2X,1PE14.7))
35    READ 15,(DAT(J),J=1,8)
      PRINT 25,(DAT(J),J=1,8)
      IF(DAT(1).EQ.PL) GOTO 55
      IF(DAT(1).EQ.EL) GOTO 75
      IF(DAT(1).EQ.CL) GOTO 95
      IF(DAT(1).EQ.SC) GOTO 115
      IF(DAT(1).EQ.BL) GOTO 125
      NBAD=1
      PRINT 45  $  RETURN
45    FORMAT(1H0,10X,13HINVALID INPUT)
C
55    DO 65 KA=1,4
65    PAR(KA)=DAT(KA+2)
      DALOW=DAT(7)
      IF(DALOW.GE.1.0E-90) GOTO 67
      PRINT 66,DAT(6)
66     FORMAT(1H ,10X,'5-TH NUMBER ON PREVIOUS CARD SHOULD BE '
     A 'POSITIVE INDICATING SHOCK DISTANCE AT T='1PE12.5)
      NBAD=66  $  PRINT 45
      RETURN
67    CONTINUE
      KPL=0
      GOTO 35
C
75    DO 85 KA=1,4
85    ER(KA)=DAT(KA+2)
      KEL=0
      GOTO 35
C
95    COR(1,1)=1.  $  COR(2,2)=1.  $  COR(3,3)=1.  $  COR(4,4)=1.
      COR(1,2)=DAT(3)  $  COR(2,1)=COR(1,2)
      COR(1,3)=DAT(4)  $  COR(3,1)=COR(1,3)
      COR(1,4)=DAT(5)  $  COR(4,1)=COR(1,4)
```

```
      COR(2,3)=DAT(6)  $  COR(3,2)=COR(2,3)
      COR(2,4)=DAT(7)  $  COR(4,2)=COR(2,4)
      COR(3,4)=DAT(8)  $  COR(4,3)=COR(3,4)
      KCL=0
      GOTO 35
C
115   SCD=DAT(3)  $  SCP=DAT(4)  $  SCT=DAT(5)
      KSC=0
      GOTO 35
C
125   IF(KPL.EQ.0.AND.KEL.EQ.0.AND.KCL.EQ.0.AND.KSC.EQ.0)GOTO 145
      NBAD=2
      PRINT 135  $  RETURN
135   FORMAT(1H0,10X,16HINCOMPLETE INPUT)
C
145   NPS=4
      ALOW=DALOW*SCD
      GAMCAP=((1.+AMG)/(2.*AMG))/AMP
      SNDSPD=SQRT(AMG*AMT*(8.3143/AMM))
      CFSCD=1.  $  CFSCP=1.  $  CFSCT=1.
C  /CF2DER/ IS NEEDED FOR SHOCK ARRIVAL TIME COMPUTATIONS
      DO 155 KA=1,4  $  DO 155 KB=1,4
155   VPAR(KA,KB)=ER(KA)*COR(KA,KB)*ER(KB)
      NBAD=0
      PRINT 165
  165 FORMAT(1H0,12X,16HSHOCK PARAMETERS,4X,6HERRORS,5X,
     A 10HDIMENSIONS,/)
      IF(SCD.EQ.1..AND.SCP.EQ.1..AND.SCT.EQ.1.) GOTO 167
      DO 166 KA=1,4
  166 DPR(KA)=DSC(KA)
      DISDI=10HSCD
      GOTO 169
  167 DO 168 KA=1,4
  168 DPR(KA)=DSI(KA)
      DISDI=10HMETRES
  169 PRINT 175,((PAR(J),ER(J),DPR(J)),J=1,4)
  175 FORMAT(1H ,14X,1PE12.5,4X,1PE10.3,2X,A10)
      PRINT 178,DALOW,DISDI
  178 FORMAT(1H0,10X,43HTHE LAST PARAMETER IS SHOCK ARRIVAL TIME AT,
     A 2X,1PE12.5,2X,A10)
      PRINT 185
185   FORMAT(1H ,///,1H ,15X,*SHOCK PARAMETER CORRELATION MATRIX*,/)
      PRINT 195,((COR(J,K),K=1,4),J=1,4)
195   FORMAT(4(1H ,10X,4(2X,0PF10.7),/))
      PRINT 205
205   FORMAT(1H ,///,1H ,15X,16HSHOCK PARAMETER ,
     A 26HVARIANCE-COVARIANCE MATRIX,/)
      PRINT 215,((VPAR(J,K),K=1,4),J=1,4)
215   FORMAT(4(1H ,10X,4(2X,1PE12.5),/))
      PRINT 225
225   FORMAT(1H ,///,1H ,16X,22HSHOCK PARAMETER SCALES,/)
      PRINT 235,SCD,SCP,SCT
  235 FORMAT(1H ,15X,12HLENGTH SCALE,4X,5HSCD =,1PE12.5,3H  M,/,
     A 1H ,15X,14HPRESSURE SCALE,2X,5HSCP =,1PE12.5,4H  PA,/,
     B 1H ,15X,10HTIME SCALE,6X,5HSCT =,1PE12.5,3H  S)
      RETURN
      END
```

```
      SUBROUTINE READFP(NBAD)
C     THIS READS OVERPRESSURE FIELD FUNCTION PARAMETERS
C
      COMMON/CFLDEX/EXNU(3)
      COMMON/COMFLD/FPAR(5),VFPAR(5,5),SCD,SCP,SCT,RMIN,RMAX
C  /COMFLD/ IS AVAILABLE TO THE MAIN PROGRAM
C
      DIMENSION DAT(8),ER(5),COR(5,5)
      DIMENSION DIMA(5),DIMB(5)
C
      DATA(FP=10HFIELDPAR  ),(FE=10HFIELDPARER),(FS=10HFIELDPARSC)
     1  ,(FC=10HFIELDPARCO),(BL=10H          )
      DATA(EX=10HFIELDPAREX),(RA=10HFIELDPARRA)
      DATA (COR1=10H 1        ),(COR2=10H 2        )
C
      PRINT 12
   12 FORMAT(1H1,10X,*INPUT READ BY READFP*,/)
   15 FORMAT(2A10,6E10.3)
   25 FORMAT(1H ,5X,2A10,6(2X,1PE14.7))
   35 READ 15,(DAT(J),J=1,8)
      PRINT 25,(DAT(J),J=1,8)
      IF(DAT(1).EQ.FP) GO TO 55
      IF(DAT(1).EQ.FE) GO TO 75
      IF(DAT(1).EQ.FS) GO TO 95
      IF(DAT(1).EQ.FC) GO TO 115
      IF(DAT(1).EQ.BL) GO TO 125
      IF(DAT(1).EQ.EX) GO TO 135
      IF(DAT(1).EQ.RA) GOTO 145
38    NBAD=1
      PRINT 45
   45 FORMAT(1H ,10X,*INVALID INPUT*)
      RETURN
C
   55 DO 65 KA=1,5
      FPAR(KA)=DAT(KA+2)
   65 CONTINUE
      GO TO 35
   75 DO 85 KA=1,5
      ER(KA)=DAT(KA+2)
   85 CONTINUE
      GO TO 35
   95 SCD=DAT(3) $ SCP=DAT(4) $ SCT=DAT(5)
      GO TO 35
115   IF(DAT(2).EQ.COR1) GOTO 116
      IF(DAT(2).EQ.COR2) GOTO 120
      GOTO 38
116   COR(1,1)=1. $ COR(2,2)=1. $ COR(3,3)=1.
      COR(4,4)=1. $ COR(5,5)=1.
      COR(1,2)=DAT(3) $ COR(2,1)=DAT(3)
      COR(1,3)=DAT(4) $ COR(3,1)=DAT(4)
      COR(1,4)=DAT(5) $ COR(4,1)=DAT(5)
      COR(1,5)=DAT(6) $ COR(5,1)=DAT(6)
      COR(2,3)=DAT(7) $ COR(3,2)=DAT(7)
      GO TO 35
  120 COR(2,4)=DAT(3) $ COR(4,2)=DAT(3)
      COR(2,5)=DAT(4) $ COR(5,2)=DAT(4)
      COR(3,4)=DAT(5) $ COR(4,3)=DAT(5)
```

```
      COR(3,5)=DAT(6) $ COR(5,3)=DAT(6)
      COR(4,5)=DAT(7) $ COR(5,4)=DAT(7)
      GO TO 35
  135 DO 160 KA=1,3
      EXNU(KA)=DAT(KA+2)
  160 CONTINUE
      GO TO 35
145   RMIN=DAT(3)  $  RMAX=DAT(4)
      GOTO 35
C
C  ENTER 125 WHEN BLANK CARD INDICATES END OF DATA
  125 DO 155 KA=1,5
      DO 155 KB=1,5
      VFPAR(KA,KB)=ER(KA)*COR(KA,KB)*ER(KB)
  155 CONTINUE
      NBAD=0
C  NOW PRINT COMPREHENSIVE LIST OF INPUT
      PRINT 165
  165 FORMAT(1H0,12X,16HFIELD PARAMETERS,3X,10HSTD.ERRORS,4X,
     A 10HDIMENSIONS,/)
      DIMA(1)=10HM**EXA/S  $ DIMB(1)=10H
      DIMA(2)=10HM**(EXA-1)  $ DIMB(2)=10H/S
      DIMA(3)=10HM**EXB/S**$ DIMB(3)=10H2
      DIMA(4)=10HM**(EXB-1)  $ DIMB(4)=10H/S**2
      DIMA(5)=10HM**EXC*PA $ DIMB(5)=10H
      IF(SCT.EQ.1..AND.SCD.EQ.1..AND.SCP.EQ.1.)GOTO 168
      DIMA(1)=10HSCD**EXA/S  $ DIMB(1)=10HCT
      DIMA(2)=10HSCD**(EXA-  $ DIMB(2)=10H1)/SCT
      DIMA(3)=10HSCD**EXB/S  $ DIMB(3)=10HCT**2
      DIMA(4)=10HSCD**(EXB-  $ DIMB(4)=10H1)/SCT**2
      DIMA(5)=10HSCD**EXC*S  $ DIMB(5)=10HCP
  168 CONTINUE
      PRINT 175,((FPAR(J),ER(J),DIMA(J),DIMB(J)),J=1,5)
  175 FORMAT(1H ,14X,1PE12.5,4X,1PE10.3,4X,2A10)
      PRINT 178,RMIN,RMAX
178   FORMAT(1H0,12X,34HTHE PARAMETERS CAN BE USED BETWEEN,
     A 6H RMIN=,1PE12.5,10H AND RMAX=,1PE12.5)
      IF(SCD.EQ.1.)PRINT 1781
 1781 FORMAT(1H+,86X,7H METRES)
      IF(SCD.NE.1.)PRINT 1782
 1782 FORMAT(1H+,86X,4H SCD)
      PRINT 180
  180 FORMAT(1H0,12X,39HEXPONENTS IN OVERPRESSURE FIELD FORMULA,/)
      PRINT 182,EXNU(1),EXNU(2),EXNU(3)
  182 FORMAT(1H ,15X,5HEXA =,F12.2,/,1H ,15X,5HEXB =,
     A F12.2,/,1H ,15X,5HEXC =,F12.2,/)
      PRINT 185
  185 FORMAT(1H ,  /,1H ,15X,*FIELD PARAMETER CORRELATION MATRIX*,/)
      PRINT 195,((COR(J,K),K=1,5),J=1,5)
  195 FORMAT(5(1H ,10X,5(2X,F10.7),/))
      PRINT 205
  205 FORMAT(1H ,///,1H ,15X,*FIELD PARAMETER *,
     A *VARIANCE-COVARIANCE MATRIX*,/)
      PRINT 215,((VFPAR(J,K),K=1,5),J=1,5)
  215 FORMAT(5(1H ,10X,5(2X,1PE12.5),/))
      PRINT 225
  225 FORMAT(1H ,///,1H ,16X,*FIELD PARAMETER SCALES*,/)
```

```
      PRINT 235,SCD,SCP,SCT
  235 FORMAT(1H ,15X,12HLENGTH SCALE,4X,5HSCD= ,1PE12.5,2H M,/,
     A 1H ,15X,14HPRESSURE SCALE,2X,5HSCP= ,1PE12.5,3H PA,/,
     B 1H ,15X,10HTIME SCALE,6X,5HSCT= ,1PE12.5,2H S)
      RETURN
      END
```

```
      SUBROUTINE FLOFLD(SCD,SCP,SCT,RMIN,RMAX,R,TMAX,PAR,VPAR,NPAR,
     A HIST,VHIST,NHIST,UTST,NUTST,NBAD)
C
C  THIS IS CALLED FROM MAIN TO COMPUTE THE FLOW HISTORY A THE
C  DISTANCE R AND FOR TIMES BETWEEN SHOCK ARRIVAL AND TMAX
C
C  SCD,SCP,SCT       = SCALES.  ALL ARGUMENTS ARE IN TERMS OF
C                      THESE SCALES
C  RMIN,RMAX         = RANGE OF PRESSURE FIELD APPROXIMATION
C  R,TMAX            = DISTANCE AND END POINT OF HISTORY
C  PAR,VPAR,NPAR     = PARAMETERS OF PRESSURE FIELD FUNCTION
C      PFIELD AND VARIANCES OF THE PARAMETERS. PAR AND
C      VPAR WILL BE SUPPLEMENTED BY SHOCK PARAMETERS AND THEIR
C      VARIANCES.  NPAR IS IGNORED AND SET EQUAL TO 9.
C  NHIST             = NUMBER OF  NODES TO BE COMPUTED. IT WILL BE
C                      REPLACED BY ACTUALLY COMPUTED NODES.
C
C  THE FOLLOWING WILL BE COMPUTED BY THIS ROUTINE
C
C  HIST(5,NHIST)     = FLOW FIELD HISTORY  (T,P,R,U,RHO,0.5*U**2*RHO)
C  VHIST(5,5,NHIST) = VARIANCE-COVARIANCE MATRICES OF HIST
C  NHIST             = NUMBER OF HISTORY NODES COMPUTED
C  UTST(NUTST)       = PARTICLE VELOCITIES COMPUTED BY TEST PROCESS
C  NUTST             = NUMBER OF TEST VELOCITIES IN UTST
C  NBAD              = ERROR INDICATOR
C
C  ROUTINE USES SUBROUTINES STRBEG, STRLIN AND FLINTER
C
      EXTERNAL PFIELD
C  PRESSURE FIELD FUNCTION
C
      DIMENSION PAR(10),VPAR(10,10),HIST(5,100),VHIST(5,5,100),UTST(100)
C
      DIMENSION SOLIN(6),TPIN(10),XPP(10),UPP(10),UPTP(10),DPIN(10)
      DIMENSION STRNU(6,200),VSTRNU(6,6,200),STROL(6,200),VSTROL(6,6,200
     1)
C
      COMMON/AK3CHA/APRE,ATEM,AGAM,AMOL,CHVOL,CHENE,CHHIG,ECHHIG
      COMMON/CSCALE/SCDI,SCPR,SCTI
      COMMON/COMSHK/NPSH,PARSH(4),VPARSH(4,4),SCDSH,SCPSH,SCTSH
C
      SCDI=SCD $ SCPR=SCP $ SCTI=SCT
C  THESE SCALES ARE NEEDED IN QFUNCT WHICH IS CALLED FROM PFIELD
C
C  NEXT SUPPLEMENT PAR AND VPAR WITH SHOCK PARAMETERS
      DO 8 KA=6,8
      PAR(KA)=PARSH(KA-5)*(SCPSH/SCP)*(SCDSH/SCD)**(KA-5)
      VPAR(KA,9)=VPARSH(KA-5,4)*(SCTSH/SCT)*(SCDSH/SCD)**(KA-5)
     A *(SCPSH/SCP)
      VPAR(9,KA)=VPAR(KA,9)
      DO 8 KB=6,8
      VPAR(KB,KA)=VPARSH(KB-5,KA-5)*(SCPSH/SCP)**2
     A *(SCDSH/SCD)**(KA+KB-10)
    8 CONTINUE
      PAR(9)=PARSH(4)*SCTSH/SCT
      VPAR(9,9)=VPARSH(4,4)*(SCTSH/SCT)**2
      DO 9 KA=1,5 $ DO 9 KB=6,9 $ VPAR(KA,KB)=0
```

