1983-08-01

Development of a Computer-Aided Graphics Capability to Display Field Test Data

L. Allen Adams
Brigham Young University - Provo

DEVELOPMENT OF A COMPUTER-AIDED GRAPHICS CAPABILITY TO DISPLAY FIELD TEST DATA

L. ALLEN ADAMS

DEVELOPMENT OF A COMPUTER-AIDED GRAPHICS CAPABILITY TO DISPLAY FIELD TEST DATA

A Thesis

Presented to the

Department of Civil Engineering

Brigham Young University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

L. Allen Adams

August 1983

This Thesis, by L. Allen Adams, is accepted in its present form by the Department of Civil Engineering of Brigham Young University as satisfying the thesis requirements for the degree of Master of Science.

Steven E. Benzley
Committee Chairman

Michael B. Stephenson
Committee Member

29 JUNE 1983
Date

Henry N. Christiansen
Department Chairman

TABLE OF CONTENTS

LIST OF FIGURES

Figure

ACKNOWLEDGMENTS

I would like to thank Dr. Steven E. Benzley for extending the opportunity to work with him on this project, and for his advice, direction, and assistance.

I would especially like to express my appreciation to my wife, Barbara, for her continual love, support, and encouragement.

CHAPTER I

Introduction

Survivability of Air Force weapons systems and facilities in nuclear and conventional environments requires an understanding of the mechanics of geotechnical materials. An understanding of the response of these materials when subjected to blast and shock loadings is required in order to determine and define the actual loads that will be transferred to critical Air Force structures. For this purpose the Air Force conducts numerous field tests and accumulates massive amounts of field test data. A convenient method of manipulating and visually displaying the field data is necessary for efficient and accurate analysis of the data. The necessity of this capability called for the development of a user-friendly interactive computer graphics program. The major consideration of this thesis is the development of an experimental data interpolation program entitled EXPERT. The work done in this thesis is part of a research project sponsored by the Air Force Weapons Laboratory (AFWL) at Kirtland Air Force Base.

In order to test the effects of dynamic loading in soils and on structures, strain and accelerometer gauges are placed at various points providing strain, acceleration, velocity, and displacement time histories at those selected points through the profile. It is impractical to place more than 25 or 30 such devices through the profile, yet this is an insufficient number to graphically illustrate the particular response function using color computer graphics packages such as MOVIE.BYU. The experimental data is too 'sparse'. In order to graphically display the response function through the profile it is necessary to perform an interpolation over the two dimensional domain using the 'sparse' data as the "pivotal values". By performing a sufficient number of interpolations (i.e., several hundred) the values over the whole profile can be determined, providing the necessary data base to display the test results with continuous tone color and/or contour, or equipotential, lines. The interpolation scheme employed by EXPERT was developed by Akima [1] and is part of both the ACM and IMSL libraries. It was modified slightly by the author in order to adapt it for use in EXPERT. The interpolation algorithm is explained in greater detail in Chapter 2.

Chapter 3 describes the capabilities and limitations of EXPERT. Chapter 4 demonstrates the practical use of EXPERT by comparing the results of an actual field test and the results from a finite element analysis of the event. Chapter 5 gives further recommendations and conclusions. An appendix is included listing the source code, a user's manual and some supporting software.

CHAPTER II

Interpolation Algorithm

There were five main criteria in the selection and incorporation of an interpolation algorithm for an experimental data graphical display system for AFWL field test data. These criteria are

1) the interpolated surface should pass through all data points rather than merely approach those points;
2) the interpolation routine should accept data points located at points other than regular rectangular grid points;
3) the interpolated surface should be continuous, that is, the interpolated values at adjacent edges of two elements should coincide;
4) the interpolated surface should be smooth, that is, the slope of the surface should not change abruptly at the intersection of elements; and
5) the interpolated surface should not exhibit excessive undulations between data points.

It was necessary that any interpolation scheme used conform as nearly as practical to these five criteria. An algorithm entitled "Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points", developed by Akima [1] at the Institute for Telecommunication Sciences, was selected as the most acceptable algorithm. It was tested using several configurations of data points to determine its accuracy and reliability. The following discussion of this method indicates how the algorithm handles each of the five criteria. Portions of the following discussion follow closely that given by Akima [2,3].

In this method the **x-y** plane is 'triangulated', or divided into a number of triangular 'cells', each having as its vertexes the projections of three data points in the plane. To each triangular cell a bivariate fifth-degree polynomial is applied, with estimated values of partial derivatives at each data point used in determining the polynomial. This triangulation allows for an irregular distribution of data points, satisfying the second of the five criteria. Undulation of the interpolated surface between data points is minimized in two ways. First, a fifth-degree polynomial is used to estimate the value of **z** at a point. Use of this higher order polynomial permits greater accuracy in the definition of the function. Second, rather than using a single global function over the whole domain, a collection of local functions, or triangular 'cells', is used. These local functions can more accurately define the function in any given section of the domain.

The method of triangulation utilized by the algorithm is illustrated in the accompanying figures. Assume five data points as shown in Figure 1-a. The closest pair of data points is determined and connected (Figure 1-b). On the basis of distance from the midpoint of the closest pair of data points one point is added at a time, closest first (Figure 1-c). This arrangement assures that the new point always lies outside any previously constructed polygons. Each time a new point is added new triangles are constructed by connecting the new point with any old points visible from the new point (Figure 1-d). Notice the relative size and shape of triangles #3 and #4 in Figure 1-e. Analogous to finite element calculations in which long slender or odd shaped elements give inaccurate results, such triangles would provide a questionable basis of interpolation. A method, therefore, of optimizing the triangles is desirable. The algorithm employs a criterion proposed by Lawson [4] which dictates that "when a set of four points are vertexes of a quadrilateral with each internal angle smaller than π, one chooses, out of two possible ways of partitioning the quadrilateral into a pair of triangles, the partitioning that maximizes the minimum interior angle of the two triangles produced." The result is a more optimal triangulation of the **x-y** domain (Figure 1-f).

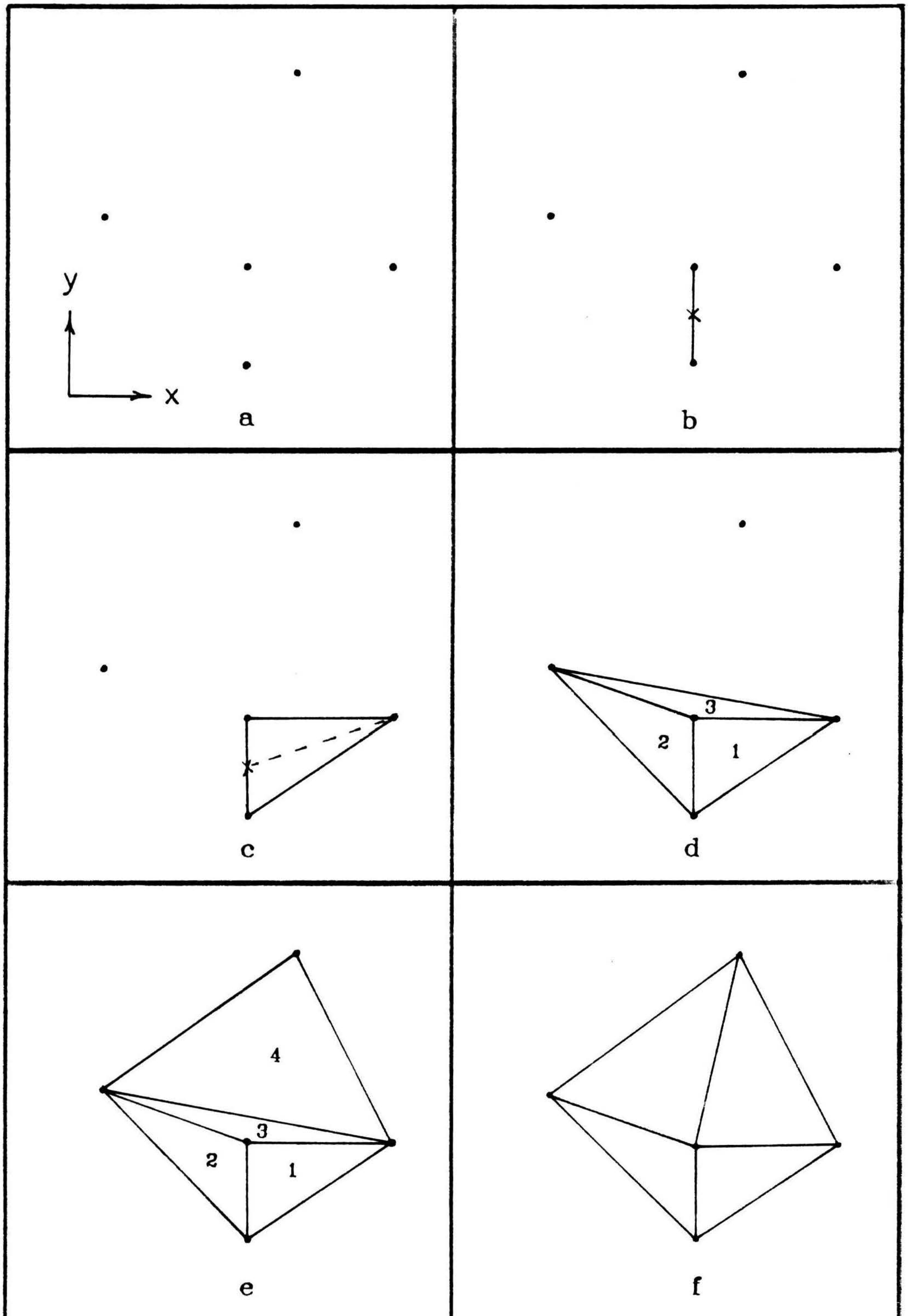


Figure 1 Method of Triangulation

Interpolation In a Triangle

The interpolation algorithm is based on the following three premises:

i) The value of the function at some point (x,y) in a triangle is determined by a bivariate fifth-degree polynomial in a Cartesian coordinate system with **x** and **y** axes of the form:

$$z(x,y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk} x^j y^k . \qquad (2\text{-}1)$$

This gives 21 coefficients to be determined for each point.

ii) The values of the function and its first- and second order partial derivatives (i.e., z, z_x, z_y, z_{xx}, z_{xy}, and z_{yy}) are given at each vertex of the triangle. This yields 18 independent conditions of the 21 required.

iii) The partial derivative of the function differentiated in the direction perpendicular (or normal) to each side of the triangle is a polynomial of degree three, at most, in the variable measured in the direction of the side of the triangle. This yields three additional conditions, one for each side of the triangle.