```
    9 VPAR(KB,KA)=0
      NPAR=9
C  THIS PROGRAM ASSUMES 5 PRESSURE FIELD PARAMETERS AND
C  4 SHOCK PARAMETERS
C
      NBAD=0
      IF(NHIST.GE.1) GOTO 12
11    NBAD=11  $  PRINT 14,NBAD
      PRINT 16 $  RETURN
16    FORMAT(1H+,45X,',BECAUSE NHIST=0')
   12 IF(O..LT.RMIN.AND.RMIN.LE.R.AND.R.LE.RMAX) GOTO 15
13    NBAD=13  $  PRINT 14,NBAD
      PRINT 17 $  RETURN
17    FORMAT(1H+,45X,',BECAUSE RMIN,RMAX,R ARE OUTSIDE RANGES')
   14 FORMAT(1H0,10X,29HRETURN FROM FLOFLD WITH NBAD=,I5)
C
   15 NHMAX=NHIST
      AIRPRSC=APRE/SCP
C  SCALED AIR PRESSURE IS NEEDED BY STRLIN
      RINNU=R $ NHIST=1
C  SET TO COMPUTE FIRST HISTORY NODE
   25 SOLIN(3)=RINNU
C
      CALL STRBEG(SOLIN,TPIN,XPP,UPP,UPTP,DPIN,LBAD)
C  THIS COMPUTES INITIAL POINT OF STREAMLINE
C
C  SOLIN(6)  = FLOW VARIABLES  (T,P,R,U,RHO,0.5*U**2*RHO)
C  TPIN(10)  =D/DPAR OF INITIAL TIME SOLIN(1)
C  XPP(10)   =D/DPAR OF THE INITIAL POSITION SOLIN(3)
C  UPP(10)   =D/DPAR OF THE INITIAL PARTICLE VELOCITY SOLIN(4)
C  UPTP(10)  =D/DPAR OF THE INITIAL PARTICLE ACCELERATION
C  DPIN(10)  =AN EXPRESSION OF DERIVATIVES NEEDED BY STRLIN
C  LBAD      = ERROR INDICATOR.  LBAD.NE.0  IF ERROR RETURN FROM STRBEG
C
      IF(LBAD.EQ.0) GOTO 35
   34 NBAD=34 $ PRINT 14, NBAD $ RETURN
   35 TMAXS=AMAX1(TMAX,SOLIN(1))
      NSTRNU=200
      DTNU=SOLIN(1)/100.
C  DEFAULT DT FOR ONE-NODE STREAMLINE COMPUTATION
      NODES=MIN0(NSTRNU-1,MAX0(NHMAX,20))
      IF(TMAXS.GT.SOLIN(1))DTNU=(TMAXS-SOLIN(1))/FLOAT(NODES-1)
C
      CALL STRLIN(TMAXS,AIRPRSC,AGAM,PFIELD,PAR,VPAR,NPAR,SOLIN,
     A TPIN,XPP,UPP,UPTP,DPIN,DTNU,STRNU,VSTRNU,NSTRNU,LBAD)
C
C  THIS COMPUTES A STREAMLINE STARTING AT SOLIN AND ENDING AT TMAXS
C
C  TMAXS          = END POINT OF STREAMLINE
C  AIRPRSC        = AIR PRESSURE
C  AGAM           = GAMMA OF AIR
C  PFIELD         = PRESSURE FIELD FUNCTION
C  PAR,VPAR,NPAR = PRESSURE FIELD PARAMETERS, VARIANCES, NUMBER
C
C  SOLIN THROUGH DPIN ARE PASSED FROM STRBEG
C
C  DTNU             = DELTA-TIME TO BE USED FOR INTEGRATION
```

```
      C  STRNU(6,200)     = STREAMLINE FLOW VARIABLES (T,P,R,U,RHO,DP)
      C  VSTRNU(6,6,200) = VARIANCE-COVARIANCE MATRICES OF STRNU
      C  NSTRNU           = NUMBER OF NODES IN STRNU. INITIALLY IT SHOULD
      C                     BE SET EQUAL TO THE MAXIMUM DESIRED
      C  LBAD             = ERROR INDICATOR.  LBAD.NE.0  IF ERROR RETURN
      C
            IF(LBAD.EQ.0) GOTO 48
         46 NBAD=46 $ PRINT 14, NBAD $ RETURN
         48 IF(NHIST.GT.1) GOTO 65
      C  BRANCH AFTER FIRST NODE. ELSE DELTR MAY BE ESTABLISHED
            IF(TMAX.LE.STRNU(1,1)) GOTO 55
            IF(NHMAX.EQ.1) GOTO 55
      C  BRANCH IF THIS WAS A ONE-NODE CALCULATION
            RHOZSC=(AMOL/8.3143)*(APRE/ATEM)*(SCD/SCT)**2/SCP
            DTHIST=(TMAX-STRNU(1,1))/FLOAT(NHMAX-1)
      C  THIS IS DELTA-TIME FOR HISTORY
            DELTR=DTHIST*STRNU(4,1)*STRNU(2,1)/(STRNU(2,1)-RHOZSC*STRNU(4,1)*
           12)
      C  DISTANCE DECREMENT FOR SUBSEQUENT STREAMLINES
      C  THE SECOND STREAMLINE WILL CROSS R AT ABOUT STRNU(1,1)+DTHIST
      C
      C  NOW STORE CALCULATED FIRST NODE
         55 DO 57 KA=1,5 $ DO 56 KB=1,5
            KC=KA $ IF(KA.GT.2)KC=KA+1
            KD=KB $ IF(KB.GT.2)KD=KB+1
         56 VHIST(KA,KB,1)=VSTRNU(KC,KD,1)
         57 HIST(KA,1)=STRNU(KC,1)
            IF(NHMAX.EQ.1.OR.TMAX.LE.HIST(1,1)) GOTO 145
      C  RETURN IN ONE-NODE HISTORY CASE
      C
            RINOL=RINNU $ RINNU=RINOL-DELTR
            DRSIGN=1.
            GOTO 100
      C  BRANCH TO STORING OF STRNU IN STROL AND NEXT STREAMLINE
      C
         65 TIME=HIST(1,NHIST-1)+DTHIST
            TIME=AMIN1(TIME,TMAX)
      C  ENTER 65 FROM 48. NOW STROL CONTAINS DATA.
      C  ALSO LOOP TO 65 FROM 88
      C
            CALL FLINTER(TIME,R,HIST,VHIST,NHIST,STROL,VSTROL,NSTROL,
           1 STRNU,VSTRNU,NSTRNU,DRSIGN,KBAD)
      C  THIS INTERPOLATED BETWEEN STROL AND STRNU AND STORED
      C  RESULTS IN HIST(...,NHIST).
      C
         73 IF(KBAD.NE.99) GOTO 75
            NHIST=NHIST-1
            GOTO 95
      C  BRANCH TO CALCULATION OF NEXT STREAMLINE INSTEAD OF USING EXTRAPOLATED
      C  VALUE
         75 IF(KBAD.EQ.0) GOTO 85
         77 NBAD=77 $ PRINT 14, NBAD $ RETURN
         85 IF(HIST(1,NHIST).GE.TMAX-DTHIST*0.1) GOTO 145
      C  THIS IS REGULAR RETURN AFTER REACHING TMAX
      C
            NHIST=NHIST+1
         88 GOTO 65
```

```
C
   95 RINOL =RINNU $ RINNU=RINOL -DELTR*DRSIGN
C  ENTER 95 FROM 73 AND GET NEXT STREAMLINE
  100 RINNU=AMAX1(RINNU,RMIN) $ RINNU=AMIN1(RINNU,RMAX)
      IF(RINNU.NE.RINOL ) GOTO 115
      MESS=1  $  GOTO 155
  105 FORMAT(1H0,10X,5HTMAX=,1PE12.5,19H  CANNOT BE REACHED,
     A33H BECAUSE OF RESTRICTIONS BY RMIN=,1PE12.5,11H  AND RMAX=,
     B 1PE12.5,/)
C
  115 DO 125 KA=1,NSTRNU
C  NOW STORE OLD STREAMLINE
      DO 122 KB=1,6 $ DO 120 KC=1,6
  120 VSTROL(KC,KB,KA)=VSTRNU(KC,KB,KA)
  122 STROL(KB,KA)=STRNU(KB,KA)
  125 CONTINUE
      NSTROL=NSTRNU
      NHIST=NHIST+1
      GOTO 25
C
  145 MESS=0
C  ENTER 145 FROM 85 FOR REGULAR RETURN
  155 CALL PRIHIS(R,HIST,VHIST,NHIST)
      IF(MESS.EQ.1)PRINT 105,TMAX,RMIN,RMAX
      CALL UTEST(SCD,SCP,SCT,RMIN,RMAX,R,TMAX,PAR,VPAR,NPAR,
     A HIST,VHIST,NHIST,UTST,NUTST,LBAD)
      CALL PRITST(R,RMAX,HIST,VHIST,NHIST,UTST,NUTST)
      IF(LBAD.NE.0)PRINT 165,LBAD
  165 FORMAT(1H0,10X,12HLBAD(UTEST)=,I5)
      RETURN
      END
```

```
      SUBROUTINE STRBEG(SOLIN,TPIN,XPP,UPP,UPTP,DPIN,NBAD)
C
C  THIS COMPUTES THE INITIAL STREAMLINE NODE ON THE SHOCK AND  ITS
C  ACCURACY. THE SOLIN COMPONENTS ARE
C       (T, P, R, U, RHO, U**2*RHO/2)
C  THE GIVEN ARGUMENT IS THE SHOCK DISTANCE R=SOLIN(3).
C  R IS ASSUMED TO BE CONSISTENT WITH THE SCALES IN /CSCALE/
C  TPIN,XPP,UPP,UPTP AND DPIN ARE INITIAL STREAMLINE VARIABLE
C  DERIVATIVES WITH RESPECT TO THE PARAMETERS
C
C  ROUTINR USES F2SHCK
C
      DIMENSION SOLIN(6),TPIN(10),XPP(10),UPP(10),UPTP(10),DPIN(10)
      DIMENSION X(5,1),PAR(10),FX(5),FP(10),FXX(5,5),FXP(5,10),
     A FPP(10,10),SOLMAT(6,4),SCALE(4)
C
      COMMON/CSCALE/SCD,SCP,SCT
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCDC,SCPC,SCTC
      COMMON/AMBCHA/PZ,TZ,GZ,AMZ,VCH,ENCH,HCH,EHCH
      COMMON/COMSHK/NPS,PARS(4),VPARS(4,4),SCDS,SCPS,SCTS
C
      DO  25 KA=1,3
25    SCALE(KA)=(SCPS/SCP)*(SCDS/SCD)**KA
      SCALE(4)=SCTS/SCT
      DO 45 KA=1,4       $  PAR(KA)=SCALE(KA)*PARS(KA)
45    CPAR(KA)=PAR(KA)
C  THE NEW PARAMETERS ARE SCALED ACCORDING TO /CSCALE/
C
      SNDSPD=SNDSPD*(SCT/SCTC)*(SCDC/SCD)
      GAMCAP=GAMCAP*(SCP/SCPC)
      ALOW=ALOW*(SCDC/SCD)
      SCDC=SCD  $  SCPC=SCP  $  SCTC=SCT
C  THIS TRANSFORMED /CF2DER/ INTO /CSCALE/ UNITS
C
      R=SOLIN(3)
C  NEXT COMPUTE SHOCK ARRIVAL TIME
      X(1,1)=0.  $  X(2,1)=R  $  X(3,1)=0.
      CALL F2SHCK(X,1,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
      IF(NBAD.NE.0) RETURN
C
      POV=((PAR(3)/R+PAR(2))/R+PAR(1))/R
      USH=SNDSPD*SQRT(1.+GAMCAP*POV)
C  SHOCK VELOCITY
      ROSI=(AMZ/8.3143)*(PZ/TZ)
C  ROSI IS AMBIENT DENSITY IN SI UNITS
      RAMB=ROSI*(SCD/SCT)**2/SCP
C  AMBIENT DENSITY IN /CSCALE/ UNITS
C
      UPSH=POV/(RAMB*USH)
C  PARTICLE VELOCITY AT THE SHOCK
      GAMTIL=((GZ-1.)/(2.*GZ*PZ))*SCP
      ROSH=RAMB*(1.+GAMCAP*POV)/(1.+GAMTIL*POV)
C  DENSITY AT THE SHOCK
      DPSH=UPSH**2*ROSH*0.5
C  DYNAMIC PRESSURE AT THE SHOCK (=SPECIFIC KINETIC ENERGY)
      SOLIN(1)=F/SNDSPD
      SOLIN(2)=POV
```

```
      SOLIN(4)=UPSH
      SOLIN(5)=ROSH
      SOLIN(6)=DPSH
C
C  NEXT COMPUTE INFLUENCE MATRIX SOLMAT WHICH EXPRESSES THE
C  RELATION BETWEEN SOLIN AND THE PARAMETER VARIANCES VPARS
      DUM=1.+GAMCAP*POV
      UPFACT=UPSH*(1./POV-0.5*GAMCAP/DUM)
      ROFACT=1./(SNDSPD**2*DUM*(1.+GAMTIL*POV))
      DPFACT=(UPSH**2*ROFACT+2.*UPSH*ROSH*UPFACT)*0.5
      DO 65 KA=1,3
65    SOLMAT(2,KA)=1./R**KA
      SOLMAT(2,4)=0.
      DO 75 KA=1,4
      SOLMAT(1,KA)=FP(KA)/SNDSPD
      SOLMAT(3,KA)=0.
      SOLMAT(4,KA)=UPFACT*SOLMAT(2,KA)
      SOLMAT(5,KA)=ROFACT*SOLMAT(2,KA)
75    SOLMAT(6,KA)=DPFACT*SOLMAT(2,KA)
C
      DO 105 KA=1,10 $ XPP(KA)=0 $ UPP(KA)=0 $ TPIN(KA)=0 $ DPIN(KA)=0
105   UPTP(KA)=0
      POVR=-((3.*PAR(3)/R+2.*PAR(2))/R+PAR(1))/R**2
      UPT=-POVR/ROSH
C  DU/DT OF PARTICLE VELOCITY AT SHOCK
      DO 115 KA=1,3
      TPIN(KA+5)=SOLMAT(1,KA)
      DPIN(KA+5)=(ROFACT/ROSH-1./(GZ*(POV+PZ/SCP)))*SOLMAT(2,KA)
      UPP(KA+5)=SOLMAT(4,KA)
115   UPTP(KA+5)=UPT*(-SOLMAT(5,KA)/ROSH+FLOAT(-KA)/(R**(KA+1)*POVR))
      TPIN(9)=SOLMAT(1,4)
      RETURN
      END
```

```
      SUBROUTINE STRLIN(TMAX,AIRPR,AIRGAM,PFIELD,PAR,VPAR,NPAR,SOLIN,
     A TPIN,XPP,UPP,UPTP,DPIN,DT,SLINA,VSLINA,NMAXA,NBAD)
C  THIS COMPUTES A STREAMLINE STARTING WITH SPECIFIED INITIAL
C  VALUES AND ENDING AT TMAX
C
C  TMAX              = TIME AT END POINT OF STREAMLINE. THE ACTUAL TIME
C                      CAN BE BY DT LARGER THAN TMAX
C  AIRPR             = AMBIENT PRESSURE
C  AIRGAM            = RATIO OF SPECIFIC HEATS
C  PFIELD            = PRESSURE FIELD SUBROUTINE
C  PAR,VPAR,NPAR     = PARAMETERS, THEIR VARIANCE AND NUMBER FOR PFIELD
C  SOLIN(6)          = INITIAL VALUES ON STREAMLINE, VIZ.
C                      TIME, PRESSURE, DISTANCE, VELOCITY, DENSITY,
C                      DYNAMIC PRESSURE (= KINETIC ENERGY DENSITY)
C  TPIN(10)          = D/DPAR OF THE INITIAL TIME
C  XPP(10)           = D/DPAR OF INITIAL POSITION
C  UPP(10)           = D/DPAR OF INITIAL PARTICLE VELOCITY
C  UPTP(10)          = D/DPAR OF INITIALL PARTICLE ACCELERATION
C  DPIN(10)          = D/DPAR EXPRESSION NEEEDED FOR INTEGRATION OF UPP
C  DT                = TIME INTERVAL FOR INTEGRATION
C
C  THE FOLLOWING WILL BE COMPUTED BY THIS ROUTINE
C
C  SLINA(6,NMAXA)    = FLOW VARIABLES ALONG THE STREAMLINE (T,P,R,U,RHO,U
C  VSLINA(6,6,NMAXA)= VARIANCE-COVARIANCE MATRIX OF SLINA
C  NMAXA             = MAXIMUM NUMBER OF NODES IN SLINE
C                      WILL BE REPLACED BY ACTUAL NUMBER COMPUTED
C  NBAD              = ERROR INDICATOR
C
      DIMENSION PAR(10),VPAR(10,10),SOLIN(6),TPIN(10),XPP(10),UPP(10),
     A UPTP(10),DPIN(10),SLINA(6,100),VSLINA(6,6,100)
C
      DIMENSION X(3,1),FX(3),FP(10),FXP(3,10),FXX(3,3),FPP(10,10)
C
      DIMENSION UT(2),XP(2,10),UTP(2,10),UP(2,10),SOLMAT(6,10)
     A,U(2),UTT(2),SLINE(6,51),VSLINE(6,6,51)
C  SLINE AND VSLINE ARE WORKING AREAS WITH LENGTH NMAX
      DATA (NMAX=51)
C
      NBAD=0
      DO 9 KA=1,6
      SLINE(KA,1)=SOLIN(KA)
    9 SLINA(KA,1)=SOLIN(KA)
      IF(NMAXA.GT.2)GOTO 12
      NBAD=11 $ PRINT 11, NBAD $ RETURN
   11 FORMAT(1H0,10X,30HRETURN FROM STRLIN WITH NBAD =,I4)
   12 IF(DT.GT.0.) GOTO 15
      IF(SLINA(1,1).GE.TMAX) GOTO 15
      NBAD=12 $ PRINT 11,NBAD $ RETURN
C  DT IS PERMITTED TO BE ZERO FOR ONE POINT STREAMLINE
   15 IF(SOLIN(3).GT.0.) GOTO 25
C  CHECK FOR NEGATIVE INITIAL DISTANCE
      NBAD=15 $ PRINT 11, NBAD $ RETURN
   25 CONTINUE
      ROZ=SOLIN(5) $ GEXP=1./AIRGAM $ PRZ=SOLIN(2)+AIRPR
      DO 31 I=1,2
      DO 30 KA=1,NPAR $ XP(I,KA)=XPP(KA) $ UP(I,KA)=UPP(KA)
```

```
   30 UTP(I,KA)=UPTP(KA)
   31 CONTINUE
C
      X(1,1)=SLINE(1,1) $ X(2,1)=0. $ X(3,1)=SLINE(3,1)
C     TIME          PRESSURE          DISTANCE
      CALL PFIELD(X,1,PAR,F,FX,FP,FXX,FXP,FPP,LBAD)
 3500 IF(LBAD.EQ.0) GOTO 39
      NBAD=3500+LBAD $ PRINT 11, NBAD $ RETURN
C
   39 UT(1)=-FX(3)*(PRZ/(F+AIRPR))**GEXP/ROZ
C  DU/DT=-(DP/DR)*(PO/P)**(1/GAMMA)/RHOZERO
      U(1)=SLINE(4,1)
      UTT(1)=UT(1)*(-GEXP*(FX(1)+U(1)*FX(3))/(F+AIRPR)
     A +(FXX(1,3)+U(1)*FXX(3,3))/FX(3) )
      DTSTOR=DT $ TSTOR=SLINA(1,1)+DTSTOR $ KT=1
C  COMPUTATION RESULTS WILL BE STORED APROXIMATELY FOR TSTOR
C  KT COUNTS STORAGE IN SLINA AND VSLINA
C  THIS IS ACTUAL INTEGRATION INTERVAL. WITH DTS=0 GET FIRST NODE
      DTS=0.
      KA=1
C
C     NEXT STATEMENT IS BEGINNING OF INTEGRATION LOOP
   45 SLINE(3,KA+1)=SLINE(3,KA)+DTS*(U(1)+0.5*DTS*(UT(1)+DTS*UTT(1)/3.)
C  NEW DISTANCE BY FOURTH ORDER FORMULA IN DTS
      SLINE(1,KA+1)=SLINE(1,KA)+DTS
C  NEW TIME
      DO 47 KB=1,NPAR
   47 XP(2,KB)=XP(1,KB)+DTS*(UP(1,KB)+0.5*DTS*UTP(1,KB))
C  NEW DX/DPARAMETER. THIRD ORDER ERROR IN DTS
C
      X(1,1)=SLINE(1,KA+1) $ X(2,1)=0 $ X(3,1)=SLINE(3,KA+1)
      CALL PFIELD(X,1,PAR,F,FX,FP,FXX,FXP,FPP,LBAD)
      IF(LBAD.EQ.0) GOTO 55
 5100 NBAD=5100+LBAD $ PRINT 11, NBAD $ RETURN
C
   55 SLINE(2,KA+1)=F
C  NEW PRESSURE
      UT(2)=-FX(3)*(PRZ/(F+AIRPR))**GEXP/ROZ
      U(2)=U(1)+0.5*DTS*(UT(1)+UT(2))
C  FIRST APPROXIMATION OF NEW VELOCITY. THIRD ORDER ERROR IN DTS
      UTT(2)=UT(2)*(-GEXP*(FX(1)+U(2)*FX(3))/(F+AIRPR)
     A +(FXX(1,3)+U(2)*FXX(3,3))/FX(3))
      U(2)=U(2)+(UTT(1)-UTT(2))*DTS**2/12.
C  NEW VELOCITY. FIFTH ORDER ERROR IN DTS
      SLINE(4,KA+1)=U(2)
      DO 65 KB=1,NPAR
      UTP(2,KB)=UT(2)*(-DPIN(KB)
     A -(FP(KB)+FX(3)*XP(2,KB))*GEXP/(F+AIRPR)
     B +(FXP(3,KB)+FXX(3,3)*XP(2,KB))/FX(3))
      UP(2,KB)=UP(1,KB)+0.5*DTS*(UTP(1,KB)+UTP(2,KB))
   65 CONTINUE
C  NEW DU/DPARAMETER. THIRD ORDER ERROR IN DTS
      SLINE(5,KA+1)=ROZ*((F+AIRPR)/PRZ)**GEXP
C  NEW DENSITY
      SLINE(6,KA+1)=0.5*SLINE(5,KA+1)*SLINE(4,KA+1)**2
C  NEW DYNAMIC PRESSURE
C
```

```
C  NEXT COMPUTE VARIANCE ESTIMATES OF SOLUTION
      DO 75 KB=1,NPAR
      SOLMAT(1,KB)=TPIN(KB)
      SOLMAT(2,KB)=FP(KB)+FX(3)*XP(2,KB)
      SOLMAT(3,KB)=XP(2,KB)
      SOLMAT(4,KB)=UP(2,KB)
      SOLMAT(5,KB)=SLINE(5,KA+1)*(DPIN(KB)
     A +GEXP*(FP(KB)+FX(3)*XP(2,KB)+FX(1)*SOLMAT(1,KB))/(F+AIRPR))
      SOLMAT(6,KB)=0.5*SLINE(4,KA+1)*(SLINE(5,KA+1)*SOLMAT(4,KB)*2.
     A +SLINE(4,KA+1)*SOLMAT(5,KB))
   75 CONTINUE
C  SOLMAT IS THE JACOBIAN MATRIX DSLINE/DPARAMETER
      DO 95 KB=1,6 $ DO 95 KC=1,6
      VSLINE(KB,KC,KA+1)=0.
      DO 85 KD=1,NPAR  $  DO 85 KE=1,NPAR
      VSLINE(KB,KC,KA+1)=VSLINE(KB,KC,KA+1)+
     A SOLMAT(KB,KD)*VPAR(KD,KE)*SOLMAT(KC,KE)
   85 CONTINUE
   95 CONTINUE
C
C  NOW STORE RESULTS IF TSTOR REACHED
      KA=KA+1
      IF(KT.EQ.1)GOTO 97
      IF(SLINE(1,KA).LT.TSTOR-DTS*0.2)GOTO 125
   97 DO 99 KB=1,6 $ DO 98 KC=1,6
   98 VSLINA(KB,KC,KT)=VSLINE(KB,KC,KA)
   99 SLINA(KB,KT)=SLINE(KB,KA)
C
      IF(SLINA(1,KT).GE.TMAX)GOTO 155
C  BRANCH TO 155 WHEN END OF STREAMLINE REACHED
      TSTOR=SLINA(1,KT)+DTSTOR
C  TIME VALUE FOR NEXT NODE TO BE STORED IN SLINA
      KT=KT+1 $ DTS=DT*0.2
C  AFTER FIRST NODE CONTINUE WITH DTS.GT.0.
C
      IF(KT.LT.NMAXA)GOTO 115
C
C  THIS IS PROGRAMMING ERROR. WITH GIVEN DT END TIME CANNOT
C  BE REACHED IN NMAXA STEPS. CORRECT BY INCREASING DT
      DTSTOR=DTSTOR*2.
C  ELIMINATE HALF OF STORED RESULTS
      KC=2 $ KB=3
  102 DO 104 KD=1,6 $ DO 103 KE=1,6
  103 VSLINA(KD,KE,KC)=VSLINA(KD,KE,KB)
  104 SLINA(KD,KC)=SLINA(KD,KB)
      KC=KC+1 $ KB=KB+2
      IF(KB.LE.NMAXA)GOTO 102
      KT=KC-1 $ TSTOR=SLINA(1,KT)+DTSTOR
      GOTO 125
C
  115 IF(KT.LE.2)KA=1
C
  125 IF(KA.LT.NMAX)GOTO 145
C
C  NOW WORK AREA IS OVERFLOWING. ELIMINATE OLD STUFF
      KC=2 $ KB=3
  131 DO 133 KD=1,6 $ DO 132 KE=1,6
```

```
  132 VSLINE(KE,KD,KC)=VSLINE(KE,KD,KB)
  133 SLINE(KD,KC)=SLINE(KD,KB)
      KC=KC+1 $ KB=KB+2
      IF(KB.LE.NMAX)GOTO 131
      KA=KC-1 $ IF(KB.EQ.NMAX+1) GOTO 45
C
C  PREPARE FOR NEXT INTEGRATION STEP
  145 U(1)=U(2) $ UT(1)=UT(2) $ UTT(1)=UTT(2)
      DO 148 KB=1,NPAR $ XP(1,KB)=XP(2,KB) $ UP(1,KB)=UP(2,KB)
  148 UTP(1,KB)=UTP(2,KB)
      GOTO 45
C
  155 NMAXA=KT
      RETURN
      END
```

```
      SUBROUTINE FLINTER(T,R,HIST,VHIST,NHIST,STROLD,VSTROLD,
     A NOLD,STRNEW,VSTRNEW,NNEW,DRSIGN,NBAD)
C  FLOW FIELD INTERPOLATION BETWEEN TWO STREAMLINES.
C  INTERPOLATED RESULTS ARE STORED IN HIST AND VHIST.
C
C  T                  = TIME VALUE FOR INTERPOLATION
C  R                  = DISTANCE VALUE FOR INTERPOLATION
C  HIST(5,100)        = HISTORY VARIABLES = T,P,U,RHO,U**2*RHO/2
C  VHIST(5,5,100)     = VARIANCE-COVARIANCE MATRIX OF HIST
C  NHIST              = NODE NUMBER WHERE TO STORE RESULTS
C  STROLD(6,100)      = PREVIOUS STREAMLINE=T,P,R,U,RHO,U**2*RHO/2
C  VSTROLD(6,6,100)   = VARIANCE-COVARIANCE MATRIX OF STROLD
C  NOLD               = NUMBER OF NODES IN STROLD
C  STRNEW(6,100)      = NEW STREAMLINE=T,P,R,U,RHO,U**2*RHO/2
C  VSTRNEW(6,6,100)   = VARIANCE-COVARIANCE MATRIX OF STRNEW
C  NNEW               = NUMBER OF NODES IN STRNEW
C  DRSIGN             = SIGN OF NEXT DELTA-R TO BE SUBTRACTED
C                       FROM PREVIOUS INITIAL POINT OF STREAMLINE
C  NBAD               = ERROR INDICATOR. NBAD=99 MEANS THAT
C                       EXTRAPOLATION WOULD BE NECESSARY.
C
      DIMENSION HIST(5,100),VHIST(5,5,100),STROLD(6,100),VSTROLD(6,6,
     A),STRNEW(6,100),VSTRNEW(6,6,100)
      DIMENSION XA(6),VXA(6,6),XB(6),VXB(6,6),XZ(6),VXZ(6,6)
C
      NBAD=0
      IF( NHIST.GE.2)GOTO 15
      NBAD=14 $ PRINT 14,NBAD $ RETURN
   14 FORMAT(1H0,10X,31HRETURN FROM FLINTER WITH NBAD =,I4)
C  NO INTERPOLATION FOR FIRST NODE OF HIST
   15 IF(NOLD.GT.1)GOTO 17
      NBAD=15 $ PRINT 14,NBAD $ RETURN
   17 IF(NNEW.GT.1)GOTO 25
      NBAD=17 $ PRINT 14,NBAD $ RETURN
C
C  NOW FIND BASE WITH TIME=T ON OLD STREAMLINE
   25 DO 29 KA=1,NOLD
      IF(T-STROLD(1,KA))35,38,29
   29 CONTINUE
      NBAD=29 $ PRINT 14,NBAD $ RETURN
   35 IF(KA.GT.1)GOTO 45
      NBAD=35 $ PRINT 14,NBAD $ RETURN
   38 KA1=KA $ KA2=2
      FA1=1. $ FA2=0. $ GOTO 51
   45 KA1=KA-1 $ KA2=KA
      DEN=STROLD(1,KA2)-STROLD(1,KA1)
      FA1=(STROLD(1,KA2)-T)/DEN
      FA2=(T-STROLD(1,KA1))/DEN
   51 DO 55 KA=1,6 $ DO 53 KB=1,6
   53 VXA(KB,KA)=FA1*VSTROLD(KB,KA,KA1)+FA2*VSTROLD(KB,KA,KA2)
   55 XA(KA)=FA1*STROLD(KA,KA1)+FA2*STROLD(KA,KA2)
C
C  NOW FIND BASE WITH TIME=T ON NEW STREAMLINE
      DO 69 KA=1,NNEW
      IF(T-STRNEW(1,KA))75,78,69
   69 CONTINUE
      NBAD=69 $ PRINT 14,NBAD $ RETURN
```

```
   75 IF(KA.GT.1)GOTO 85
      NBAD=75 $ PRINT 14,NBAD $ RETURN
   78 KB1=KA $ KB2=2
      FB1=1. $ FB2=0. $ GOTO 91
   85 KB1=KA-1 $ KB2=KA
      DEN=STRNEW(1,KB2)-STRNEW(1,KB1)
      FB1=(STRNEW(1,KB2)-T)/DEN
      FB2=(T-STRNEW(1,KB1))/DEN
   91 DO 95 KA=1,6 $ DO 93 KB=1,6
   93 VXB(KB,KA)=FB1*VSTRNEW(KB,KA,KB1)+FB2*VSTRNEW(KB,KA,KB2)
   95 XB(KA)=FB1*STRNEW(KA,KB1)+FB2*STRNEW(KA,KB2)
C
C  NOW CHECK IF EXTRAPOLATION REQUIRED
      IF((XA(3)-R)*(XB(3)-R).LE.0.)GOTO 105
      DRSIGN=1. $ IF(XA(3)-R.LT.0.)DRSIGN=-1.
   99 NBAD=99
C  THIS INDICATES THAT THE NEW VALUE IS OBTAINED BY EXTRAPOLATION
C
      IF(XA(3)-XB(3).NE.0.)GOTO 105
  102 NBAD=102 $ PRINT 14,NBAD $ RETURN
C  NOW INTERPOLATE
  105 FA=(R-XB(3))/(XA(3)-XB(3))
      FB=(XA(3)-R)/(XA(3)-XB(3))
      DO 115 KA=1,6 $ DO 114 KB=1,6
  114 VXZ(KB,KA)=FA*VXA(KB,KA)+FB*VXB(KB,KA)
  115 XZ(KA)=FA*XA(KA)+FB*XB(KA)
C
C  NEXT STORE RESULTS IN HIST AND VHIST
      DO 125 KA=1,5 $ DO 124 KB=1,5
      KC=KA $ IF(KA.GT.2)KC=KA+1
      KD=KB $ IF(KB.GT.2)KD=KB+1
  124 VHIST(KA,KB,NHIST)=VXZ(KC,KD)
  125 HIST(KA,NHIST)=XZ(KC)
      RETURN
      END
```