The third premise not only yields the necessary three additional conditions for solving the 21 coefficients, it also guarantees smoothness and continuity of the interpolated values, satisfying the third and fourth criteria in the following way.

A new coordinate system, called the **s-t** system, is introduced such that the **s**-axis is parallel to a side of the triangle. Since the coordinate transformation between the **x-y** system and the **s-t** system is linear, the values of z_x, z_y, z_{xx}, z_{xy}, and z_{yy} at each vertex uniquely determine the values of z_s, z_t, z_{ss}, z_{st}, and z_{tt} at the same vertex, each of the latter as a linear combination of the former. Then, the **z**, z_s, and z_{ss} values at two vertexes uniquely determine a fifth-degree polynomial in **s** for **z** on the side between these vertexes. Since two fifth-degree polynomials in **x** and **y** representing **z** values in two triangles that share the common side are reduced to fifth-degree polynomials in **s** on the side, these two polynomials in **x** and **y** coincide with each other on the common side. This establishes continuity of the interpolated **z** values along a side of a triangle. Similarly, the values of z_t and $z_{st} = (z_t)_s$ at two vertexes uniquely determine a third-degree polynomial in **s** for z_t on the side. Since the polynomial representing z_t is assumed to be third degree at most with respect to **s**, two polynomials representing z_t in two triangles that share the common side also coincide with each other on the side. This establishes continuity of z_t and thus smoothness of **z** along the side of the triangle.

Coordinate System Associated
With the Triangle

Reference was previously made to a transformed **s-t** coordinate system. Under the proposed method there are essentially five coordinate systems associated with each triangle. In the original **x-y** Cartesian coordinate system the vertexes of the triangle are denoted by V_1, V_2, and V_3 in a counter-clockwise order, and their respective coordinates by (x_1,y_1), (x_2,y_2), and (x_3,y_3) as shown in Figure 2-a. A new coordinate system, called the **u-v** system, is associated with each triangle where the vertexes are represented by (0,0), (1,0), and (0,1), as shown in Figure 2-b.

The coordinate transformation between the **x-y** and the **u-v** systems is represented by

$$x = a u + b v + x_0 ,$$
$$y = c u + d v + y_0 , \qquad (2\text{-}2)$$

where

$$a = x_2 - x_1 ,$$
$$b = x_3 - x_1 ,$$
$$c = y_2 - y_1 ,$$
$$d = y_3 - y_1 , \qquad (2\text{-}3)$$
$$x_0 = x_1 ,$$
$$y_0 = y_1 .$$

The inverse relation is

$$u = [\ d(x - x_0) - b(y - y_0)] / (ad - bc) ,$$
$$v = [- c(x - x_0) + a(y - y_0)] / (ad - bc) . \qquad (2\text{-}4)$$

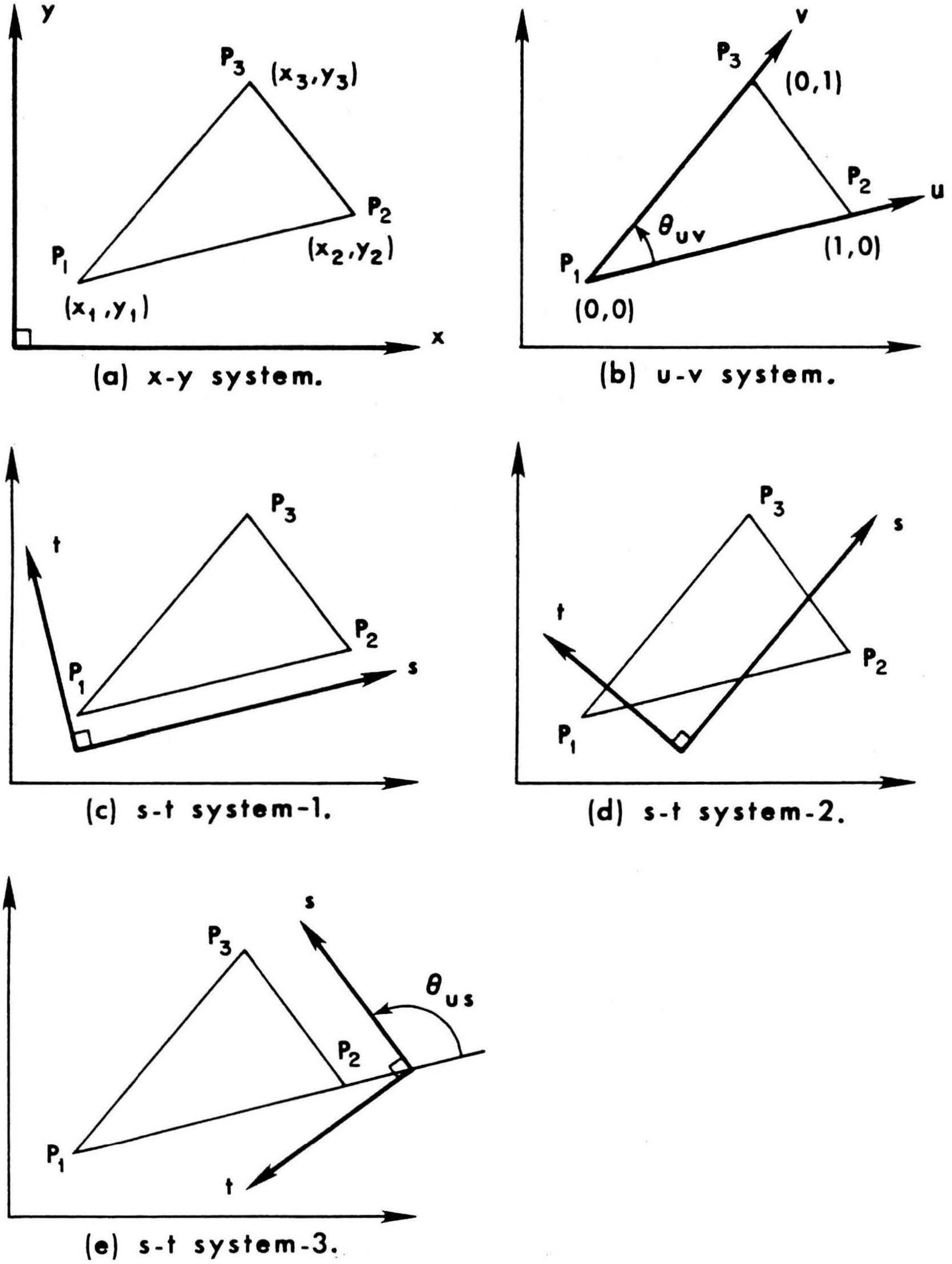


Figure 2 Various Coordinate Systems [2]

The transformation of the partial derivatives in the **x-y** system to the **u-v** system is given by

$$
\begin{aligned}
z_u &= a\,z_x + c\,z_y\,,\\
z_v &= b\,z_x + d\,z_y\,,\\
z_{uu} &= a^2\,z_{xx} + 2\,a\,c\,z_{xy} + c^2\,z_{yy}\,,\\
z_{uv} &= a\,b\,z_{xx} + (a\,d + b\,c)\,z_{xy} + c\,d\,z_{yy}\,,\\
z_{vv} &= b^2\,z_{xx} + 2\,b\,d\,z_{xy} + d^2\,z_{yy}\,.
\end{aligned}
\tag{2-5}
$$

Since this coordinate transformation is linear, the interpolating polynomial (2-1) is transformed to

$$
z(u,v) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} p_{jk}\, u^j\, v^k\,. \tag{2-6}
$$

The partial derivatives of **z**(u,v) in the **u-v** system are

$$
\begin{aligned}
z_u(u,v) &= \sum_{j=1}^{5} \sum_{k=0}^{5-j} j\, p_{jk}\, u^{j-1}\, v^k\,,\\
z_v(u,v) &= \sum_{j=0}^{4} \sum_{k=1}^{5-j} k\, p_{jk}\, u^j\, v^{k-1}\,,\\
z_{uu}(u,v) &= \sum_{j=2}^{5} \sum_{k=0}^{5-j} j\,(j-1)\, p_{jk}\, u^{j-2}\, v^k\,,\\
z_{uv}(u,v) &= \sum_{j=1}^{4} \sum_{k=1}^{5-j} j\,k\, p_{jk}\, u^{j-1}\, v^{k-1}\,,\\
z_{vv}(u,v) &= \sum_{j=0}^{3} \sum_{k=2}^{5-j} k\,(k-1)\, p_{jk}\, u^j\, v^{k-2}\,.
\end{aligned}
\tag{2-7}
$$

The lengths of the unit vectors in the **u-v** system (i.e., the lengths of sides $\overline{V_1V_2}$ and $\overline{V_1V_3}$) are denoted by L_u and L_v, respectively, and the angle between the **u** and **v** axes by θ_{uv}. They are given by

$$L_u = a^2 + c^2 ,$$
$$L_v = b^2 + d^2 , \quad (2\text{-}8)$$
$$\theta_{uv} = \tan^{-1}(d/b) - \tan^{-1}(c/a) ,$$

where **a, b, c,** and **d** are constants given in (2-3).

Implementation of the Third Premise

It follows from the third of the three premises mentioned previously that when the coordinate system is transformed to another Cartesian system, called the **s-t** system, in such a way that the **s** axis is parallel to each of the sides of the triangle, the bivariate polynomial in **s** and **t** representing the **z** values must satisfy

$$z_{tssss} = 0 . \quad (2\text{-}9)$$

By representing Equation (2-9) in the **u-v** system several equations can be derived for determining the coefficients of the polynomial.

First, consider the case where the **s** axis is parallel to side $\overline{V_1V_2}$ as shown in Figure 2-c. The coordinate transformation between the **u-v** and the **s-t** systems is expressed by

$$u = [(\sin\theta_{uv})(s - s_0) - (\cos\theta_{uv})(t - t_0)] / (L_u \sin\theta_{uv}) ,$$
$$v = (t - t_0) / (L_v \sin\theta_{uv}) , \quad (2\text{-}10)$$

where L_u, L_v, and θ_{uv} are constants given in (2-8). Partial derivatives with respect to **s** and **t** are expressed by

$$\frac{\partial}{\partial s} = \frac{1}{L_u}\frac{\partial}{\partial u},$$
$$\frac{\partial}{\partial t} = -\frac{\cos\theta_{uv}}{L_u \sin\theta_{uv}}\frac{\partial}{\partial u} + \frac{1}{L_v \sin\theta_{uv}}\frac{\partial}{\partial v}, \tag{2-11}$$

respectively. From (2-1), (2-6), and (2-11) it follows that

$$L_u P_{41} - 5 L_v \cos\theta_{uv} P_{50} = 0 . \tag{2-12}$$

Next, consider the case where the **s** axis is parallel to side $\overline{\mathbf{V_1V_3}}$ as shown in Figure 2-d. The coordinate transformation is expressed by

$$u = -(t - t_0)/(L_u \sin\theta_{uv}),$$
$$v = [(\sin\theta_{uv})(s - s_0) + (\cos\theta_{uv})(t - t_0)]/(L_v \sin\theta_{uv}) . \tag{2-13}$$

Partial derivatives are expressed by

$$\frac{\partial}{\partial s} = \frac{1}{L_v}\frac{\partial}{\partial v},$$
$$\frac{\partial}{\partial t} = -\frac{1}{L_u \sin\theta_{uv}}\frac{\partial}{\partial u} + \frac{\cos\theta_{uv}}{L_v \sin\theta_{uv}}\frac{\partial}{\partial v} . \tag{2-14}$$

Then, from (2-6), (2-9), and (2-14)

$$L_v P_{14} - 5 L_u \cos\theta_{uv} P_{05} = 0 . \tag{2-15}$$

Next, consider the third case where the **s** axis is parallel to side $\overline{\mathbf{V_2V_3}}$ as shown in Figure 2-e. The coordinate transformation is expressed by

$$u = A(s - s_0) + B(t - t_0),$$
$$v = C(s - s_0) + D(t - t_0), \tag{2-16}$$

where

$$A = \sin(\theta_{uv} - \theta_{us}) / (L_u \sin \theta_{uv}) ,$$

$$B = - \cos(\theta_{uv} - \theta_{us}) / (L_u \sin \theta_{uv}) ,$$

$$C = \sin \theta_{us} / (L_v \sin \theta_{uv}) , \quad (2\text{-}17)$$

$$D = \cos \theta_{us} / (L_v \sin \theta_{uv}) ,$$

$$\theta_{us} = \tan^{-1}[(d - c) / (b - a)] - \tan^{-1}(c / a) .$$

The constants **a**, **b**, **c**, and **d** are given in (2-3), and L_u, L_v, and θ_{uv} are given in (2-8). Partial derivatives with respect to **s** and **t** are expressed by

$$\frac{\partial}{\partial s} = A \frac{\partial}{\partial u} + C \frac{\partial}{\partial v} ,$$

$$\frac{\partial}{\partial t} = B \frac{\partial}{\partial u} + D \frac{\partial}{\partial v} . \quad (2\text{-}18)$$

From (2-6), (2-9), and (2-18),

$$5 A^4 B p_{50} + A^3 (4 B C + A D) p_{41} + A^2 C (3 B C + 2 A D) p_{32}$$

$$+ A C^2 (2 B C + 3 A D) p_{23} + C^3 (B C + 4 A D) p_{14} + 5 C^4 D p_{05} = 0 . \quad (2\text{-}19)$$

Equations (2-12), (2-15), and (2-19), the results of implementation of the third premise (2-9) in the **u-v** coordinate system, are used for determining the coefficients of the polynomial (2-6).

Determination of the Coefficients of the Polynomial

The coefficients of the lower-power terms can be determined by letting **u** = 0 and **v** = 0 and by inserting the values of $\mathbf{z}$, $\mathbf{z}_u$, $\mathbf{z}_v$, $\mathbf{z}_{uu}$, $\mathbf{z}_{uv}$, and $\mathbf{z}_{vv}$ at $\mathbf{V}_1$ (i.e., $\mathbf{u} = 0$ and $\mathbf{v} = 0$) in (2-6) and (2-7). The results are

$$\begin{aligned} p_{00} &= z(0,0) , \\ p_{10} &= z_u(0,0) , \\ p_{01} &= z_v(0,0) , \\ p_{20} &= z_{uu}(0,0) / 2 , \\ p_{11} &= z_{uv}(0,0) , \\ p_{02} &= z_{vv}(0,0)/2 . \end{aligned} \tag{2-20}$$

Note that the first equation of (2-20) guarantees that the interpolated surface passes through the data points as required by the first criterion since all terms of (2-6) except $\mathbf{P}_{00}$ go to zero at the vertex defined by the data point in question, where $\mathbf{u} = 0$ and $\mathbf{v} = 0$. Next, letting $\mathbf{u} = 1$ and $\mathbf{v} = 0$ and inserting the values of $\mathbf{z}$, $\mathbf{z}_u$, and $\mathbf{z}_{uu}$ at $\mathbf{V}_2$ (i.e., $\mathbf{u} = 1$ and $\mathbf{v} = 0$) in (2-6) and the first and third equations in (2-7), and solving for $\mathbf{P}_{30}$, $\mathbf{P}_{40}$, and $\mathbf{P}_{50}$,

$$\begin{aligned} p_{30} &= [20\,z(1,0) - 8\,z_u(1,0) + z_{uu}(1,0) - 20\,p_{00} - 12\,p_{10} - 6\,p_{20}]/2, \\ p_{40} &= -\,15\,z(1,0) + 7\,z_u(1,0) - z_{uu}(1,0) + 15\,p_{00} + 8\,p_{10} + 3\,p_{20}, \\ p_{50} &= [12\,z(1,0) - 6\,z_u(1,0) + z_{uu}(1,0) - 12\,p_{00} - 6\,p_{10} - 2\,p_{20}]/2. \end{aligned} \tag{2-21}$$

is derived in which $\mathbf{P}_{00}$, $\mathbf{P}_{10}$, and $\mathbf{P}_{20}$ are given by (2-20).

Similarly, using the values of $\mathbf{z}$, $\mathbf{z}_v$, and $\mathbf{z}_{vv}$ at $\mathbf{V}_3$ (i.e., $\mathbf{u} = 0$ and $\mathbf{v} = 1$)

$$\begin{aligned} p_{03} &= [20\,z(0,1) - 8\,z_v(0,1) + z_{vv}(0,1) - 20\,p_{00} - 12\,p_{01} - 6\,p_{02}]/2, \\ p_{04} &= -\,15\,z(0,1) + 7\,z_v(0,1) - z_{vv}(0,1) + 15\,p_{00} + 8\,p_{01} + 3\,p_{02}, \\ p_{05} &= [12\,z(0,1) - 6\,z_v(0,1) + z_{vv}(0,1) - 12\,p_{00} - 6\,p_{01} - 2\,p_{02}]/2 \end{aligned} \tag{2-22}$$

is derived from (2-6) and the second and last equations in (2-7).

From (2-12) and (2-15) the equations for $\mathbf{P}_{41}$ and $\mathbf{P}_{14}$ can be written as

$$
\begin{aligned}
p_{41} &= \frac{5\,L_v \cos\theta_{uv}}{L_u}\, p_{50}\,, \\
p_{14} &= \frac{5\,L_u \cos\theta_{uv}}{L_v}\, p_{05}\,.
\end{aligned}
\tag{2-23}
$$

where $\mathbf{P}_{50}$ and $\mathbf{P}_{05}$ are given by (2-21) and (2-22), respectively.

Next, using the values of $\mathbf{z}_v$ and $\mathbf{z}_{uv}$ at $\mathbf{V}_2$ (i.e., $\mathbf{u}$ = 1 and $\mathbf{v}$ = 0) with the second and the fourth equations in (2-7) and solving for $\mathbf{P}_{21}$ and $\mathbf{P}_{31}$

$$
\begin{aligned}
p_{21} &= \ \ 3\,z_v(1,0) - z_{uv}(1,0) - 3\,p_{01} - 2\,p_{11} + \ p_{41}\,, \\
p_{31} &= -2\,z_v(1,0) + z_{uv}(1,0) + 2\,p_{01} + \ p_{11} - 2\,p_{41}\,.
\end{aligned}
\tag{2-24}
$$

Similarly, using the values of $\mathbf{z}_u$ and $\mathbf{z}_{uv}$ at $\mathbf{V}_3$ (i.e., $\mathbf{u}$ = 0 and $\mathbf{v}$ = 1) with the first and the fourth equations in (2-7)

$$
\begin{aligned}
p_{12} &= \ \ 3\,z_u(0,1) - z_{uv}(0,1) - 3\,p_{10} - 2\,p_{11} + \ p_{14}\,, \\
p_{13} &= -2\,z_u(0,1) + z_{uv}(0,1) + 2\,p_{10} + \ p_{11} - 2\,p_{14}\,.
\end{aligned}
\tag{2-25}
$$

Equation (2-19) can be rewritten as

$$
g_1\,p_{32} + g_2\,p_{23} = h_1\,, \tag{2-26}
$$

where

$$g_1 = A^2 C(3BC + 2AD),$$

$$g_2 = AC^2(2BC + 3AD),$$

$$h_1 = -5A^4 B p_{50} - A^3(4BC + AD)p_{41}$$

$$- C^3(BC + 4AD)p_{14} - 5C^4 D p_{05}, \qquad (2\text{-}27)$$

with **A**, **B**, **C**, and **D** defined by (2-17). From the value of $\mathbf{z}_{vv}$ at $\mathbf{V}_2$ and the last equation in (2-7),

$$p_{22} + p_{32} = h_2, \qquad (2\text{-}28)$$

where

$$h_2 = (1/2)z_{vv}(1,0) - p_{02} - p_{12}. \qquad (2\text{-}29)$$

Similarly, from the value of $\mathbf{z}_{uu}$ at $\mathbf{V}_3$ and the third equation in (2-7),

$$p_{22} + p_{23} = h_3, \qquad (2\text{-}30)$$

where

$$h_3 = (1/2)z_{uu}(0,1) - p_{20} - p_{21}. \qquad (2\text{-}31)$$

Solving (2-26), (2-28), and (2-30) with respect to $\mathbf{p}_{22}$, $\mathbf{p}_{32}$, and $\mathbf{p}_{23}$,

$$p_{22} = (g_1 h_2 + g_2 h_3 - h_1)/(g_1 + g_2),$$

$$p_{32} = h_2 - p_{22}, \qquad (2\text{-}32)$$

$$p_{23} = h_3 - p_{22},$$

with $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{h}_1$, $\mathbf{h}_2$, and $\mathbf{h}_3$ given by (2-27), (2-29), and (2-31).