```
      SUBROUTINE PFIELD(X,KK,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C
C  THIS IS PRESSURE FIELD CONSTRAINT SUBROUTINE.
C  THE FUNCTION F IS DEFINED AS
C     F=(PSHOCK-C)K*EXP(Q(T,R,P(1),...,P(4))+C(R,P(5)) - P
C  THE OBSERVABLES ARE
C     TIME T=X(1), OVERPRESSURE P=X(2), RADIUS R=X(3)
C  THE FUNCTIONS Q,PSHOCK,C WILL BE OBTAINED BY CALLING
C  QFUNCT AND CCOEF.
C
      DIMENSION X(3,1),PAR(10),FX(3),FP(10),FXX(3,3),FXP(3,10),FPP(10,
     1)
      DIMENSION QX(3),QP(10),QXX(3,3),QXP(3,10),QPP(10,10),CX(3),
     ACP(10),CXX(3,3),CXP(3,10),CPP(10,10),PSP(10),PSRP(10),PSPP(10,10
      DIMENSION PSCX(3),PSCP(10)
C
      NPSHK=4  $  GOTO 10
      ENTRY PFIELDC
      NPSHK=0
10    CONTINUE
C
C  ENTRY PFIELDC IS USED AS CONSTRAINT FOR PRESSURE FIELD ADJUSTMENT
C  IT DOES NOT COMPUTE DERIVATIVES WITH RESPECT TO THE SHOCK
C  PARAMETERS PAR(6) THROUGH PAR(9)
C
C  ENTRY PFIELD IS USED TO COMPUTE THE PRESSURE FIELD AFTER ADJUSTMENT
C  IT COMPUTES DERIVATIVES OF THE OVERPRESSURE WITH RESPECT TO
C  ALL PARAMETERS
C
      DO 12 KB=1,10
      FXP(1,KB)=0  $  FXP(2,KB)=0  $  FXP(3,KB)=0   $ FP(KB)=0
      DO 12 KC=1,10
12    FPP(KC,KB)=0
      NBAD=0
      CALL QFUNCT(X,KK,PAR,Q,QX,QP,QXX,QXP,QPP,
     APS,PSR,PSP,PSRR,PSRP,PSPP,NPSHK,NBAD)
      IF(NBAD.NE.0)RETURN
      CALL CCOEF(X,KK,PAR,C,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.NE.0)RETURN
C
      PSC=PS-C
13    IF(Q.LT.740.) GOTO 14  $  NBAD=740  $  RETURN
14    EXPQ=0.  $  IF(Q.GT.-670.) EXPQ=EXP(Q)
C  STATEMENTS 13 AND 14 AVOID OVERFLOW OR UNDERFLOW BY EXP FUNCTION
      FEX=PSC*EXPQ
      F=FEX+C-X(2,KK)
      DO 15 KB=1,3
      PSCX(KB)=-CX(KB)
15    FX(KB)=EXPQ*(PSC*QX(KB)+PSCX(KB))+CX(KB)
      FX(2)=FX(2)-1.
      PSCX(3)=PSCX(3)+PSR
      FX(3)=FX(3)+EXPQ*PSR
      DO 25 KB=1,5
      PSCP(KB)=-CP(KB)
25    FP(KB)=EXPQ*(PSC*QP(KB)+PSCP(KB))+CP(KB)
C
      DO 35 KB=1,3 $ DO 35 KC=1,5
```

```
      FXP(KB,KC)=EXPQ*(PSC*(QXP(KB,KC)+QX(KB)*QP(KC))
     A+QX(KB)*PSCP(KC)+PSCX(KB)*QP(KC)-CXP(KB,KC))+CXP(KB,KC)
   35 CONTINUE
C
      DO 32 KB=1,3 $ DO 32 KC=1,3
      FXX(KB,KC)=EXPQ*(PSC*(QXX(KB,KC)+QX(KB)*QX(KC))
     A+QX(KB)*PSCX(KC)+PSCX(KB)*QX(KC)-CXX(KB,KC))+CXX(KB,KC)
   32 CONTINUE
      FXX(3,3)=FXX(3,3)+EXPQ*PSRR
C
      DO 45 KB=1,5 $ DO 45 KC=1,5
      FPP(KB,KC)=EXPQ*(PSC*(QPP(KB,KC)+QP(KB)*QP(KC))
     A+QP(KB)*PSCP(KC)+PSCP(KB)*QP(KC)-CPP(KB,KC))+CPP(KB,KC)
45    CONTINUE
C
      IF(NPSHK.LE.0)GOTO 75
C  NPSHK IS THE NUMBER OF SHOCK PARAMETERS.  NPSHK=0  OR  =4
      KUP=5+4
C  ASSUME THAT PRESSURE FUNCTION HAS 5 PARAMETERS AND SHOCK HAS 4 PAR.
      DO 55 KB=6,KUP
      PSCP(KB)=PSP(KB)
      FP(KB)=EXPQ*(PSC*QP(KB)+PSCP(KB))
      DO 52 KC=1,3
      FXP(KC,KB)=EXPQ*(PSC*(QXP(KC,KB)+QX(KC)*QP(KB))
     A+QX(KC)*PSCP(KB)+PSCX(KC)*QP(KB))
52    CONTINUE
      FXP(3,KB)=FXP(3,KB)+EXPQ*PSRP(KB)
      DO 55 KC=6,KUP
      FPP(KB,KC)=EXPQ*(PSC*(QPP(KB,KC)+QP(KB)*QP(KC))
     A+QP(KB)*PSCP(KC)+PSCP(KB)*QP(KC)+PSPP(KB,KC))
   55 CONTINUE
      DO 65 KB=1,5  $  DO 65 KC=6,KUP
      FPP(KB,KC)=EXPQ*(PSC*(QPP(KB,KC)+QP(KB)*QP(KC))
     A+QP(KB)*PSCP(KC)+PSCP(KB)*QP(KC)+PSPP(KB,KC))
65    FPP(KC,KB)=FPP(KB,KC)
   75 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QFUNCT(X,KK,PAR,Q,QX,QP,QXX,QXP,QPP,
     APS,PSR,PSP,PSRR,PSRP,PSPP,NPSHK,NBAD)
C  AUXILIARY ROUTINE FOR PFIELD. IT COMPUTES THE EXPONENT Q OF THE
C  PRESSURE FIELD FUNCTION. IT ALSO TRANSMITS THE SHOCK
C  OVERPRESSURE PS(R) WITH DERIVATIVES.
C
C     SUBROUTINES ACOEF,BCOEF AND SHODER ARE NEEDED
C
      DIMENSION X(3,1),PAR(10),QX(3),QP(10),QXX(3,3),QXP(3,10),
     A QPP(10,10),AX(3),AP(10),AXX(3,3),AXP(3,10),APP(10,10),
     B TAUX(3)
      DIMENSION TP(10),TRP(10),TPP(10,10),PSP(10),PSRP(10),PSPP(10,10)
C
      COMMON/CSCALE/SCDIS,SCPRE,SCTIM
      COMMON/COMSHK/NPSH ,PARSH(4),VPARSH(4,4),SCDSH,SCPSH,SCTSH
C
      DO 12 KA=1,10 $ QP(KA)=0 $ DO 10 KB=1,3
   10 QXP(KB,KA)=0 $ DO 12 KC=1,10
   12 QPP(KA,KC)=0
      NBAD=0 $ R=X(3,KK)*SCDIS
C
      IF(NPSHK.GT.0) GOTO 13
C  IF NPSHK = NUMBER OF SHOCK PARAMETERS IS ZERO THEN COMPUTE ONLY
C  DERIVATIVES WITH RESPECT TO PRESSURE PARAMETERS PAR(1) THROUGH PAR(5
      CALL SHOCK2(R,T,TR,TRR,PS,PSR,PSRR,NBAD)
      IF(NBAD.NE.0) RETURN
      GOTO 14
C
13    CONTINUE
      CALL SHODER(R,T,TR,TP,TRR,TRP,TPP,PS,PSR,PSP,
     A PSRR,PSRP,PSPP,NBAD)
      IF(NBAD.NE.0)RETURN
C
14     CONTINUE
C  SHOCK2 OR SHODER COMPUTED EVRYTHING IN SI UNITS. NOW SCALE RESULTS
C  ACCORDING TO THE SCALES IN  /CSCALE/
      T=T/SCTIM  $  TR=TR*SCDIS/SCTIM  $  TRR=TRR*SCDIS**2/SCTIM
      PS=PS/SCPRE  $  PSR=PSR*SCDIS/SCPRE  $ PSRR=PSRR*SCDIS**2/SCPRE
      IF(NPSHK.LE.0) GOTO 16
C
      DO 15 KB=6,8
      TP(KB)=TP(KB)*SCPRE*SCDIS**(KB-5)/SCTIM
      PSP(KB)=PSP(KB)*SCDIS**(KB-5)
      TRP(KB)=TRP(KB)*SCDIS**(KB-4)*SCPRE/SCTIM
      PSRP(KB)=PSRP(KB)*SCDIS**(KB-4)
      TPP(9,KB)=TPP(9,KB)*SCPRE*SCDIS**(KB-5)  $  TPP(KB,9)=TPP(9,KB)
      PSPP(9,KB)=PSPP(9,KB)*SCTIM*SCDIS**(KB-5)  $ PSPP(KB,9)=PSPP(9,KB
      DO 15 KC=6,8
      TPP(KC,KB)=TPP(KC,KB)*(SCPRE/SCTIM)**2*SCDIS**(KB+KC-10)
      PSPP(KC,KB)=PSPP(KC,KB)*SCDIS**(KB+KC-10)
15    CONTINUE
      PSP(9)=PSP(9)*SCTIM/SCPRE
      TPP(9,9)=TPP(9,9)*SCTIM
      PSPP(9,9)=PSPP(9,9)*(SCTIM/SCPRE)**2
C
16    CONTINUE
      TAU=X(1,KK)-T
```

```
      TAUX(1)=1. $ TAUX(2)=0. $ TAUX(3)=-TR
C
C  NEXT COMPUTE THE LINEAR TERM IN THE EXPONENT
      CALL ACOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
      IF(NBAD.NE.0)RETURN
      Q=A*TAU
C
      DO 25 KB=1,3
      QX(KB)=AX(KB)*TAU+A*TAUX(KB)
      DO 25 KC=1,3
      QXX(KB,KC)=AXX(KB,KC)*TAU+AX(KB)*TAUX(KC)+AX(KC)*TAUX(KB)
   25 CONTINUE
      QXX(3,3)=QXX(3,3)-A*TRR
C
      DO 35 KB=1,3 $ DO 35 KC=1,5
   35 QXP(KB,KC)=AXP(KB,KC)*TAU+AP(KC)*TAUX(KB)
C
      DO 45 KB=1,5 $ QP(KB)=AP(KB)*TAU
      DO 45 KC=1,5
   45 QPP(KB,KC)=APP(KB,KC)*TAU
      IF(NPSHK.LE.0)GOTO 53
C  NPSHK IS THE NUMBER OF SHOCK PARAMETERS
      KUP=5+NPSHK
C  ASSUME THAT PRESSURE FIELD HAS 5 PARAMETERS
      DO 48 KA=6,KUP
      QP(KA)=-A*TP(KA)
      QXP(3,KA)=-AX(3)*TP(KA)-A*TRP(KA)
      DO 48 KB=6,KUP
   48 QPP(KA,KB)=-A*TPP(KA,KB)
      DO 50 KA=1,5 $ DO 50 KB=6,KUP
      QPP(KA,KB)=-AP(KA)*TP(KB)
   50 QPP(KB,KA)=QPP(KA,KB)
C
C  NEXT COMPUTE QUADRATIC TERM
53    CALL BCOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
      IF(NBAD.NE.0)RETURN
      Q=Q+A*TAU*TAU
C
      DO 55 KB=1,3
      QX(KB)=QX(KB)+TAU*(AX(KB)*TAU+2.*A*TAUX(KB))
      DO 55 KC=1,3
      QXX(KB,KC)=QXX(KB,KC)+TAU*(AXX(KB,KC)*TAU+2.*AX(KB)*TAUX(KC)
     A+2.*AX(KC)*TAUX(KB))+2.*A*TAUX(KB)*TAUX(KC)
   55 CONTINUE
      QXX(3,3)=QXX(3,3)-2.*A*TAU*TRR
C
      DO 65 KB=1,3 $ DO 65 KC=1,5
      QXP(KB,KC)=QXP(KB,KC)+TAU*(AXP(KB,KC)*TAU+2.*
     ATAUX(KB)*AP(KC))
   65 CONTINUE
C
      DO 75 KB=1,5 $ QP(KB)=QP(KB)+AP(KB)*TAU*TAU
      DO 75 KC=1,5
   75 QPP(KB,KC)=QPP(KB,KC)+APP(KB,KC)*TAU*TAU
      IF(NPSHK.LE.0)GOTO 97
      DO 85 KA=6,KUP
      QP(KA)=QP(KA)-A*2.*TAU*TP(KA)
```

```
      QXP(3,KA)=QXP(3,KA)+2.*(-AX(3)*TAU*TP(KA)+A*TP(KA)*TR
     A-A*TAU*TRP(KA))
      DO 85 KB=6,KUP
      QPP(KA,KB)=QPP(KA,KB)+A*2.*(TP(KA)*TP(KB)-TAU*TPP(KA,KB))
   85 CONTINUE
      DO 95 KA=6,KUP $ DO 95 KB=1,5
      QPP(KB,KA)=QPP(KB,KA)-2.*AP(KB)*TP(KA)*TAU
   95 QPP(KA,KB)=QPP(KB,KA)
   97 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ACOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
C  LINEAR COEFFICIENT IN PRESSURE FIELD EXPONENT
C  AUXILIARY ROUTINE FOR QFUNCT
C
      DIMENSION X(3,1),PAR(10),AX(3),AP(10),AXX(3,3),AXP(3,10),
     AAPP(10,10),CP(2),CXP(2),CPP(2,2)
      COMMON/CFLDEX/EXA,EXB,EXC
C
      NBAD=0
      R=X(3,KK) $ P1=PAR(1) $   P2=PAR(2)
      EX=EXA
      CALL COEFFI(R,P1,P2,EX,A,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.EQ.0)GOTO 15 $ NBAD=NBAD+100 $ RETURN
C
   15 DO 25 KA=1,5 $ AP(KA)=0 $ IF(KA.LE.3)AX(KA)=0
      DO 25 KB=1,5 $ IF(KA.LE.3)AXP(KA,KB)=0
      IF(KA.LE.3.AND.KB.LE.3)AXX(KA,KB)=0
   25 APP(KA,KB)=0
C
      AX(3)=CX $ AP(1)=CP(1) $ AP(2)=CP(2)
      AXX(3,3)=CXX $ AXP(3,1)=CXP(1) $ AXP(3,2)=CXP(2)
      DO 35 KA=1,2 $ DO 35 KB=1,2
   35 APP(KA,KB)=CPP(KA,KB)
      RETURN $ END
```

```
      SUBROUTINE BCOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
C  QUADRATIC COEFFICIENT IN PRESSURE FIELD EXPONENT
C  AUXILIARY ROUTINE FOR QFUNCT
C
      DIMENSION X(3,1),PAR(10),AX(3),AP(10),AXX(3,3),
     AAXP(3,10),APP(10,10),CP(2),CXP(2),CPP(2,2)
      COMMON/CFLDEX/EXA,EXB,EXC
C
      NBAD=0
      R=X(3,KK) $ P1=PAR(3) $   P2=PAR(4)
      EX=EXB
      CALL COEFFI(R,P1,P2,EX,A,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.EQ.0)GOTO 15 $ NBAD=200+NBAD $ RETURN
C
   15 DO 25 KA=1,5 $ AP(KA)=0 $ IF(KA.LE.3)AX(KA)=0
      DO 25 KB=1,5 $ IF(KA.LE.3)AXP(KA,KB)=0
      IF(KA.LE.3.AND.KB.LE.3)AXX(KA,KB)=0
   25 APP(KA,KB)=0
C
      AX(3)=CX $ AP(3)=CP(1) $ AP(4)=CP(2)
      AXX(3,3)=CXX $ AXP(3,3)=CXP(1) $ AXP(3,4)=CXP(2)
      DO 35 KA=1,2 $ DO 35 KB=1,2
   35 APP(2+KA,2+KB)=CPP(KA,KB)
      RETURN $ END
```

```
      SUBROUTINE CCOEF(X,KK,PAR,A,AX,AP,AXX,AXP,APP,NBAD)
C  THIS IS ADDITIVE COEFFICIENT IN PRESSURE FIELD FORMULA
C  AUXILIARY ROUTINE FOR PFIELD
C
      DIMENSION X(3,1),PAR(10),AX(3),AP(10),AXX(3,3),AXP(3,10),
     A APP(10,10),CP(2),CXP(2),CPP(2,2)
      COMMON/CFLDEX/EXA,EXB,EXC
C
      NBAD=0
      R=X(3,KK) $ P1=PAR(5) $ P2=0.
      EX=EXC
      CALL COEFFI(R,P1,P2,EX,A,CX,CP,CXX,CXP,CPP,NBAD)
      IF(NBAD.EQ.0)GOTO 15 $ NBAD=NBAD+300 $ RETURN
C
   15 DO 25 KA=1,5 $ AP(KA)=0 $ IF(KA.LE.3)AX(KA)=0
      DO 25 KB=1,5 $ IF(KA.LE.3)AXP(KA,KB)=0
      IF(KA.LE.3.AND.KB.LE.3)AXX(KA,KB)=0
   25 APP(KA,KB)=0
C
      AX(3)=CX $ AP(5)=CP(1)
      AXX(3,3)=CXX  $  AXP(3,5)=CXP(1)
      APP(5,5)=CPP(1,1)
      RETURN $ END
```