Estimation of Partial Derivatives

The partial derivatives in the **x-y** coordinate system in (2-5) are calculated in two steps. To estimate the first-order partial derivatives at data point P_0, several additional data points $P_i(i=1,2,...,ncp)$, the projections of which are closest to the projection of P_0 selected from all data points, are used. The vector product of $\overline{P_0P_i}$ and $\overline{P_0P_j}$ is constructed where P_i and P_j are two of the ncp closest data points. This gives a vector that is perpendicular to both $\overline{P_0P_i}$ and $\overline{P_0P_j}$ with the right hand rule where P_i and P_j are taken such that the resulting vector product always points upward (i.e., the **z** component of the vector is always positive). The magnitude of the vector equals the area of the parallelogram defined by $\overline{P_0P_i}$ and $\overline{P_0P_j}$. Vector products for all possible combinations of $\overline{P_0P_i}$ and $\overline{P_0P_j}$ are constructed and summed vectorially, and the first-order derivatives z_x and z_y at P_0 are taken as those of a plane that is normal to the resultant vector sum thus composed. The second-order partial derivates are calculated in the second step by repeating the procedure described in the preceding paragraph with the computed values of z_x at $P_i(i = 0,1,2,...,ncp)$ and obtaining estimates of z_{xx} and z_{xy} at P_0, and with the values of z_y at $P_i(i = 0,1,2,...,ncp)$ and obtaining estimates of z_{yy} and $z_{xy} = (z_y)_x$. A simple arithmetic mean of two z_{xy} values thus estimated is used as the value for z_{xy} at P_0.

Number of Closest Points, NCP

In calculating the coefficients of the polynomial, the values of some specified number of closest data points, NCP, is used. There is no theory that dictates the optimum value of NCP to use, but rather it is somewhat dependent upon the distribution of the data points. Too small or too large of a value gives poor results. In addition, larger values require additional computer time. As this value must be specified by the user it is recommended that further testing and comparison be done to determine the optimum range of NCP values.

Step-by-Step Description of the Procedure

In summary, the coefficients of the polynomial are determined by the following steps:

i) Determine the coordinate transformation coefficients **a, b, c,** and **d** from (2-3).

ii) Calculate partial derivatives z_x, z_y, z_{xx}, z_{xy}, and z_{yy} in (2-5) by the method described above.

iii) Calculate partial derivatives z_u, z_v, z_{uu}, z_{uv}, and z_{vv} from (2-5).

iv) Calculate L_u, L_v, and θ_{uv} from (2-8).

v) Calculate the coordinate transformation coefficients **A, B, C,** and **D** from (2-17).

vi) Determine 18 coefficients of the polynomial from (2-20), (2-21), (2-22), (2-23), (2-24), and

(2-25)--in that order.

vii) Calculate g_1, g_2, h_1, h_2, and h_3 from (2-27), (2-29), and (2-31).

viii) Determine the remaining three coefficients from (2-32).

The **z** value for a given point (x,y) can be interpolated by the following steps:

i) Transform **x** and **y** to **u** and **v** by (2-4) with the necessary coefficients given by (2-3).

ii) Evaluate the polynomial for **z**(u,v) given in (2-6).

Limitations

As is true for any interpolation scheme a high degree of accuracy is not expected for extrapolation outside of the polygon defined by the set of data points. Under the proposed scheme, however, reasonable results can be expected for regions reasonably near the polygon. This is accomplished by defining a semi-infinite rectangle normal to each face of the polygon in Figure 1, semi-infinite in the sense that the rectangle is unbounded on the side away from the polygon as shown in Figure 3. This will also create semi-infinite triangles bounded by two semi-infinite rectangles, with the polygon and the semi-infinite triangle sharing a common vertex. In a semi-infinite rectangular area extrapolation is performed by a bivariate polynomial of the fifth-degree in the direction of the line segment of the polygon and of the second degree in the direction normal to the face of the polygon. In a semi-infinite triangular area extrapolation is performed by a bivariate second-degree polynomial that smoothly connects to the two polynomials in the two adjacent neighboring rectangles.

The algorithm conforms nicely to each of the five critical criteria set forth at the beginning of this chapter. It provides an excellent basis for graphically displaying sparse field data in a way that the data can be much more easily and efficiently analyzed and interpreted.

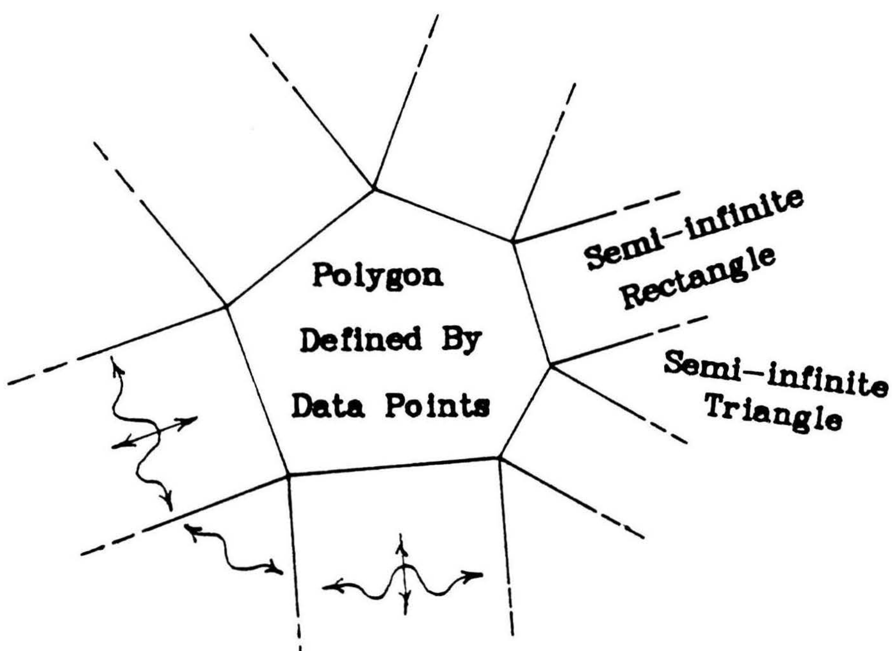


Figure 3 Partitioning of the Domain for Extrapolation

CHAPTER III

EXPERT

Once an interpolation algorithm was selected it was requisite to make the utilization of the algorithm as versatile as possible both in data entry and in the form of the results. EXPERT allows the user wide latitude in selecting the parameters governing the interpolation of the field data and in the form of the output, depending on the intended use. The program is general enough that any set of data, the projections of which lie in an **x-y** plane, can be displayed.

Entry of Field Data

The user can interactively input the field or experimental data by responding to prompts initiated by the program. If desired a disk file can be created and the data stored. Once the data is thus stored it can be used by EXPERT at a later time without the necessity of re-entering every value by hand. Data from any source could be accessed this way provided it is properly formatted. The format of the FORTRAN read statements is as follows:

```
READ(IDSK,*) NDP
READ(IDSK,*),(XD(N),YD(N),ZD(N),N=1,NDP)
```

where NDP is the number of data points, XD is an array containing the **x** coordinates, YD is an array containing the

y coordinates, and ZD is an array containing the **z** values of the data points.

A unique feature of EXPERT is the ability to read a series of such files, perform the interpolation, and save the results in the form of time histories at a point or as a sequence of surface files over an interval of time. This is especially applicable when the devices that record the responses from the field tests record the data in the form of a time history at a given location. From these time histories a series of files, each representing the whole surface, or **x-y** domain, at a given time step, can be created. These files are read sequentially and the desired interpolations performed.

Development of Interpolation Points

It is necessary to specify the coordinates of the points at which interpolation is to be performed. In this aspect, EXPERT provides the user considerable latitude. If there are only a few select points of interest the coordinates of these points can be used to define the interpolation points. This allows the user the capability of determining the interpolated value at specific points anywhere within the **x-y** domain.

An additional capability of EXPERT is to construct a regular rectangular grid as specified by the user and interpolate over the entire surface at the points defined by the intersections of the grid lines. This feature is useful when the domain under consideration is a fractional part of a larger surface, such as in the case of a soil profile.

A versatile feature is the ability to read files created by QMESH, a mesh generation program [5], and interpolate at the points defined by the mesh. This provides a virtually unlimited variety of configurations that can be used to define the points of interpolation. This feature is especially useful when the experimental data is obtained from gauges placed on a structural element or object. A mesh can be created which conforms to the geometry of the element under question, such as a cross section of a structural element.

Solutions

At the user's option the interpolated results can be handled in a number of ways. The results can be listed on the terminal, or stored in a disk file to permit printing or later reference. A highly useful option is to create a disk file of the results formatted for use with MOVIE.BYU, a computer graphics package [6]. With the aid of MOVIE.BYU the response function value at any given time step for every point on the domain can be displayed graphically with continuous tone color and/or contour lines. This can greatly aid in the interpretation of field data and in determining critical locations within the domain. This is probably the most useful capability of the program.

Time Histories

Another useful feature of EXPERT is the ability to create and display time histories. Time histories can be created from a series of multiple data files as previously described for any point in the domain. In addition, time history files can be created from HONDO.MOV files, the results from a finite element program entitled HONDO [7]. The two types of history files can be plotted together or separately and the field test results compared with the computed results. Any properly formatted history file from any source could be plotted.

The actual plots are created with the aid of a graphics subroutine package entitled DISSPLA [8]. These subroutines draw and label the axes, plot the curves, and create a legend.

Several of the capabilities of EXPERT are demonstrated graphically in the following chapter by comparing the results of an actual field test and the results of a finite element calculation of the same event. A listing of the FORTRAN code is found in Appendix A. The procedure involved in using EXPERT and the options available are explained in greater detail in the USER'S MANUAL found in Appendix B. A sample run is shown in Appendix C.

CHAPTER IV

Demonstration of the Capabilities of EXPERT

A buried high explosive ground shock test was conducted at the McCormick Ranch site near Albuquerque, New Mexico by the Air Force Weapons Laboratory. The experiment is referred to as CHEAT. A high explosive charge was detonated at a depth of 15 meters, and the soil response was monitored by gauges with the layout shown in Figure 4. The data obtained by these gauges was recorded in the form of time histories. Figure 5 shows the displacement and velocity curves from the gauge located at (3.00,-2.30). This is typical of the form of the data for each of the 23 gauge locations. To create the files to be used with EXPERT it was necessary to obtain the values from each graph for several timesteps. This was done with the aid of a Summagraphics Bit Pad One digitizer using the program COORD. COORD, listed in Appendix D, is supporting software written specifically for this task. Fifty-one files were created based on the velocity curves, one for each time step between t=0 milliseconds and t=150 milliseconds in increments of 3 milliseconds. Each file contains the **x** and **y** coordinates and the corresponding **z** values representing the response at a specific time for each of the 23 gauges. These files could have been created by taking the values directly from AFWL

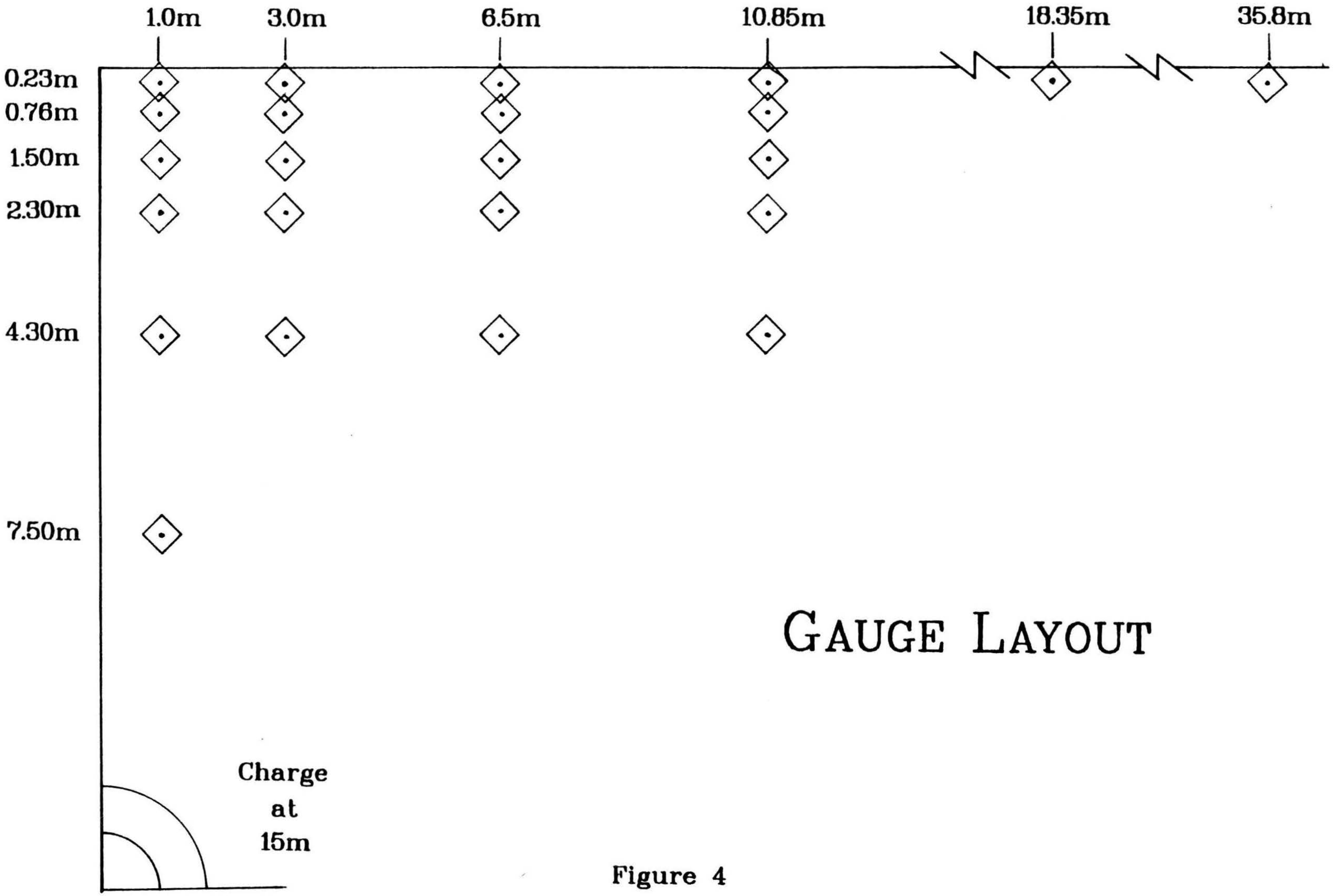

1.0m
3.0m
6.5m
10.85m
18.35m
35.8m
0.23m
0.76m
1.50m
2.30m
4.30m
7.50m
GAUGE LAYOUT
Charge
at
15m

Figure 4

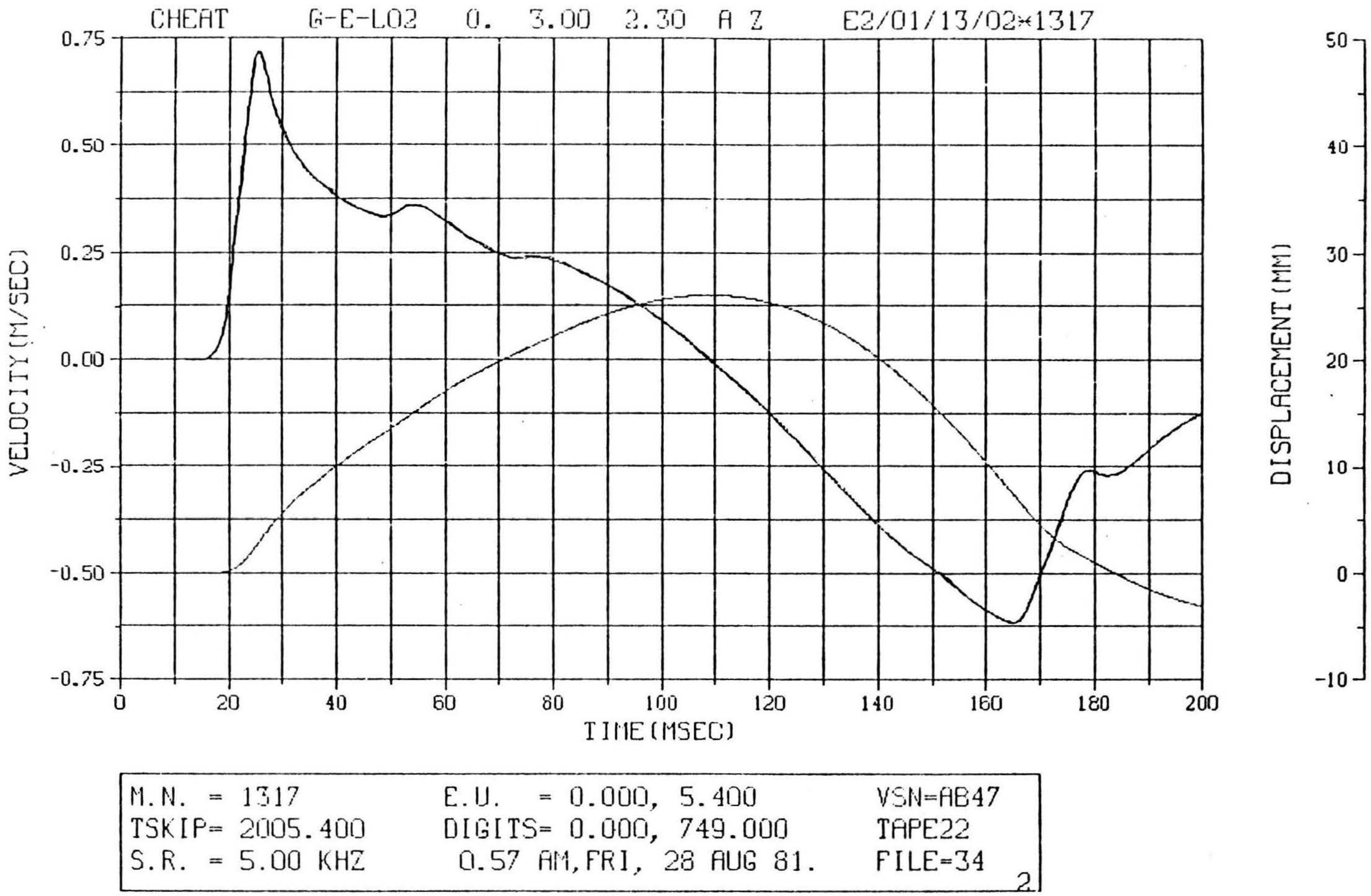


Figure 5 Typical Data Curve

Engineering Units tapes by utilizing the subroutines described in the User's Manual in Appendix B.

A finite element computer program entitled HONDO II was also used to analyze this problem. HONDO was developed to treat problems that involve extreme loads, and computes the time-dependent displacements, velocities, accelerations, and stresses within elastic or inelastic, two-dimensional or axisymmetric bodies of arbitrary shapes and materials. The soil profile is composed of three layers of soil of differing properties. The material properties used by HONDO for the analysis are listed in Appendix E. The mesh used is shown in Figure 6. The analysis by HONDO is explained in full detail by Benzley [9].

Using options available in EXPERT time histories were created for the point (4.913,-3.067) from both the interpolated CHEAT data and the HONDO analysis. The point (4.913,-3.067) was selected because it coincided with a point on the mesh used for the finite element analysis. Any mesh point could have been used for the comparison. It is instructive to note the position of the point, indicated by an * in Figure 7, relative to the location of the gauge points. One is not limited merely to the locations of the gauges, but can create time histories at any point within the **x-y** domain. This allows determination of a time history at critical points not monitored by a gauge. The plot comparing the two time histories is shown in Figure 8. The