```
      SUBROUTINE COEFFI(R,P1,P2,EX,A,AX,AP,AXX,AXP,APP,NBAD)
C  THIS COMPUTES TAU COEFFICIENTS TO BE USED IN PRESSURE FIELD
C  FUNCTION EXPONENT AND AS ADDITIVE TERM. THE COEFFICIENTS DEPEND ON
C
      DIMENSION AP(2),AXP(2),APP(2,2)
C
      NBAD=0
      REX=1./R**EX
      A=REX*(P1+P2*R)
C  A IS THE COEFFICIENT.  NEXT COMPUTE FIRST ORDER DERIVATIVES
      AX=REX*(-P1*EX/R+P2*(1.-EX))
      AP(1)=REX  $  AP(2)=REX*R
C  NEXT COMPUTE SECOND ORDER DERIVATIVES
      AXX=REX*(P1*EX*(EX+1.)/R-P2*(1.-EX)*EX)/R
      AXP(1)=REX*(-EX)/R  $  AXP(2)=REX*(1.-EX)
      APP(1,1)=0. $ APP(1,2)=0. $ APP(2,1)=0. $ APP(2,2)=0.
      RETURN  $ END
```

```
      SUBROUTINE SHOCK(R,T,POV,US,UP,RHO,NBAD)
C  THIS COMPUTES SHOCK VALUES USING PARAMETERS FROM /COMSHCK/
C  ALL ARGUMENTS ARE ASSUMED TO BE  EXPRESSED IN SI UNITS
C  ROUTINE USES ROMBIN AND SHTINT TO COMPUTE SHOCK ARRIVAL TIME
C
C    R     =  SHOCK DISTANCE (GIVEN)
C    T     =  SHOCK ARRIVAL TIME
C    POV   =  INCIDENTAL SHOCK OVERPRESSURE
C    US    =  SHOCK SPEED
C    UP    =  PARTICLE VELOCITY BEHIND SHOCK
C    RHO   =  SHOCK DENSITY
C    NBAD  =  ERROR INDICATOR.  NBAD.NE.0  IN CASE OF ERROR RETURN
C
      EXTERNAL SHTINT
C  INTEGRAND TO COMPUTE SHOCK ARRIVAL TIME
C
      COMMON/COMSHK/NPS,PARSH(4),VPARSH(4,4),SCDIS,SCPRE,SCTIM
      COMMON/AMBCHA/PZ,TZ,GAM,AMOL,CHVOL,CHEN,CHH,CHHER
      COMMON/CF2DER/GAMCAP,SNDSPD,PAR(4),ALOW,SCD,SCP,SCT
C
      GAMCAP=GAMCAP/SCP  $  SNDSPD=SNDSPD*SCD/SCT  $ ALOW=ALOW*SCD
      SCD=1.  $  SCP=1.  $  SCT=1.
      DO 15 KA=1,3
15    PAR(KA)=PARSH(KA)*SCPRE *SCDIS**KA
      PAR(4)=PARSH(4)*SCTIM
C  THIS CHANGED THE CONTENTS OF /CF2DER/ INTO SI UNITS
C
      POV=((PAR(3)/R+PAR(2))/R+PAR(1))/R
      CALL ROMBIN(SHTINT,ALOW,R,F,NBAD)
C  QUADRATURE TO COMPUTE SHOCK ARRIVAL TIME
      IF(NBAD.EQ.0) GO TO 30
      PRINT 20,NBAD
   20 FORMAT(1H ,*RETURN FROM SHOCK WITH NBAD= *,I5)
      RETURN
C
   30 CONTINUE
      T=F/SNDSPD +PAR(4)
      US=SQRT(SNDSPD**2*(1.+GAMCAP*POV))
      RHOZ=(AMOL/8.3143)*(PZ/TZ)
      UP=POV/(RHOZ*US)
      RHO=RHOZ*(1.+GAMCAP*POV)/(1.+(GAM-1.)*POV*0.5/(GAM*PZ))
      RETURN
      END
```

```
      SUBROUTINE SHOCK2(R,T,TR,TRR,P,PR,PRR,NBAD)
C  THIS ROUTINE COMPUTES SHOCK ARRIVAL TIME AND OVERPRESSURE FOR
C  GIVEN DISTANCE
CC
C  R  = SHOCK DISTANCE (GIVEN)
C  T  = SHOCK ARRIVAL TIME
C  TR, TRR  =  DERIVATIVES OF T WITH RESPECT TO R
C  P  = SHOCK OVERPRESSURE
C  PR, PRR  =  DERIVATIVES OF P WITH RESPECT TO R
C
C  ALL QUANTITIES ARE COMPUTED IN SI UNITS
C
      EXTERNAL SHTINT
      COMMON/COMSHK/NPS,PARS(4),VP(4,4),SCDS,SCPS,SCTS
      COMMON/CF2DER/GAMCAP,SNDSPD,CP(4),ALOW,SCD,SCP,SCT
C
      GAMCAP=GAMCAP/SCP  $  SNDSPD=SNDSPD*SCD/SCT  $  ALOW=ALOW*SCD
      SCD=1.  $  SCP=1.  $  SCT=1.
      DO 15 KA=1,3
15    CP(KA)=PARS(KA)*SCPS*SCDS**KA
      CP(4)=PARS(4)*SCTS
C  THIS TRANSFORMED /CF2DER/ INTO SI UNITS
C
      CALL ROMBIN(SHTINT,ALOW,R,T,NBAD)
C  QUADRATURE TO COMPUTE SHOCK ARRIVAL TIME
      IF(NBAD.EQ.0) GO TO 30
      PRINT 20,NBAD
   20 FORMAT(1H ,*RETURN FROM SHOCK2 WITH NBAD= *,I5)
   30 CONTINUE
C
      P=((CP(3)/R+CP(2))/R+CP(1))/R
      PR=-((3.*CP(3)/R+2.*CP(2))/R+CP(1))/R**2
      PRR=((12.*CP(3)/R+6.*CP(2))/R+CP(1))/R**3
      T=T/SNDSPD+CP(4)
      SQ=1.+GAMCAP*P
      TR=1./(SQRT(SQ)*SNDSPD)
      TRR=-0.5*GAMCAP*TR*PR/SQ
      RETURN
      END
```

```
      SUBROUTINE SHTINT(X,F,NBAD)
C  INTEGRAND FOR SHOCK ARRIVAL TIME COMPUTATION
C
      COMMON/CF2DER/GAMCAP,SNDSPD,PAR(4),ALOW,SCD,SCP,SCT
C
      IF(X.GT.1.E-10) GOTO 15  $  NBAD=1  $  RETURN
15    SQ=1.+GAMCAP*((PAR(3)/X+PAR(2))/X+PAR(1))/X
      IF(SQ.GT.1.E-100) GOTO 25  $  NBAD=2  $  RETURN
25    F=1./SQRT(SQ)  $  NBAD=0
      RETURN
      END
```

```
      SUBROUTINE ROMBIN (F,A,B,FINT,NBAD)
C  ROMBERG INTEGRATION SUBROUTINE
C
      DIMENSION T(10,20),CORKM(10)
C
      NBAD=0
      CALL F(A,FA,NBAD) $ IF(NBAD.NE.0)RETURN
      CALL F(B,FB,NBAD) $ IF(NBAD.NE.0)RETURN
      T(1,1)=(FA+FB)*0.5
      KM=1 $ KMA=1
C
   15 DEN=FLOAT(KMA)*2. $ FM=0
      DO 25 KA=1,KMA
      AC=FLOAT(1+2*(KMA-KA))/DEN
      BC=FLOAT(2*KA-1)/DEN
      ARG=AC*A+BC*B
      CALL F(ARG,FN,NBAD) $ IF(NBAD.NE.0)RETURN
      FM=FM+FN
   25 CONTINUE
      FM=FM/FLOAT(KMA)
      T(1,KM+1)=(T(1,KM)+FM)*0.5
C  THIS IS TRAPEZ. NOW COMPUTE ROMBERG
      KM=KM+1 $ KC=1 $ DDEN=1.
C
   35 KC=KC+1 $ DDEN=DDEN*4.
      CORKM(KC)=(T(KC-1,KM)-T(KC-1,KM-1))/(DDEN-1.)
      T(KC,KM)=T(KC-1,KM)+CORKM(KC)
      IF(KC.LT.KM.AND.KC.LT.10)GOTO 35
      IF(KC.GE.3)GOTO 45
C  AFTER AT LEAST 3 STEPS BRANCH TO 45 AND TEST CONVERGENCE
      KMA=KMA*2  $  GOTO 15
C
   45 DO 55 KA=2,KC
      TEST=ABS(CORKM(KA))
      IF(TEST.LE.ABS(T(KC,KM))*1.E-10)GOTO 65
      IF(TEST.LE.1.E-100)GOTO 65
   55 CONTINUE
      IF(KM.GE.20)GOTO 65
C  COMPUTE NOT MORE THAN 20 ROMBERG CORRECTIONS
      KMA=KMA*2  $  GOTO 15
C
   65 FINT=T(KC,KM)*(B-A)
      RETURN
      END
```

```
      SUBROUTINE SHODER(R,T,TR,TP,TRR,TRP,TPP,
     A POV,PR,PP,PRR,PRP,PPP,NBAD)
C  THIS COMPUTES FOR GIVEN DISTANCE R THE CORRESPONDING
C  SHOCK TIME T AND OVERPRESSURE POV, AND DERIVATIVES
C  SUBROUTINE USES F2SHCK TO COMPUTE SHOCK ARRIVAL TIME
C  ALL ARGUMENTS ARE ASSUMED TO BE IN SI UNITS
C
      DIMENSION TP(10),TRP(10),TPP(10,10),PP(10),PRP(10),PPP(10,10),
     A SPAR(10),X(5,1),FX(5),FP(10),FXX(5,5),FXP(5,10),FPP(10,10)
C
      COMMON/COMSHK/NPS,PARSH(4),VPARSH(4,4),SCDIS,SCPRE,SCTIM
      COMMON/CF2DER/GAMCAP,SNDSPD,PRS(4),ALOW,SCD,SCP,SCT
C
      GAMCAP=GAMCAP/SCP  $  SNDSPD=SNDSPD*SCD/SCT  $  ALOW=ALOW*SCD
      SCD=1. $  SCP=1.  $  SCT=1.
C  THIS CHANGED  /CF2DER/  TO SI UNITS
      IF(NPS.GE.O.AND.NPS.LE.5)GOTO 15
C  THIS IS CODED FOR NPS = NUMBER OF SHOCK PARAMETERS = 4
      NBAD=IABS(NPS) $ RETURN
   25 NBAD=25 $ RETURN
C
15    IF(R.LE.O.)GOTO 25
      NBAD=0
      IF(NPS.EQ.0)GOTO 55
C
C  NOW COMPUTE SHOCK OVERPRESSURE IN PASCALS BY 3-PARAMETER FORMULA
      DO 35 KA=1,3
35    SPAR(KA)=PARSH(KA)*SCPRE*SCDIS**KA
      SPAR(4)=PARSH(4)*SCTIM
C  SPAR IS FOR COMPUTATION OF POV IN PASCALS WHEN R IS IN METRES
C
      POV=((SPAR(3)/R+SPAR(2))/R+SPAR(1))/R
      PR=-((SPAR(3)*3./R+SPAR(2)*2.)/R+SPAR(1))/R**2
      PRR=((SPAR(3)*12./R+SPAR(2)*6.)/R+SPAR(1)*2.)/R**3
C
      DO 37 KA=1,10 $ PP(KA)=0 $ PRP(KA)=0
      TP(KA)=0 $ TRP(KA)=0
      DO 37 KB=1,10 $ TPP(KA,KB)=0
   37 PPP(KA,KB)=0
C
C  ASSUME THAT SHOCK PARAMETERS ARE NR. 6,7,8,9.
      PP(6)=1./R $ PP(7)=PP(6)/R $ PP(8)=PP(7)/R
      PRP(6)=-PP(7) $ PRP(7)=-2.*PP(8) $ PRP(8)=-3.*PP(8)/R
C  NEXT COMPUTE SHOCK ARRIVAL TIME. X(1)=PRESSURE, X(3)=TIME
      X(1,1)=0 $ X(2,1)=R $ X(3,1)=0
      CALL F2SHCK(X,1,SPAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C
      IF(NBAD.NE.0)GOTO 55
      T=F/SNDSPD  $  TR=FX(2)/SNDSPD  $  TRR=FXX(2,2)/SNDSPD
C
      DO 45 KA=1,NPS $ TP(5+KA)=FP(KA)/SNDSPD
      TRP(5+KA)=FXP(2,KA)/SNDSPD
      DO 45 KB=1,NPS
   45 TPP(5+KA,5+KB)=FPP(KA,KB)/SNDSPD
C
   55 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE F2SHCK(XX,KA,PAR,F,FX,FP,FXX,FXP,FPP,NBAD)
C  THIS IS SECOND CONSTRAINT COMPONENT FOR SHOCK FITTING
C
      DIMENSION XX(5,100),PAR(10),FX(5),FP(10),FXX(5,5),FXP(5,10),
     A FPP(10,10),SF(9)
C
      EXTERNAL F2DER
C
      COMMON/CF2DER/GAMCAP,SNDSPD,CPAR(4),ALOW,SCD,SCP,SCT
C  GAMCAP=((1.+GAM)/(2.*GAM))*(SCPR/AMBPR)
C  GAMCAP, SNDSPD AND ALOW ARE SET BY SUBROUTINE SCALSH
C
      DO 15 KB=1,4
   15 CPAR(KB)=PAR(KB)
C  THE PARAMETERS CPAR WILL BE USED BY SUBROUTINE F2DER
      X=XX(2,KA)
      DO 25 KB=1,3 $ DO 25 KC=1,3
   25 FXX(KB,KC)=0
      IF(X.GT.1.E-30) GOTO 35 $ NBAD=1 $ RETURN
C
   35 NBAD=0
      SQ=1.+GAMCAP*((PAR(3)/X+PAR(2))/X+PAR(1))/X
      IF(SQ.GT.1.E-50 ) GOTO 45 $ NBAD=2 $ RETURN
   45 FX(1)=0.  $  FX(2)=1./SQRT(SQ)  $  FX(3)=-SNDSPD
      FXX(2,2)=0.5*GAMCAP*FX(2)*((3.*PAR(3)/X+2.*PAR(2))/X
     A +PAR(1))/(X*X*SQ)
C  COMPUTE PARTS OF F2 AND DERIVATIVES BY MULTIPLE QUADRATURE
      CALL ROMULT(F2DER,ALOW,X,SF,NBAD)
      IF(NBAD.EQ.0) GOTO 55 $ NBAD=NBAD+10 $ RETURN
   55 F=SF(1)+(PAR(4)-XX(3,KA))*SNDSPD
      FP(1)=SF(2)  $  FP(2)=SF(3)  $  FP(3)=SF(4)  $  FP(4)=SNDSPD
      FPP(1,1)=SF(5) $ FPP(1,2)=SF(6) $ FPP(1,3)=SF(7)
      FPP(2,1)=SF(6) $ FPP(2,2)=SF(7) $ FPP(2,3)=SF(8)
      FPP(3,1)=SF(7) $ FPP(3,2)=SF(8) $ FPP(3,3)=SF(9)
      DO 65 KB=1,4 $ FPP(4,KB)=0 $ FPP(KB,4)=0 $ FXP(1,KB)=0
   65 FXP(3,KB)=0
      FXP(2,1)=-0.5*GAMCAP*FX(2)/(X*SQ)
      FXP(2,2)=FXP(2,1)/X $ FXP(2,3)=FXP(2,2)/X $ FXP(2,4)=0
      RETURN
      END
```

```
      SUBROUTINE F2DER(X,F,NBAD)
C  INTEGRAND FOR NINE COMPONENTS OF F2 AND DERIVATIVES
C  USED BY F2SHCK AS ARGUMENT OF ROMULT
C
      DIMENSION F(9)
C
      COMMON/CF2DER/GAMCAP,SNDSPD, PAR(4),ALOW,SCD,SCP,SCT
C  GAMCAP=((1.+GAM)/(2.*GAM))*(SCP /AMBPR)
C  GAMCAP, SNDSPD, ALOW AND SCALES ARE SET BY SUBROUTINE SCALSH
C
      NBAD=0 $ IF(X.GT.1.E-30) GOTO 15 $ NBAD=1 $ RETURN
C
   15 Y=1./X
      SQ=1.+GAMCAP*((PAR(3)*Y+PAR(2))*Y+PAR(1))*Y
      IF(SQ.GT.1.E-50 ) GOTO 25 $ NBAD=2 $ RETURN
C
C  INTEGRANDS CORRESPOND TO FOLLOWING QUANTITIES  .
C  F,FP(1),(2),(3),FPP(1,1),(1,2),(1,3)=(2,2),(2,3),(3,3)
   25 F(1)=1./SQRT(SQ)
      F(2)=-0.5*GAMCAP*F(1)*Y/SQ
      F(3)=F(2)*Y $ F(4)=F(3)*Y
      F(5)=-1.5*GAMCAP*F(3)/SQ
      F(6)=F(5)*Y $ F(7)=F(6)*Y $ F(8)=F(7)*Y $ F(9)=F(8)*Y
      RETURN
      END
```