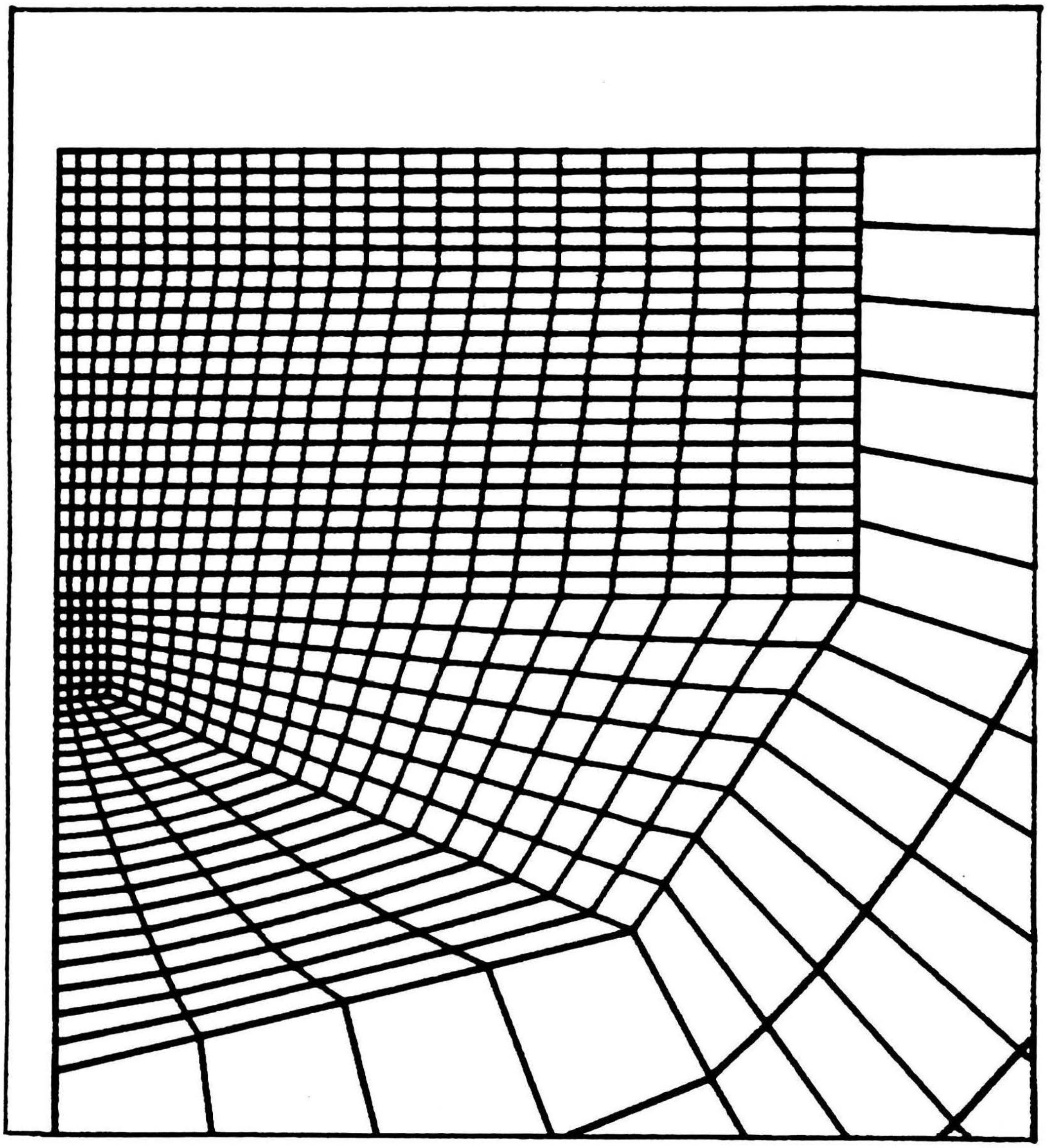


Figure 6 Near Field of Mesh Used in Finite Element Analysis

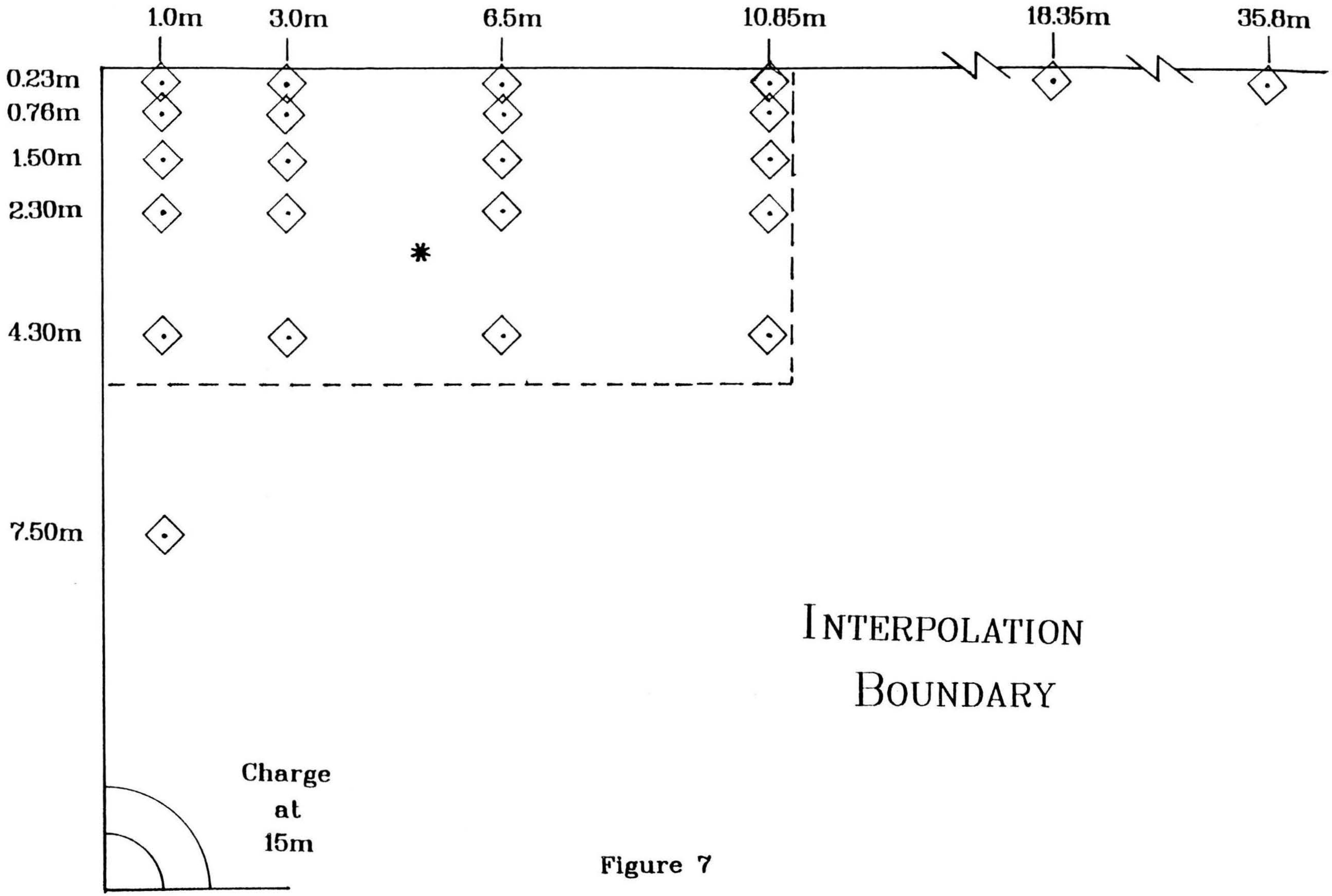

1.0m
3.0m
6.5m
10.85m
18.35m
35.8m
0.23m
0.76m
1.50m
2.30m
4.30m
7.50m
INTERPOLATION
BOUNDARY
Charge
at
15m

Figure 7

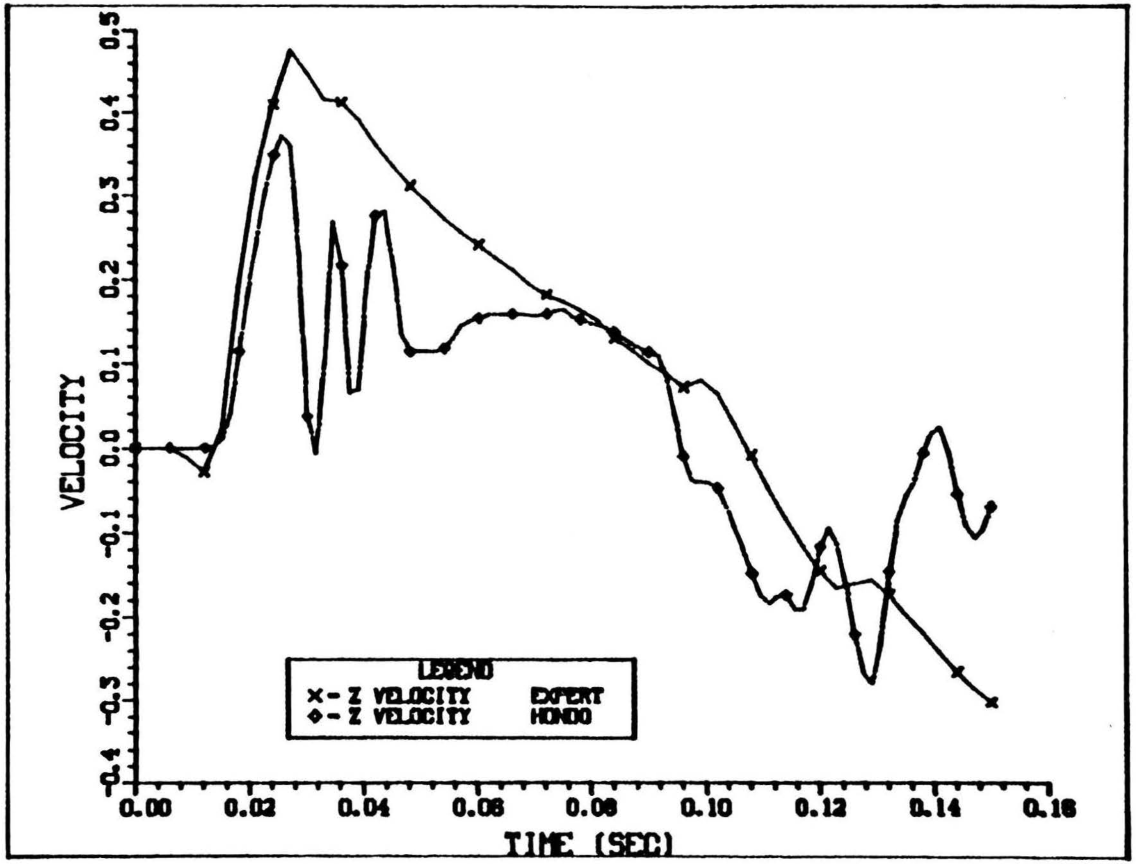


Figure 8 Typical Time History Comparison of EXPERT (field data) and HONDO (computed data)

curve based on the field data is the smooth curve, and the curve based on the HONDO calculation is the oscillating curve, typical of the results from finite element analyses. This capability permits a verification of the accuracy and adequacy of the material modeling in the finite element analysis and even of the finite element code itself.

A more graphic comparison is provided by the use of continuous tone color as shown in Figure 9. This picture represents the vertical velocity through the profile at time t=21 milliseconds. The top figure was created from the CHEAT field data. A rectangular grid was used as interpolation points with boundary dimensions of 11 meters wide by 5 meters deep as shown by the dashed line in Figure 7. The number of closest data points used in determining the coefficients of the polynomial was NCP=10. Several values of NCP ranging from 8 to 18 were tried with only minor variations in the results. The bottom figure was created from the HONDO calculation. The graphics were created on a Ramtek 9460 using MOVIE.BYU. The value of color fringes range from 0.0 m/sec for the blue to 0.5 m/sec for the red. The value of contour A is 0.1 m/sec, B is 0.2 m/sec, C is 0.3 m/sec, D is 0.4 m/sec, and E is 0.5 m/sec. This capability affords an excellent means of analyzing the soil response through the whole profile at a given instance of time and comparing directly computed with experimental results. Two additional capabilities of MOVIE.BYU to display

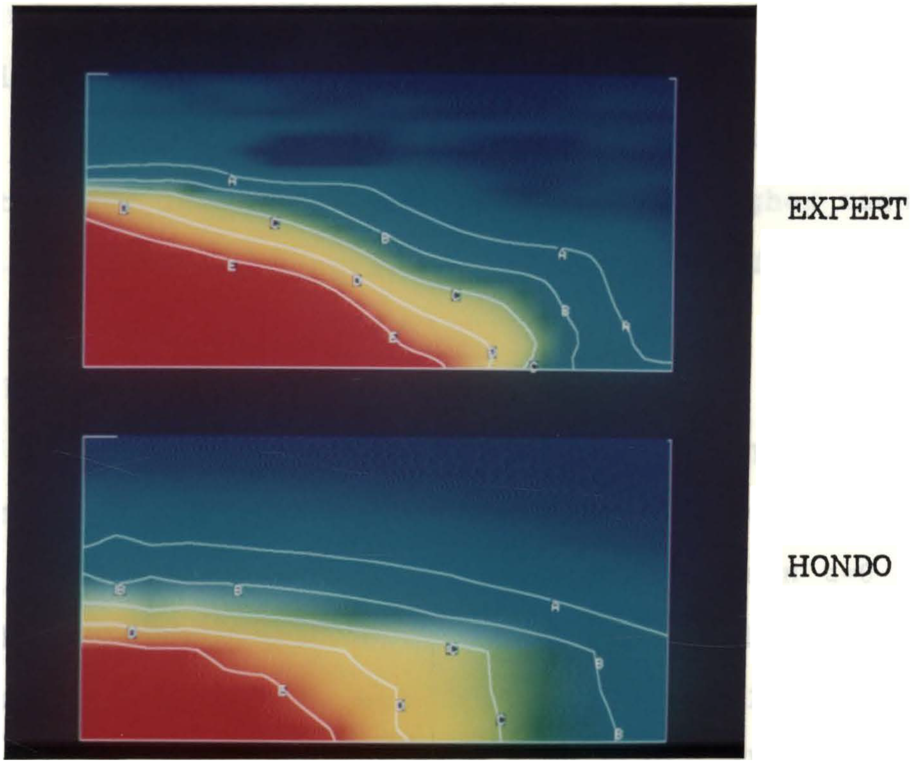


Figure 9 Comparison of EXPERT (field data) and HONDO (computed data) for Time t=21 msec Using Continuous Tone Color

the data comparison of Figure 9 are shown in Figures 10 and 11. Figure 10 uses simple line contour drawings and Figure 11 uses warped three dimensional surfaces. In each case the upper figure is from the CHEAT field data using EXPERT and the lower figure is from the finite element calculations using HONDO. For display purposes the warped surfaces in Figure 11 have been rotated as indicated by the coordinate triad. The 'peaks' represent the higher magnitudes of velocity and correspond to the red fringes in Figure 9.

The capabilities previously illustrated can be extended further. Interpolation based on the parameters stated in the previous paragraph was performed on the CHEAT data for each of the fifty-one timesteps between t=0 and t=150 milliseconds. The results for times t=15, 21, 27, 33, 39, and 45 milliseconds are shown in Figure 12. Similar results could be displayed for each time step. The value of the fringes range from 0.0 m/sec for the blue to 0.6 m/sec for the red. The wave propagation is graphically illustrated. The sequence indicates the arrival of the velocity wave at the surface of the earth. Not only is the value of the response function illustrated for every point in the profile, the whole profile is illustrated at every time step. By examining the sequence it is much easier to conceptualize what is occurring than by examining a collection of time histories at individual points.

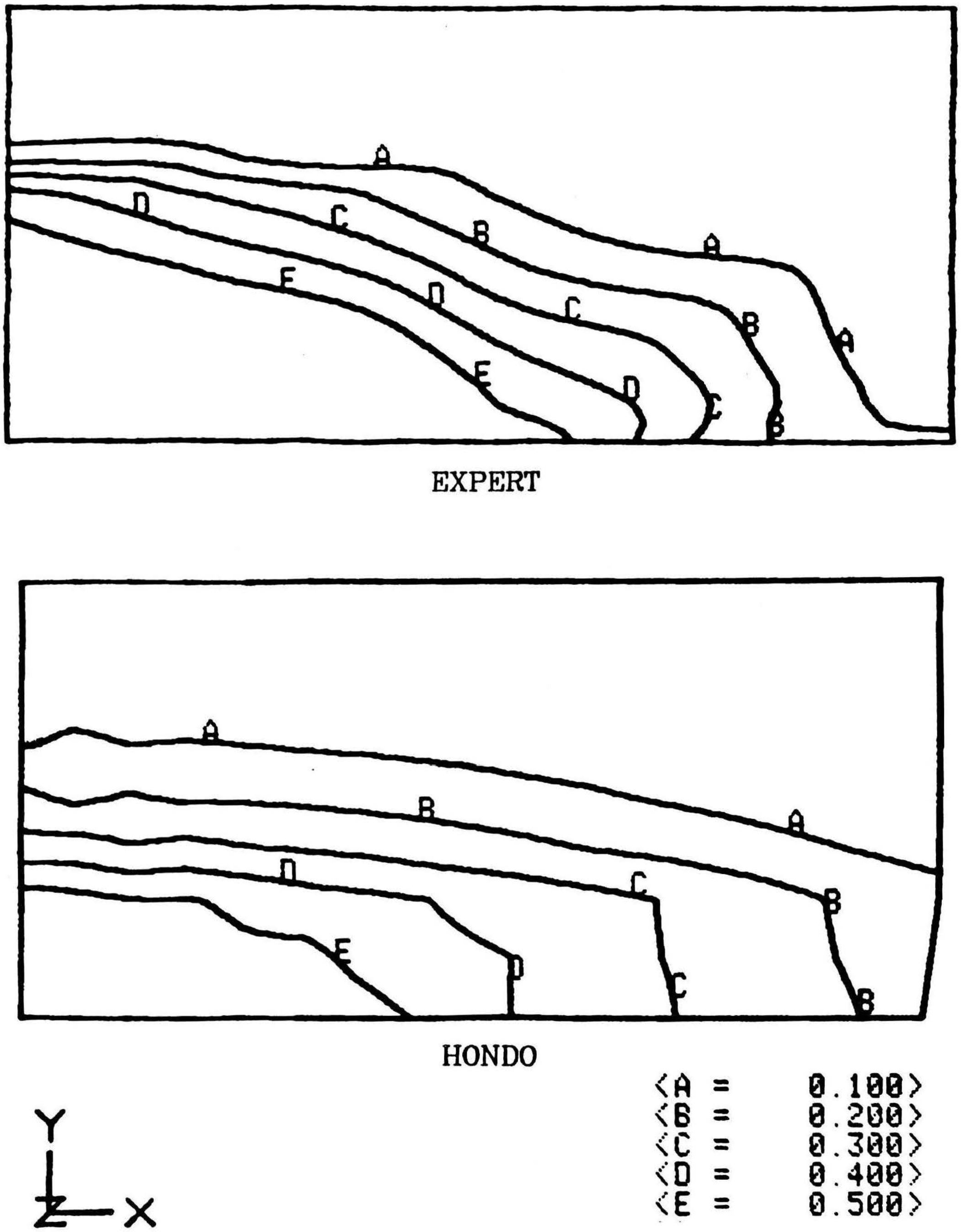


Figure 10 Comparison of EXPERT and HONDO for Time t=21 msec Using Simple Contour Lines