```
      SUBROUTINE ROMULT(F,A,B,SF,NBAD)
C  ROMBERG INTEGRATION OF A 9-DIMENSIONAL VECTOR FUNCTION
C
      DIMENSION SF(9),T(9,10,20),FA(9),FB(9),FN(9),FM(9),CORKM(9,10)
C
      NBAD=0
      CALL F(A,FA,NBAD) $ IF(NBAD.NE.0) RETURN
      CALL F(B,FB,NBAD) $ IF(NBAD.NE.0) RETURN
      DO 14 KD=1,9
   14 T(KD,1,1)=(FA(KD)+FB(KD))*0.5
      KM=1 $ KMA=1
C
   15 DO 16 KD=1,9
   16 FM(KD)=0
      DEN=FLOAT(KMA)*2.
      DO 25 KA=1,KMA
      AC=FLOAT(1+2*(KMA-KA))/DEN $ BC=FLOAT(2*KA-1)/DEN
      ARG=AC*A+BC*B
      CALL F(ARG,FN,NBAD) $ IF(NBAD.NE.0) RETURN
      DO 23 KD=1,9
   23 FM(KD)=FM(KD)+FN(KD)
   25 CONTINUE
      DO 26 KD=1,9 $ FM(KD)=FM(KD)/FLOAT(KMA)
   26 T(KD,1,KM+1)=(T(KD,1,KM)+FM(KD))*0.5
C
C  THIS IS TRAPEZ. NEXT COMPUTE ROMBERG
      KM=KM+1 $ KC=1 $ DDEN=1.
C
   35 KC=KC+1 $ DDEN=DDEN*4.
      DO 37 L=1,9
      CORKM(L,KC)=(T(L,KC-1,KM)-T(L,KC-1,KM-1))/(DDEN-1.)
37    T(L,KC,KM)=T(L,KC-1,KM)+CORKM(L,KC)
      IF(KC.LT.KM.AND.KC.LT.10) GOTO 35
C
      IF(KM.GE.3) GOTO 45 $ KMA=KMA*2 $ GOTO 15
C  AFTER THREE STEPS TEST CONVERGENCE
C
   45 IF(KM.GE.20) GOTO 56
C  MAXIMUM OF 20 STEPS ALLOWED
C
      DO 53 L=1,9
      TEST=ABS(CORKM(L,KC))
C  KC=MIN(KM,10)
      IF(TEST.LE.1.E-100) GOTO 53
      IF(TEST.LE.ABS(T(L,KC,KM))*1.E-10) GOTO 53
      KMA=KMA*2 $ GOTO 15
   53 CONTINUE
C
   56 DO 58 L=1,9
   58 SF(L)=T(L,KC,KM)*(B-A)
      RETURN
      END
```

```
      SUBROUTINE PRIHIS(R,HIST,RHIST,NR)
C  THIS IS CALLED FROM FLOFLD TO PRINT FLOW HISTORY AT DISTANCE R
C  R                 = DISTANCE FROM THE EXPLOSION
C  HIST(5,100)       = TIME, OVERPRESSURE, VELOCITY, DENSITY, V**2*RHO/2
C  RHIST(5,5,100)    = VARIANCE-COVARIANCE MATRICES OF HIST
C  NR                = NUMBER OF NODES IN HIST
C
      DIMENSION HIST(5,100),RHIST(5,5,100)
      DIMENSION ERH(5),MES(5)
      DIMENSION PRH(5),NEX(5),NT(5),NU(5),S(5)
      COMMON/CSCALE/SCD,SCP,SCT
C
      DO 85 KA=1,NR
      IF(MOD(KA,35).NE.1)GOTO 47
      PRINT 25,R
   25 FORMAT(1H1,/,1H0,20X,27HFLOW HISTORY AT DISTANCE R=,1PE12.5)
      IF(SCD.EQ.1.)PRINT 26
   26 FORMAT(1H+,60X,1HM,/)
      IF(SCD.NE.1.)PRINT 27
   27 FORMAT(1H+,60X,3HSCD,/)
      PRINT 35
   35 FORMAT(1H0,/,1H ,5X,3HNR.,6X,4HTIME,4X,6HST.ER.,8X,
     A 9HOVERPRES.   ,2X,6HST.ER. ,9X,8HVELOCITY,2X,6HST.ER.,4X
     B5X,7HDENSITY,3X,6HST.ER., 7X,10HU**2*RHO/2,2X,6HST.ER.,/)
      IF(SCT.EQ.1.)PRINT 36
   36 FORMAT(1H+,15X,3H(S),6X,3H(S))
      IF(SCT.NE.1.)PRINT 37
   37 FORMAT(1H+,14X,5H(SCT),4X,5H(SCT))
      IF(SCP.EQ.1.)PRINT 38
   38 FORMAT(1H+,38X,4H(PA),6X,4H(PA))
      IF(SCP.NE.1.)PRINT 39
   39 FORMAT(1H+,38X,5H(SCP),4X,5H(SCP))
      IF(SCT.EQ.1..AND.SCD.EQ.1.)PRINT 41
   41 FORMAT(1H+,63X,5H(M/S),4X,5H(M/S))
      IF(SCT.NE.1..OR.SCD.NE.1.)PRINT 42
   42 FORMAT(1H+,67X,9H(SCD/SCT))
      IF(SCT.EQ.1..AND.SCP.EQ.1..AND.SCD.EQ.1.)PRINT 43
   43 FORMAT(1H+,91X,9H(KG/M**3))
      IF(SCT.NE.1..OR.SCP.NE.1..OR.SCD.NE.1.)PRINT 44
   44 FORMAT(1H+,87X,19H(SCP*SCT**2/SCD**2))
      IF(SCP.EQ.1.)PRINT 45
   45 FORMAT(1H+,114X,4H(PA),4X,4H(PA))
      IF(SCP.NE.1.)PRINT 46
   46 FORMAT(1H+,113X,5H(SCP),3X,5H(SCP))
   47 IF(MOD(KA,5).EQ.1)PRINT 471
  471 FORMAT(1H )
      MESC=0
      MESS=0

      DO 479 KB=1,5
      MES(KB)=1H
      PRH(KB)=HIST(KB,KA)
      ERH(KB)=SQRT(ABS(RHIST(KB,KB,KA)))
      IF(RHIST(KB,KB,KA).LT.0.0) MESS=1
      IF(RHIST(KB,KB,KA).LT.0.0) MES(KB)=1HN
      DM=AMAX1(ABS(PRH(KB)),ERH(KB))
      IF(DM.LE.0.) NEX(KB)=0
```

```
      IF(DM.GT.0.)NEX(KB)=INT(ALOG10(DM)+100.)-100
      PRH(KB)=PRH(KB)/10.**NEX(KB)
      ERH(KB)=ERH(KB)/10.**NEX(KB)
      S(KB)=1H+  $  IF(NEX(KB).LT.0) S(KB)=1H-
      NT(KB)=IABS(NEX(KB))/10  $  NU(KB)=IABS(NEX(KB))-NT(KB)
  479 CONTINUE

      PRINT 48,KA,(PRH(J),ERH(J),S(J),NT(J),NU(J),J=1,5)
 48   FORMAT(1H ,3X,I3,2X,5(3X,1H(,0PF7.4,2H  ,0PF6.4,3H )E,A1,I1,I1))
      IF(MESS.EQ.1) PRINT 49,(MES(J),J=1,5)
   49 FORMAT(1H+,9X,5(11X,A1,13X))
      IF(MESS.EQ.1)MESC=1
      IF(MOD(KA,35).NE.0.AND.KA.NE.NR)GOTO 85
      IF(MESC.EQ.1)PRINT 65
      MESC=0
   65 FORMAT(1H0,10X,35HNEGATIVE VARIANCES INDICATED BY "N")
      IF(SCT.EQ.1..AND.SCP.EQ.1..AND.SCD.EQ.1.)GOTO 85
      DENSC=SCP*(SCT/SCD)**2
      PRINT 70,SCT,DENSC
   70 FORMAT(1H0,10X,32HTHE OUTPUT IS SCALED AS FOLLOWS:,//,1H ,20X,
     A 4HTIME,10X,5HSCT =,1PE12.5,2H S,20X,7HDENSITY,3X,
     B 18HSCP*(SCT/SCD)**2 =,1PE12.5,8H KG/M**3)
      PRINT 75,SCP,SCP
   75 FORMAT(1H ,20X,12HOVERPRESSURE,2X,5HSCP =,1PE12.5,
     A 3H PA,19X,16HDYNAMIC PRESSURE,7X,5HSCP =,1PE12.5,3H PA)
      VELSC=SCD/SCT
      PRINT 80,VELSC,SCD
   80 FORMAT(1H ,20X,8HVELOCITY,2X,9HSCD/SCT =,1PE12.5,4H M/S,18X,
     A8HDISTANCE,15X,5HSCD =,1PE12.5,2H M)
   85 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE UTEST(SCD,SCP,SCT,RMIN,RMAX,RH,TMAX,PAR,VPAR,NPAR,
     A HIST,VHIST,NRHIST,UTST,NRUTST,NBAD)
C  THIS ROUTINE COMPUTES TEST VELOCITIES UTST BY INTEGRATION ALONG
C  CONSTANT TIME LINES
C  IT IS CALLED FROM FLOFLD AFTER HIST HAS BEEN COMPUTED
C
C  ROUTINE USES SHOCK, ROMBIN2 AND UTINT
C
      DIMENSION PAR(10),VPAR(10,10),HIST(5,2),VHIST(5,5,1),UTST(1)
      COMMON/AMBCHA/APRE,ATEM,AGAM,ADUM(5)
      COMMON/COUTST/TIME,CPAR(10),CAGAM,CAPRE
      EXTERNAL UTINT
C
      NBAD=0
      CAGAM=AGAM  $  CAPRE=APRE/SCP
      DO 10 KA=1,10
10    CPAR(KA)=PAR(KA)
      NRUTST=0
      IF(NRHIST.LE.0) RETURN
      NRUTST=1
      UTST(1)=HIST(3,1)  $  IF(NRHIST.EQ.1) RETURN
      IF(RH.GE.RMAX) RETURN
      RD=RH*SCD  $  R1=RD
      CALL SHOCK(RD,T1,POV,USH,UP,RHO,LBAD)
      IF(LBAD.EQ.0) GOTO 25
12    NBAD=100+IABS(LBAD)*10+NRUTST
13    PRINT 15,NBAD
      RETURN
15    FORMAT(1H0,10X,28HRETURN FROM UTEST WITH NBAD=I6)
C
25    DTIMD=(HIST(1,2)-HIST(1,1))*SCT
      TD=HIST(1,2)*SCT
27    R2=R1+DTIMD*USH
      CALL SHOCK(R2,T2,POV,USH,UP,RHO,LBAD)
      IF(LBAD.NE.0) GOTO 12
C
C  AT 35 START REGULA FALSI ALGORITHM TO FIND PROPER R
C  SUCH THAT SHOCK ARRIVES AT GIVEN TIME TD AT R
35    R3=R2+(TD-T2)*(R2-R1)/(T2-T1)
      CALL SHOCK(R3,T3,POV,USH,UP,RHO,LBAD)
      IF(LBAD.NE.0) GOTO 12
      IF(ABS(T3-TD).LE.DTIMD*0.01) GOTO 41
      R1=R2  $  T1=T2  $  R2=R3  $  T2=T3
      GOTO 35
C
41    RS=R3/SCD  $  TIME=T3/SCT
      CALL ROMBIN2(UTINT,RH,RS,UIN,LBAD)
C  QUADRATURE TO COMPUTE TEST VELOCITY
      IF(LBAD.EQ.0) GOTO 45
      NBAD=200+IABS(LBAD)*10+NRUTST
      GOTO 13
C
45    NRUTST=NRUTST+1
      UTST(NRUTST)=UP*(SCT/SCD)*(RS/RH)**2
     A *((POV/SCP+CAPRE)/(HIST(2,NRUTST)+CAPRE))**(1./AGAM)
     B +UIN/(AGAM*RH**2*(HIST(2,NRUTST)+CAPRE)**(1./AGAM))
C  THIS IS THE NEW TEST VELOCITY
```

```
      IF(RS.GE.RMAX) RETURN
      IF(NRUTST.GE.NRHIST) RETURN
      TD=HIST(1,NRUTST+1)*SCT
C  NEXT TIME VALUE FOR WHICH TEST VELOCITY IS NEEDED
      DTIMD=TD-T3
      R1=R3  $  T1=T3
      GOTO 27
      END
```

```
      SUBROUTINE UTINT(X,F,NBAD)
C  INTEGRAND ROUTINE FOR TEST VELOCITY COMPUTATION
C
      COMMON/COUTST/TH,PAR(10),GAM,APRE
      DIMENSION XX(3,1),FX(3),FP(10),FXX(3,3),FXP(3,10),FPP(10,10)
C
      XX(1,1)=TH  $  XX(2,1)=0  $  XX(3,1)=X
      CALL PFIELDC(XX,1,PAR,FF,FX,FP,FXX,FXP,FPP,NBAD)
      IF(NBAD.NE.0) RETURN
      F=X**2*(FF+APRE)**(1./GAM-1.)*FX(1)
      RETURN  $  END
```

```
      SUBROUTINE ROMBIN2 (F,A,B,FINT,NBAD)
C  ROMBERG INTEGRATION SUBROUTINE
C
      DIMENSION T(10,20),CORKM(10)
C
      NBAD=0
      CALL F(A, . .,NBAD) $ IF(NBAD.NE.0)RETURN
      CALL F(B,FB,NBAD) $ IF(NBAD.NE.0)RETURN
      T(1,1)=(FA+FB)*0.5
      KM=1 $ KMA=1
C
   15 DEN=FLOAT(KMA)*2. $ FM=0
      DO 25 KA=1,KMA
      AC=FLOAT(1+2*(KMA-KA))/DEN
      BC=FLOAT(2*KA-1)/DEN
      ARG=AC*A+BC*B
      CALL F(ARG,FN,NBAD) $ IF(NBAD.NE.0)RETURN
      FM=FM+FN
   25 CONTINUE
      FM=FM/FLOAT(KMA)
      T(1,KM+1)=(T(1,KM)+FM)*0.5
C  THIS IS TRAPEZ. NOW COMPUTE ROMBERG
      KM=KM+1 $ KC=1 $ DDEN=1.
C
   35 KC=KC+1 $ DDEN=DDEN*4.
      CORKM(KC)=(T(KC-1,KM)-T(KC-1,KM-1))/(DDEN-1.)
      T(KC,KM)=T(KC-1,KM)+CORKM(KC)
      IF(KC.LT.KM.AND.KC.LT.10)GOTO 35
      IF(KC.GE.3)GOTO 45
C  AFTER AT LEAST 3 STEPS BRANCH TO 45 AND TEST CONVERGENCE
      KMA=KMA*2  $  GOTO 15
C
   45 DO 55 KA=2,KC
      TEST=ABS(CORKM(KA))
      IF(TEST.LE.ABS(T(KC,KM))*1.E-10)GOTO 65
      IF(TEST.LE.1.E-100)GOTO 65
   55 CONTINUE
      IF(KM.GE.20)GOTO 65
C  COMPUTE NOT MORE THAN 20 ROMBERG CORRECTIONS
      KMA=KMA*2  $  GOTO 15
C
   65 FINT=T(KC,KM)*(B-A)
      RETURN
      END
```

```
      SUBROUTINE PRITST(R,RMAX,HIST,VHIST,NRHIST,UTST,NRUTST)
C
C  THIS ROUTINE PRINTS THE TEST VELOCITIES UTST TOGETHER WITH
C  CORRESPONDING VELOCITIES FROM THE ARRAY HIST
C  AND THE DYNAMIC PRESSURE COMPUTED USING THE TEST VELOCITY
C
      DIMENSION HIST(5,1),VHIST(5,5,1),UTST(1)
C
      COMMON/CSCALE/SCD,SCP,SCT

      IF(NRUTST.LE.0)RETURN
      IF(NRHIST.LE.20) PRINT 8
8     FORMAT(1H ,/////)

      DO 55 KA=1,NRUTST
      IF(MOD(KA,35).NE.1)GOTO 35
      IF(NRHIST.GT.20) PRINT 11
11     FORMAT(1H1)
      PRINT 15 ,R
   15 FORMAT(1H ,20X,32HTEST VELOCITIES FOR DISTANCE R =,1PE12.5)
      IF(SCD.EQ.1.)PRINT 151
  151 FORMAT(1H+,65X,1HM,/)
      IF(SCD.NE.1.)PRINT 152
  152 FORMAT(1H+,65X,3HSCD,/)
      PRINT 153
  153 FORMAT(1H ,83X,*DYNAMIC PRESSURE*)
      PRINT 34
34    FORMAT(1H ,10X,2HNR, 7X,4HTIME, 8X,8HVELOCITY,2X,
     A 6HST.ER.,9X,9HTEST VEL.,2X,6HUTST-U,)
      PRINT 340
  340 FORMAT(1H+,84X,13HUTST**2*RHO/2,)
      IF(SCT.EQ.1.)PRINT 341
  341 FORMAT(1H ,20X,3H(S))
      IF(SCT.NE.1.)PRINT 342
  342 FORMAT(1H ,19X,5H(SCT))
      IF(SCT.EQ.1..AND.SCD.EQ.1.)PRINT 343
  343 FORMAT(1H+,33X,5H(M/S),3X,5H(M/S),12X,5H(M/S),4X,5H(M/S))
      IF(SCT.NE.1..OR.SCD.NE.1.)PRINT 344
  344 FORMAT(1H+,37X,9H(SCD/SCT),16X,9H(SCD/SCT))
      IF(SCP.EQ.1.)PRINT 345
  345 FORMAT(1H+,88X,4H(PA))
      IF(SCP.NE.1.)PRINT 346
  346 FORMAT(1H+,88X,5H(SCP))

   35 IF(MOD(KA,5).EQ.1)PRINT 33
33     FORMAT(1H )

      TIM=HIST(1,KA)
      ERU=SQRT(ABS(VHIST(3,3,KA)))
      U=HIST(3,KA)  $  UT=UTST(KA)
      UD=UTST(KA)-HIST(3,KA)
      SUD=1H+  $  IF(UD.LT.0.) SUD=1H-
      DM=AMAX1(ABS(U),ERU,ABS(UT),ABS(UD))
      IF(DM.LE.0.) NEX=0
      IF(DM.GT.0.)NEX=INT(ALOG10(DM)+100.)-100
      NT=IABS(NEX)/10  $  NU=IABS(NEX)-NT
      SNEX=1H+  $  IF(NEX.LT.0) SNEX=1H-
```

```
      FCT=10.**NEX
      U=U/FCT  $  ERU=ERU/FCT  $  UT=UT/FCT  $  ABUD=ABS(UD)/FCT
      PRINT 36,KA,TIM,U,ERU,SNEX,NT,NU,UT,SUD,ABUD,SNEX,NT,NU
36    FORMAT(1H , 8X,I4,3X,1PE12.5,3X,1H(,OPF7.4,2H  ,OPF6.4,3H )E,
     A A1,I1,I1,4X,1H(,OPF7.4,2X,A1,OPF6.4,3H )E,A1,I1,I1)

      RHO=HIST(4,KA)$ DYP=UTST(KA)**2*RHO/2.
      PRINT 361,DYP
  361 FORMAT(1H+,85X,1PE11.4)

      IF(MOD(KA,35).NE.0.AND.KA.NE.NRUTST)GOTO 55
      IF(SCT.EQ.1..AND.SCD.EQ.1.)GOTO 55
      VELSC=SCD/SCT
      PRINT 45,SCD,SCT,VELSC
   45 FORMAT(1H0,15X,32HTHE OUTPUT IS SCALED AS FOLLOWS:,
     A 10X,8HDISTANCE,10X,5HSCD =,1PE12.5,2H M,/,
     B 1H ,57X,4HTIME,14X,5HSCT =,1PE12.5,2H S,/,
     C 1H ,57X,8HVELOCITY,6X,9HSCD/SCT =,1PE12.5,4H M/S)
      PRINT 451,SCP
  451 FORMAT(1H ,57X,16HDYNAMIC PRESSURE,2X,5HSCP =,1PE12.5,3H PA)

55    CONTINUE

      IF(NRHIST.LE.NRUTST) RETURN
      PRINT 65,RMAX
65    FORMAT(1H0,10X,22HTEST VELOCITIES CANNOT,
     A 36H BE COMPUTED FOR LATER TIMES BECAUSE, 6H RMAX=,1PE12.5,
     B 31H LIMITS THE COMPUTATIONAL RANGE)
      RETURN  $  END
```

```
      SUBROUTINE PLFFLD(SCD,SCP,SCT,D,HIST,RHIST,NR,UTST,NRUTST,TITLE)
C     THIS ROUTINE PLOTS THE FOLLOWING FLOW VARIABLES
C     OVERPRESSURE VERSUS TIME
C     VELOCITY VERSUS TIME
C     DENSITY VERSUS TIME
C     DYNAMIC PRESSURE VERSUS TIME
C     DYNAMIC PRESSURE FROM TEST VELOCITY VERSUS TIME
C     TEMPERATURE VERSUS TIME
C
C     SCD,SCP,SCT      =  SCALES OF DISTANCE, PRESSURE, TIME
C     D                =  DISTANCE FROM EXPLOSION
C     HIST(5,NR)       =  FLOW FIELD HISTORY(T,P,U,RHO,U**2*RHO/2)
C     RHIST(5,5,NR)    =  VARIANCE COVARIANCE MATRICES OF HIST
C     UTST(NRUTST)     =  PARTICLE VELOCITIES COMPUTED BY TEST PROCESS
C
      DIMENSION HIST(5,100),RHIST(5,5,100),TEMP(8)
      DIMENSION X(102),XA(100),Y(102),Y1(100),Y2(100)
      DIMENSION TITLE(3)
      DIMENSION UTST(100)
      COMMON/AMBCHA/ PO,TO,G,M,VC,EC
      COMMON/PLOT/AP,AH,Z(4),PLABL(4)
      REAL M
C
      CALL PLTBEG(8.7,11.2,1.0,13,PLABL)
C
C     THIS SECTION PLOTS OVERPRESSURE VERSUS TIME
C
      X(1)=HIST(1,1)-0.1*(HIST(1,NR)-HIST(1,1))
      Y(1)=0.
      X(2)=HIST(1,1)
      Y(2)=Y(1)
      DO 50 I=1,NR
      X(I+2)=HIST(1,I)
      Y(I+2)=HIST(2,I)+Y(1)
      EY=SQRT(ABS(RHIST(2,2,I)))
      Y1(I)=Y(I+2)-AH*EY
      Y2(I)=Y(I+2)+AH*EY
      XA(I)=HIST(1,I)
   50 CONTINUE
      N=NR+2
      CALL FIXSCA(X,N,5.0,XS,XMIN,XMAX,DX)
      CALL GRAPH(Y,Y1,Y2,XMIN,XMAX,YMIN,YMAX,XS,YS,DX,D,SCD,
     A AH,SCT,TITLE,N)
      CALL PLTWND(XMIN,XMAX,YMIN,YMAX)
      ENCODE(80,90,TEMP)
   90 FORMAT(18HOVERPRESSURE (PA)>)
      IF(SCP.NE.1.)ENCODE(80,91,TEMP)
   91 FORMAT(19HOVERPRESSURE (SCP)>)
      TX=XMIN-0.7*XS
      TY=(YMAX+YMIN)*0.5-8.5*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.,TX,TY)
      CALL PLTDTS(1,0,X,Y,N,0)
      CALL PLTDTS(1,0,XA,Y1,NR,0)
      CALL PLTDTS(1,0,XA,Y2,NR,0)
      CALL PLTPGE
C
C     THIS SECTION PLOTS VELOCITY VERSUS TIME
```

```
C
      Y(1)=0.0
      Y(2)=Y(1)
      DO 100 I=1,NR
      Y(I+2)=HIST(3,I)
      EY=SQRT(ABS(RHIST(3,3,I)))
      Y1(I)=Y(I+2)-AH*EY
      Y2(I)=Y(I+2)+AH*EY
  100 CONTINUE
      CALL GRAPH(Y,Y1,Y2,XMIN,XMAX,YMIN,YMAX,XS,YS,DX,D,SCD,
     A AH,SCT,TITLE,N)
      CALL PLTWND(XMIN,XMAX,YMIN,YMAX)
      ENCODE(80,110,TEMP)
  110 FORMAT(15HVELOCITY (M/S)>)
      IF(SCT.NE.1..OR.SCD.NE.1.)ENCODE(80,111,TEMP)
  111 FORMAT(19HVELOCITY (SCD/SCT)>)
      TY=(YMAX+YMIN)*0.5-7.0*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.,TX,TY)
      CALL PLTDTS(1,0,X,Y,N,0)
      CALL PLTDTS(1,0,XA,Y1,NR,0)
      CALL PLTDTS(1,0,XA,Y2,NR,0)
      CALL PLTDTS(4,0,XA,UTST,NRUTST,0)
      CALL PLTPGE
C
C     THIS SECTION PLOTS DENSITY VERSUS TIME
C
      Y(1)=(M/8.3143)*(PO/TO)*(SCD/SCT)**2*(1./SCP)
      Y(2)=Y(1)
      DO 120 I=1,NR
      Y(I+2)=HIST(4,I)
      EY=SQRT(ABS(RHIST(4,4,I)))
      Y1(I)=Y(I+2)-AH*EY
      Y2(I)=Y(I+2)+AH*EY
  120 CONTINUE
      CALL GRAPH(Y,Y1,Y2,XMIN,XMAX,YMIN,YMAX,XS,YS,DX,D,SCD,
     A AH,SCT,TITLE,N)
      CALL PLTWND(XMIN,XMAX,YMIN,YMAX)
      ENCODE (80,130,TEMP)
  130 FORMAT(18HDENSITY (KG/M**3)>)
      IF(SCT.NE.1..OR.SCD.NE.1..OR.SCP.NE.1.)ENCODE(80,131,TEMP)
  131 FORMAT(27HDENSITY (SCP*(SCT/SCD)**2)>)
      TY=(YMAX+YMIN)*0.5-8.5*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.,TX,TY)
      CALL PLTDTS(1,0,X,Y,N,0)
      CALL PLTDTS(1,0,XA,Y1,NR,0)
      CALL PLTDTS(1,0,XA,Y2,NR,0)
      CALL PLTPGE
C
C     THIS SECTION PLOTS DYNAMIC PRESSURE VERSUS TIME
C
      Y(1)=0.0
      Y(2)=Y(1)
      DO 140 I=1,NR
      Y(I+2)=HIST(5,I)
      EY=SQRT(ABS(RHIST(5,5,I)))
      Y1(I)=Y(I+2)-AH*EY
      Y2(I)=Y(I+2)+AH*EY
```

```
  140 CONTINUE
      CALL GRAPH(Y,Y1,Y2,XMIN,XMAX,YMIN,YMAX,XS,YS,DX,D,SCD,
     A AH,SCT,TITLE,N)
      CALL PLTWND(XMIN,XMAX,YMIN,YMAX)
      ENCODE(80,150,TEMP)
  150 FORMAT(33HDYNAMIC PRESSURE RHO*V**2/2 (PA)>)
      IF(SCP.NE.1.)ENCODE(80,151,TEMP)
  151 FORMAT(34HDYNAMIC PRESSURE RHO*V**2/2 (SCP)>)
      TY=(YMAX+YMIN)*0.5-16.0*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.,TX,TY)
      CALL PLTDTS(1,0,X,Y,N,0)
      CALL PLTDTS(1,0,XA,Y1,NR,0)
      CALL PLTDTS(1,0,XA,Y2,NR,0)
      CALL PLTPGE

C     THIS SECTION PLOTS DYNAMIC PRESSURE FROM TEST VELOCITY
C     VERSUS TIME.

      Y(1)=0.$ Y(2)=Y(1)
      DO 160 I=1,NRUTST
      Y(I+2)=HIST(4,I)*UTST(I)**2/2.
  160 CONTINUE
      BH=-2.$ NT=NRUTST+2
C  BY SETTING THE ERROR FACTOR BH=-2 INDICATE FOR GRAPH
C  THAT FOR THIS PLOT THE SAME SCALES AS PREVIOUSLY SHOLD
C  BE USED, AND THAT TITLE OF PLOT SHOULD BE DIFFERENT
      CALL GRAPH(Y,Y1,Y2,XMIN,XMAX,YMIN,YMAX,XS,YS,DX,D,SCD,
     A BH,SCT,TITLE,NT)
      CALL PLTWND(XMIN,XMAX,YMIN,YMAX)
      ENCODE(80,170,TEMP)
  170 FORMAT(33HDYNAMIC PRESSURE RHO*V**2/2 (PA)>)
      IF(SCP.NE.1.)ENCODE(80,171,TEMP)
  171 FORMAT(34HDYNAMIC PRESSURE RHO*V**2/2 (SCP)>)
      TY=(YMAX+YMIN)*0.5-16.0*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.,TX,TY)
      CALL PLTDTS(1,0,X,Y,NRUTST,0)
      CALL PLTPGE

C  THIS SECTION PLOTS TEMPERATURE VERSUS TIME
C
      Y(1)=TO
      Y(2)=Y(1)
C
      DO 180 I=1,NR
      PR=HIST(2,I)*SCP+PO
C  PRESSURE IN SI UNITS
      RO=HIST(4,I)*SCP*(SCT/SCD)**2
C  DENSITY IN SI UNITS
      Y(I+2)=PR*M/(RO*8.3143)
C  THIS IS TEMPERATURE=PRESSURE*(MOLAR MASS)/DENSITY  IN KELVINS
      EY=Y(I+2)*SQRT(RHIST(2,2,I)*(SCP/PR)**2
     A -2.0*RHIST(2,4,I)*SCP/(PR*HIST(4,I))+RHIST(4,4,I)/HIST(4,I)**2)
      Y1(I)=Y(I+2)-AH*EY
      Y2(I)=Y(I+2)+AH*EY
180   CONTINUE
C
      CALL GRAPH(Y,Y1,Y2,XMIN,XMAX,YMIN,YMAX,XS,YS,DX,D,SCD,AH,SCT,
```

```
     A TITLE,N)
      CALL PLTWND(XMIN,XMAX,YMIN,YMAX)
      ENCODE(80,190,TEMP)
190   FORMAT(16HTEMPERATURE (K)>)
      TY=(YMAX+YMIN)*0.5-8.0*0.1*YS
      CALL PLTSYM(0.1,TEMP,90.0,TX,TY)
      CALL PLTDTS(1,0,X,Y,N,0)
      CALL PLTDTS(1,0,XA,Y1,NR,0)
      CALL PLTDTS(1,0,XA,Y2,NR,0)
      CALL PLTPGE
C
      RETURN
      END
```

```
      SUBROUTINE GRAPH(Y,Y1,Y2,XMIN,XMAX,YMIN,YMAX,XS,YS,DX,D,SCD,
     A AH,SCT,TITLE,N)
C  AUXILIARY ROUTINE OF PLFFLD FOR ESTABLISHING SCALES ETC.
C  IT IS CALLED FROM PLFFLD

      DIMENSION Y(102),Y1(100),Y2(100),TITLE(3)
      DIMENSION TEMP(8)

      IF(AH.LT.-1.)GOTO 35
C  IF ERROR FACTOR IS NEGATIVE THEN THIS IS A PLOT OF
C  THE DYNAMIC PRESSURE FROM TEST VELOCITY.  IN THIS CASE
C  USE THE SAME SCALES AS FOR PREVIOUS PLOT
      CALL FIXSCA(Y,N,4.0,YS,YMIN,YMAX,DY)
      CALL CONSCA(Y1,N-2,4.0,YS,YMIN,YMAX,DY)
      CALL CONSCA(Y2,N-2,4.0,YS,YMIN,YMAX,DY)
   35 CONTINUE
      CALL PLTSCA(2.5,4.0,XMIN,YMIN,XS,YS)

      CALL PLTAXS(DX,DY,XMIN,XMAX,YMIN,YMAX,4)
      CALL LABAX(DX*2.,DY*2.,XMIN,XMAX,YMIN,YMAX)

      ENCODE(80,50,TEMP) TITLE
   50 FORMAT(3A10,1H>)
      TX=(XMAX+XMIN)*0.5-15.0*0.1*XS
      TY=YMAX+1.0*YS
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)
      IF(SCD.EQ.1.)ENCODE(80,60,TEMP) D
   60 FORMAT(28HDISTANCE FROM THE EXPLOSION ,F7.2,8H METRES>)
      IF(SCD.NE.1.) ENCODE(80,61,TEMP) D
61    FORMAT(28HDISTANCE FROM THE EXPLOSION ,F7.2,8H (SCD) >)
      TX=(XMAX+XMIN)*0.5-22.0*0.1*XS
      TY=YMAX+.75*YS
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)
      IF(AH.LT.-1.)GOTO 72

      ENCODE(80,70,TEMP) AH
   70 FORMAT(*ERROR LIMITS CORRESPOND TO *,F5.2,* STANDARD ERRORS>*)
      TX=(XMAX+XMIN)*0.5-24.0*0.1*XS
      TY=YMAX+0.5*YS
      CALL PLTSYM(C.1,TEMP,0.0,TX,TY)
      GOTO 78

   72 ENCODE(80,73,TEMP)
   73 FORMAT(46HDYNAMIC PRESSURE COMPUTED USING TEST VELOCITY>)
      TX=(XMAX+XMIN)/2.-23.0*0.1*XS
      TY=YMAX+0.5*YS
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)

   78 CONTINUE
      ENCODE(80,80,TEMP)
   80 FORMAT(9HTIME (S)>)
      IF(SCT.NE.1.)ENCODE(80,81,TEMP)
   81 FORMAT(11HTIME (SCT)>)
      TX=(XMAX+XMIN)*0.5-5.0*0.1*XS
      TY=YMIN-0.5*YS
      CALL PLTSYM(0.1,TEMP,0.0,TX,TY)
      RETURN
      END
```

LIST OF SYMBOLS

a, b, c, d	- shock fitting parameters.
A, B, C	- fitting parameters of a single overpressure history.
$A(r), B(r), C(r)$	- functions, defined by Equation 6.3.
A_1, A_2, B_1, B_2, C_1	- overpressure field fitting parameters.
c_o	- sound speed in ambient air, m/s.
c_{ab}, c_{12}, etc.	- correlation coefficients.
e	- specific internal energy, J/kg.
E	- effective energy released by the explosion, J.
e_H, e_p, etc.	- standard error of the quantity in the index.
h	- elevation of the pressure probe, m.
H	- elevation of the center of the explosion, m.
M	- molar mass, kg/mol.
p	- pressure, Pa.
p_o	- ambient pressure, Pa.
$p_f(r, t; A_1, A_2, B_1, B_2, C_1)$	- fitted overpressure field function.
$p_h(t; A, B, C)$	- fitted overpressure history function, Pa.
$p_s(r; a, b, c)$	- fitted shock overpressure function, Pa.
Q	- exponent in Equation 4.3.
r	- distance from the center of the explosion, m.
r_o	- a reference distance used in shock fitting, m.
s_r, s_p, s_t	- distance pressure and time scales used in the calculations, m, Pa, s.
t	- time after the explosion, s.
$t_s\ (r; a, b, c, d)$	- fitted shock arrival time function, s.
T_o	- ambient temperature, K.

V	- volume of the fireball, m^3.
V_θ	- variance-covariance matrix of θ.
x	- range (ground distance) from the explosion, m.
[illegible]	- ratio of specific heats.
θ	- a model fitting parameter vector.
ρ	- density, kg/m^3.
τ	- time after shock arrival = $t - t_s$, s.

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314
1	Director of Defense Research & Engineering ATTN: DD/TWP Washington, DC 20301
1	Asst. to the Secretary of Defense (Atomic Energy) ATTN: Document Control Washington, DC 20301
1	Director Defense Advanced Research Projects Agency ATTN: Tech Lib 1400 Wilson Boulevard Arlington, VA 22209
2	Director Federal Emergency Management Agency ATTN: D. A. Bettge Technical Library Washington, DC 20472
1	Director Defense Intelligence Agency ATTN: DT-2/Wpns & Sys Div Washington, DC 20301
1	Director National Security Agency ATTN: E. F. Butala, R15 Ft. George G. Meade, MD 20755
1	Director Joint Strategic Target Planning Staff JCS Offut AFB Omaha, NB 68113

No. of Copies	Organization
1	Director Defense Communications Agency ATTN: 930 Washington, DC 20305
9	Director Defense Nuclear Agency ATTN: DDST TIPL/Tech Lib SPSS/K. Goering SPTD/T. Kennedy SPAS/P.R. Rohr G. Ullrich STSP/COL Kovel NATD NATA Washington, DC 20305
2	Commander Field Command, DNA ATTN: FCPR FCTMOF Kirtland AFB, NM 87117
1	Commander Field Command, DNA Livermore Branch ATTN: FCPRL P.O. Box 808 Livermore, CA 94550
1	HQDA DAMA-ART-M Washington, DC 20310
1	Program Manager US Army BMD Program Office ATTN: John Shea 5001 Eisenhower Avenue Alexandria, VA 22333

DISTRIBUTION LIST

No. of Copies	Organization
2	Director US Army BMD Advanced Technology Center ATTN: CRDABH-X CRDABH-S Huntsville, AL 35807
1	Commander US Army BMD Command ATTN: BDMSC-TFN/N.J. Hurst P.O. Box 1500 Huntsville, AL 35807
1	Commander US Army Engineer Division ATTN: HNDED-FD P.O. Box 1500 Huntsville, AL 35807
2	Deputy Chief of Staff for Operations and Plans ATTN: Technical Library Director of Chemical & Nuc Operations Department of the Army Washington, DC 20310
2	Office, Chief of Engineers Department of the Army ATTN: DAEN-MCE-D DAEN-RDM 890 South Pickett Street Alexandria, VA 22304
3	Commander US Army Engineer Waterways Experiment Station ATTN: Technical Library Jim Watt Jim Ingram P.O. Box 631 Vicksburg, MS 39180
1	Commander Naval Weapons Center ATTN: Tech Svcs Br, Code 3433 China Lake, CA 93555
1	US Army MERADCOM ATTN: DRDME-EM, D. Frink Fort Belvoir, VA 22060
1	Commander US Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TDC Dover, NJ 07801
2	Commander Armament R&D Center US Army AMCCOM ATTN: SMCCR-LCN-F, W. Reiner SMCCR-TSS Dover, NJ 07801
1	Commander US Army Armament, Munitions and Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299
1	Director Benet Weapons Laboratory Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCB-TL Watervliet, NY 12189

DISTRIBUTION LIST

No. of Copies	Organization	No. of Copies	Organization
1	Commander Naval Electronic Systems Com ATTN: PME 117-21A Washington, DC 20360	1	Commander Naval Weapons Evaluation Fac ATTN: Document Control Kirtland AFB, NM 87117
1	Commander Naval Facilities Engineering Command Washington, DC 20360	1	Commander Naval Research Laboratory ATTN: Code 2027, Tech Lib Washington, DC 20375
1	Commander Naval Sea Systems Command ATTN: SEA-62R Department of the Navy Washington, DC 20362	1	AFIT (Lib Bldg. 640, Area B) Wright-Patterson AFB Ohio 45433
3	Officer-in-Charge(Code L31) Civil Engineering Laboratory Naval Constr Btn Center ATTN: Stan Takahashi R. J. Odello Technical Library Port Hueneme, CA 93041	1	AFSC/SDOA Andrews Air Force Base MD 20334
		1	AFATL/DLODL, Tech Lib Eglin AFB, FL 32542-5000
		1	AFWL/SUL Kirtland AFB, NM 87117
1	Commander David W. Taylor Naval Ship Research & Development Ctr ATTN: Lib Div, Code 522 Bethesda, MD 20084	1	AFATL (DLYV) Eglin AFB, FL 32542-5000
		1	RADC (EMTLD/Docu Libray) Griffiss AFB, NY 13441
1	Commander Naval Surface Weapons Center ATTN: DX-21, Library Br. Dahlgren, VA 22448	1	AFWL/NTES (R. Henny) Kirtland AFB, NM 87117
2	Commander Naval Surface Weapons Center ATTN: Code WA501/Navy Nuclear Programs Office Code WX21/Tech Library Silver Spring, MD 20910	1	AFWL/NTE, CPT J. Clifford Kirtland AFB, NM 87117
		2	Commander-in-Chief Strategic Air Command ATTN: NRI-STINFO Lib Offutt AFB, NB 68113

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Aviation Research and Development Command ATTN: AMSAV-E 4300 Goodfellow Boulevard St. Louis, MO 63120
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications-Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703
3	Commander US Army Electronics Research and Development Command ATTN: DELSD-L AMDSD-E, W. S. McAfee AMDSD-EI, J. Roma Fort Monmouth, NJ 07703-5301
7	Director US Army Harry Diamond Labs ATTN: Mr. James Gaul Mr. L. Belliveau Mr. J. Meszaros Mr. J. Gwaltney Mr. Bill Vault Mr. R. J. Bostak Mr. R. K. Warner 2800 Powder Mill Road Adelphi, MD 20783
4	Director US Army Harry Diamond Labs ATTN: DELHD-TA-L DRXDO-TI/002 DRXDO-NP DELHD-RBA/J. Rosado 2800 Powder Mill Road Adelphi, MD 20783
1	Commander US Army Missile Command ATTN: AMSMI-R Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-YDL Redstone Arsenal, AL 35898
3	Commander US Army Natick Research and Development Center ATTN: DRXRE/Dr. D. Sieling DRXNE-UE/A. Johnson J. Calligeros Natick, MA 01762
1	Commander US Army Tank Automotive Rsch and Development Command ATTN: AMSTA-TSL Warren, MI 48090
1	Commander US Army Foreign Science and Technology Center ATTN: Rsch & Cncepts Br 220 7th Street , NE Charlottesville, VA 22901

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Logistics Management Ctr ATTN: ATCL-O Mr. Robert Cameron Fort Lee, VA 23801
3	Commander US Army Materials and Mechanics Research Center ATTN: Technical Library DRXMR-ER, Joe Prifti Eugene de Luca Watertown, MA 02172
1	Commander US Army Research Office P.O. Box 12211 Research Triangle Park NC 27709-2211
4	Commander US Army Nuclear & Chemical Agency ATTN: ACTA-NAW MONA-WE Technical Library LTC Finno 7500 Backlick Rd, Bldg. 2073 Springfield, VA 22150
1	Commander US Army TRADOC ATTN: DCST&E Fort Monroe, VA 23651
2	Director US Army TRADOC Systems Analysis Activity ATTN: LTC John Hesse ATAA-SL White Sands Missile Range NM 8802
1	Commander US Combined Arms Combat Developments Activity ATTN: ATCA-CO, Mr. L. C. Pleger Fort Leavenworth, KS 66027
1	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905
1	Commander US Army Development & Employment Agency ATTN: MODE-TED-SAB Fort Lewis, WA 98433
1	Commandant Interservice Nuclear Weapons School ATTN: Technical Library Kirtland AFB, NM 87115
1	Chief of Naval Material ATTN: MAT 0323 Department of the Navy Arlington, VA 22217
2	Chief of Naval Operations ATTN: OP-03EG OP-985F Department of the Navy Washington, DC 20350
1	Chief of Naval Research ATTN: N. Perrone Department of the Navy Arlington, VA 22217
1	Director Strategic Systems Projects Ofc ATTN: NSP-43, Tech Library Department of the Navy Washington, DC 20360

DISTRIBUTION LIST

No. of Copies	Organization
1	FTD/NIIS Wright-Patterson AFB Ohio 45433
1	Director Lawrence Livermore Lab. ATTN: Tech Info Dept L-3 P.O. Box 808 Livermore, CA 94550
2	Director Los Alamos Scientific Lab. ATTN: Doc Control for Rpts Lib P.O. Box 1663 Los Alamos, NM 87544
2	Director Sandia Laboratories ATTN: Doc Control for 3141 Sandia Rpt Collection L. J. Vortman Albuquerque, NM 87115
1	Director Sandia Laboratories Livermore Laboratory ATTN: Doc Control for Tech Lib P.O. Box 969 Livermore, CA 94550
1	Director National Aeronautics and Space Administration Scientific & Tech Info Fac P.O. Box 8757 Baltimore/Washington International Airport MD 21240
1	Aerospace Corporation ATTN: Tech Info Services P.O. Box 92957 Los Angeles, CA 90009
1	Agbabian Associates ATTN: M. Agbabian 250 North Nash Street El Segundo, CA 90245
1	The BDM Corporation ATTN: Richard Hensley P.O. Box 9274 Albuquerque International Albuquerque, NM 87119
1	The Boeing Company ATTN: Aerospace Library P.O. Box 3707 Seattle, WA 98124
2	California Research and Technology ATTN: M. Rosenblatt F. Sauer Suite B 130 11875 Dublin Blvd Dublin, CA 94568
1	Carpenter Research Corporation ATTN: H. Jerry Carpenter 6230 Scotmist Drive Rancho Palos Verdes, CA 90274
1	Goodyear Aerospace Corp ATTN: R. M. Brown, Bldg 1 Shelter Engineering Litchfield Park, AZ 85340
1	Director Inst for Defense Analyses ATTN: Library 1801 Beauregard St. Alexandria, VA 22311

DISTRIBUTION LIST

No. of Copies	Organization
6	Kaman AviDyne ATTN: Dr. R. Reutenick (4 cys) Mr. S. Criscione Mr. R. Milligan 83 Second Avenue Northwest Industrial Park Burlington, MA 01830
3	Kaman Sciences Corporation ATTN: Library P. A. Ellis F. H. Shelton 1500 Garden of the Gods Road Colorado Springs, CO 80907
1	Science Applications, Inc ATTN: Technical Library 1250 Prospect Plaza La Jolla, CA 92037
1	Kaman-TEMPO ATTN: DASIAC P.O. Drawer QQ Santa Barbara, CA 93102
1	Kaman-TEMPO ATTN: E. Bryant, Suite UL-1 715 Shamrock Road Bel Air, MD 21014
1	Lockheed Missiles & Space Co. ATTN: J. J. Murphy, Dept. 81-11, Bldg. 154 P.O. Box 504 Sunnyvale, CA 94086
1	Martin Marietta Aerospace Orlando Division ATTN: G. Fotieo P.O. Box 5837 Orlando, FL 32805

No. of Copies	Organization
2	McDonnell Douglas Astronautics Corporation ATTN: Robert W. Halprin K.A. Heinly 5301 Bolsa Avenue Huntington Beach, CA 92647
1	The Mitre Corporation ATTN: Library P.O. Box 208 Bedford, MA 01730
1	New Mexico Engineering Research Institute (CERF) ATTN: J. Leigh P.O. Box 25 UNM Albuquerque, NM 87131
2	Physics International Corp 2700 Merced Street San Leandro, CA 94577
2	R&D Associates ATTN: Technical Library Allan Kuhl P.O. Box 9695 Marina del Rey, CA 90291
1	RCA Government Communications Systems 13-5-2 Front & Cooper Streets Camden, NJ 08102
2	Science Applications, Inc. ATTN: W. Layson John Cockayne PO BOX 1303 1710 Goodridge Drive McLean, VA 22102

DISTRIBUTION LIST

No. of Copies	Organization
2	Systems, Science and Software ATTN: C. E. Needham Lynn Kennedy PO Box 8243 Albuquerque, NM 87198
3	Systems, Science and Software ATTN: Technical Library R. Duff K. Pyatt PO Box 1620 La Jolla, CA 92037
1	TRW Electronics & Defense ATTN: Benjamin Sussholtz One Space Park Redondo Beach, CA 90278
2	Union Carbide Corporation Holifield National Laboratory ATTN: Doc Control for Tech Lib Civil Defense Research Proj PO Box X Oak Ridge, TN 37830
1	Weidlinger Assoc. Consulting Engineers 110 East 59th Street New York, NY 10022
1	Battelle Memorial Institute ATTN: Technical Library 505 King Avenue Columbus, OH 43201
1	California Inst of Tech ATTN: T. J. Ahrens 1201 E. California Blvd. Pasadena, CA 91109
2	Denver Research Institute University of Denver ATTN: Mr. J. Wisotski Technical Library PO Box 10127 Denver, CO 80210

No. of Copies	Organization
1	IIT Research Institute ATTN: Milton R. Johnson 10 West 35th Street Chicago, IL 60616
1	TRW Ballistic Missile Division ATTN: H. Korman, Mail Station 526/614 P.O. Box 1310 San Bernardino, CA 92402
1	J. D. Haltiwanger Consulting Services B106a Civil Engineering Bldg. 208 N. Romine Street Urbana, IL 61801
1	Massachusetts Institute of Technology Aeroelastic and Structures Research Laboratory ATTN: Dr. E. A. Witmer Cambridge, MA 02139
2	Southwest Research Institute ATTN: Dr. W. E. Baker A. B. Wenzel 8500 Culebra Road San Antonio, TX 78228
1	SRI International ATTN: Dr. G. R. Abrahamson 333 Ravenswood Avenue Menlo Park, CA 94025
1	Stanford University ATTN: Dr. D. Bershader Durand Laboratory Stanford, CA 94305
1	Washington State University Physics Department ATTN: G. R. Fowles Pullman, WA 99163

DISTRIBUTION LIST

Organization

Aberdeen Proving Ground

Dir, USAMSAA
  ATTN: AMXSY-D
    AMXSY-MP, H. Cohen
Cdr, USATECOM
  ATTN: AMSTE-TO-F
Cdr, CRDC, AMCCOM
  ATTN: SMCCR-RSP-A
    SMCCR-MU
    SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number____________________ Date of Report__________

2. Date Report Received______________________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)______________

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.)______________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate.______________

6. *General Comments.* What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

CURRENT ADDRESS

Name

Organization

Address

City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

Name

Organization

Address

City, State, Zip

(Remove this sheet along the perforation, fold as indicated, staple or tape closed, and mail.)

FOLD HERE

Director
US Army Ballistic Research Laboratory
ATTN: AMXBR-OD-ST
Aberdeen Proving Ground, MD 21005-5066

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC
POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
US Army Ballistic Research Laboratory
ATTN: AMXBR-OD-ST
Aberdeen Proving Ground, MD 21005-9989

FOLD HERE