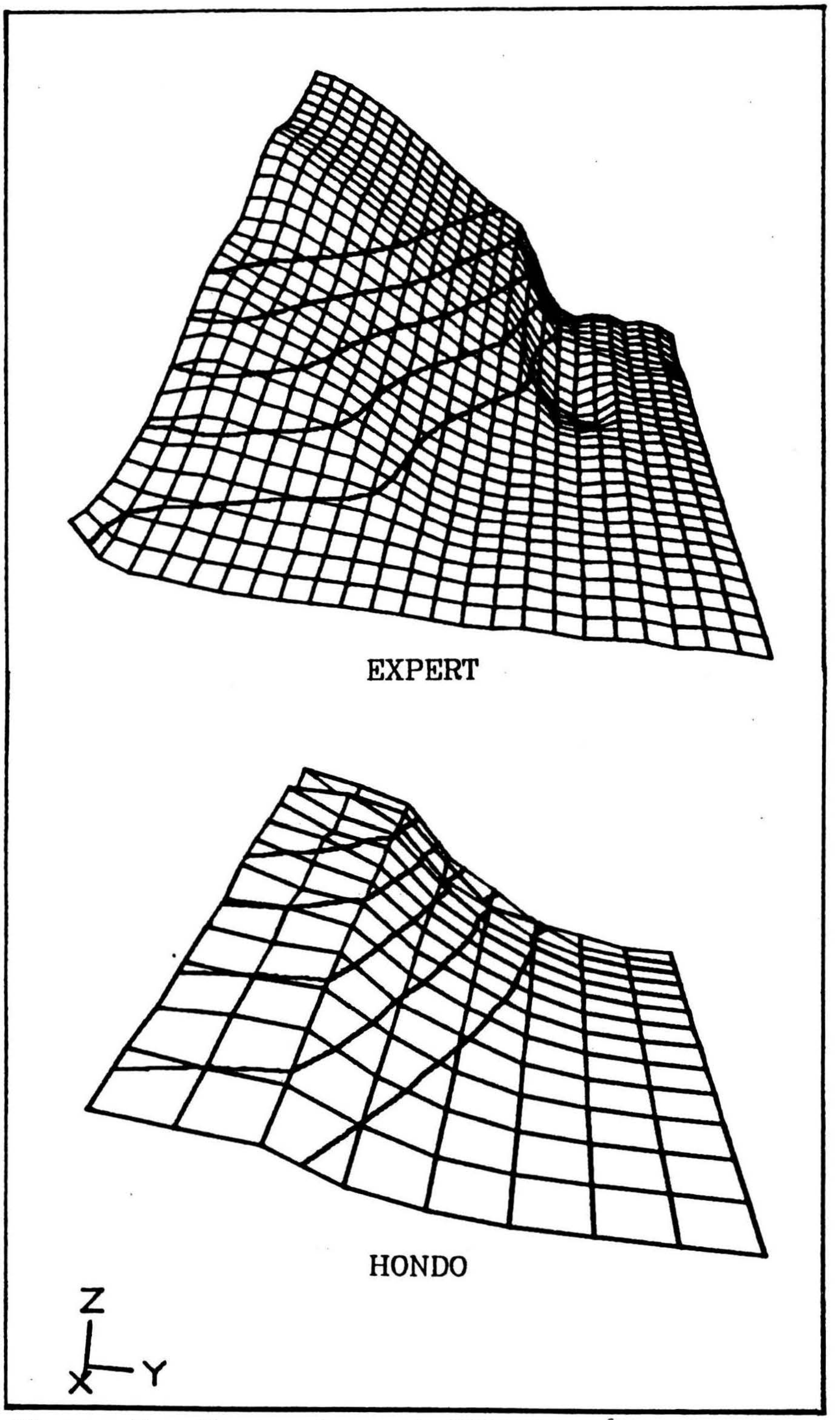
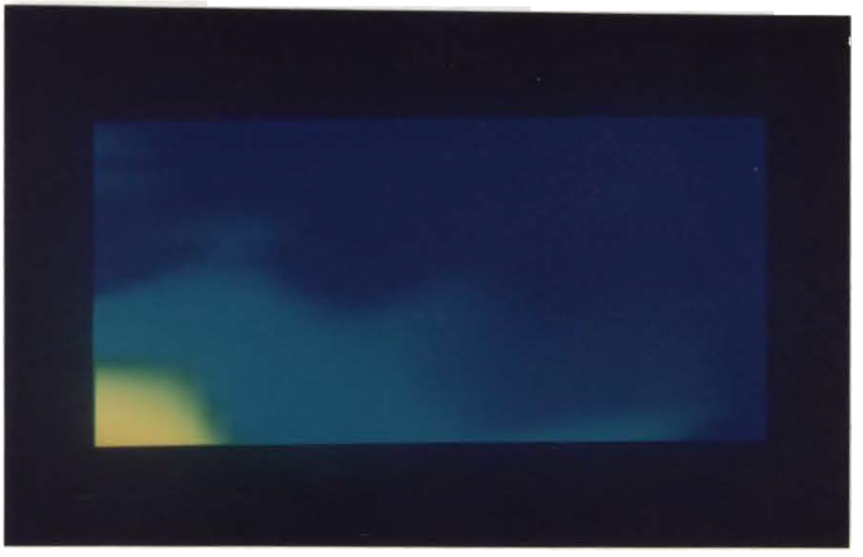
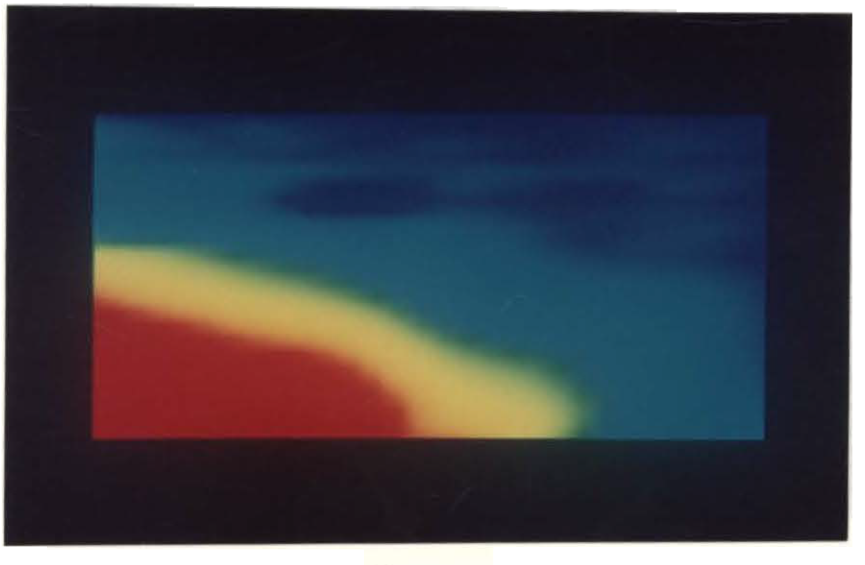


Figure 11 Comparison of EXPERT and HONDO for Time t=21 msec Using Warped Three Dimensional Surfaces

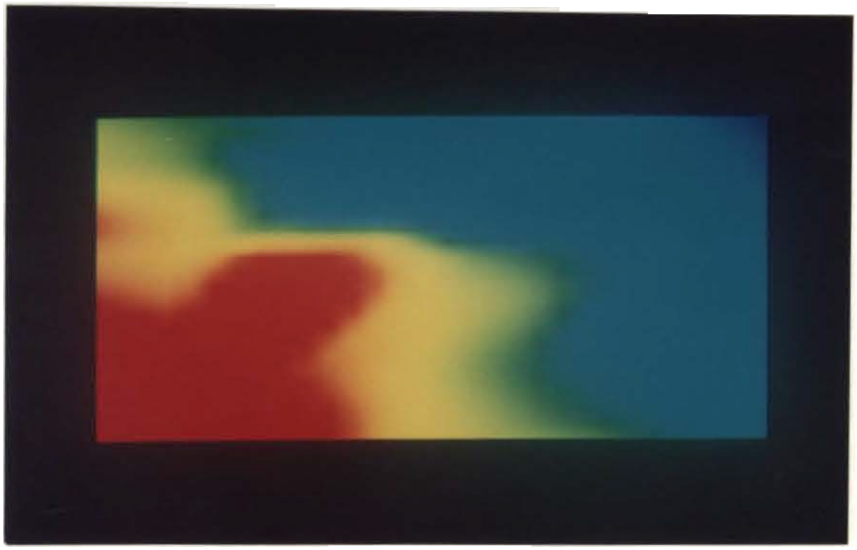


15 msec



21 msec

Figure 12 Sequence of CHEAT Field Data From EXPERT Using Continuous Tone Color



27 msec

33 msec

Figure 12 (continued)

39 msec

45 msec

Figure 12 (continued)

This presentation is representative of some of the tasks that can be performed by EXPERT. It illustrates how effectively field and test data can be displayed to aid in the analysis of the data. Any type of response function could be similarly displayed based on a field of sparse data.

CHAPTER V

Conclusions and Recommendations

A method of graphically displaying a set of experimental or field data has been developed and its usefulness demonstrated. Any planar set of "sparse" data can be used as the pivotal points for interpolation over the whole field to provide an adequate data base to employ computer graphics techniques. In addition, the response function value for any particular point within the field can be determined. Graphical display of the field data provides a convenient and effective means of analyzing the data. It enables one to determine what is occuring over the whole surface rather than at a few select points. EXPERT provides a powerful tool in the analysis of experimental and field test data.

It is recommended that EXPERT be utilized to display data from other sources, and not be limited to the graphic display of the type of ground shock problem described herein. There is a wide range of problems that can be similarly displayed. One application presently under consideration is the graphic display of data related to pulverized coal combustion reactors: temperature fields, compositions, mixture, mole fractions, etc. Other applications include the display of water quality sampling data over a lake surface or through a profile, topographical contours, and stress and strain gauge data. EXPERT is a highly versatile program that can be utilized to display almost any set of planar field or test data.

LIST OF REFERENCES

[1] Akima, Hiroshi. "Algorithm 526: Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points." ACM Transactions on Mathematical Software Vol. 4, No. 2 (June 1978): 160-164.

[2] U.S. Department of Commerce. Office of Telecommunications. A Method of Bivariate Interpolation and Smooth Surface Fitting for Values Given at Irregularly Distributed Points, by Hiroshi Akima. OT Report No. 75-70. Washington, D.C.: Government Printing Office, August 1975.

[3] Akima, Hiroshi. "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points." ACM Transactions on Mathematical Software Vol. 4, No. 2 (June 1978): 148-159.

[4] Lawson, C.L. Generation of a Triangular Grid with Application to Contour Plotting. Technical Memo. 299, Sect. 914. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology, [February 1972].

[5] Jones, R.E. Users Manual for QMESH, a Self-Organizing MESH Generation Program. SLA-74-0239. Albuquerque, New Mexico: Sandia Laboratories, [July 1974].

[6] Christiansen, H.N., and Stephenson, M.B. "MOVIE.BYU--A Computer Graphics Software System." Journal of the Technical Councils of ASCE Vol. 5, No. TCI (April 1979): 3-12.

[7] Key, S.W., Beisinger, Z.E., and Krieg, R.D. HONDO II--A Finite Element Computer Program for the Large Deformation Dynamic Response of Axisymmetric Solids. SAND78-0422. Albuquerque, New Mexico: Sandia National Laboratories, [October 1978].

[8] DISSPLA User's Manual. Version 8.2, Copyright 1970 by Integrated Software Systems Corporation (ISSCO). San Diego, CA: December 1978.

[9] Benzley, S.E. "Two and Three Dimensional Finite Element Calculations and Computer Graphic Displays for Ground Shock Problems." Technical Report for Air Force Weapons Laboratory/NTESG, (Brigham Young University, November 1981).

APPENDIXES

APPENDIX A

EXPERT Program Listing

```
      PROGRAM EXPERT(TTY,TAPE5=TTY)
C**********************************************************************C
C                                                                      C
C                     BRIGHAM YOUNG UNIVERSITY                         C
C                  DEPARTMENT OF CIVIL ENGINEERING                     C
C                                                                      C
C**********************************************************************C
C                                                                      C
C     PROGRAM EXPERT                                                   C
C                          VERSION 4.1                                 C
C                             CRAY                                     C
C                    RELEASED: 27 JUNE 1983                            C
C                                                                      C
C                   WRITTEN BY: L. ALLEN ADAMS                         C
C                                                                      C
C                                                                      C
C       TO ASK QUESTIONS CONCERNING THIS PROGRAM OR REPORT BUGS,       C
C       CONTACT :                                                      C
C                                                                      C
C                        STEVEN E. BENZLEY                             C
C                        CIVIL ENGINEERING                             C
C                        368 CB      B.Y.U.                            C
C                        PROVO, UTAH 84602                             C
C                        (801)  378 - 6322                             C
C                                                                      C
C                                                                      C
C**********************************************************************C
C                                                                      C
C                                                                      C
C     THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE  C
C     UNITED STATES GOVERNMENT.  NEITHER BRIGHAM YOUNG UNIVERSITY      C
C     NOR ANY OF ITS EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUBCON-  C
C     TRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR     C
C     IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR    C
C     THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,     C
C     APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS  C
C     USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.                   C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINES CALLED                                                  C
C     DISKN  =ASSIGNS LUN FILENAMES (CRAY VERSION)                     C
C     ENC    =ENCODES FILENAMES FOR MULTIPLE FILES                     C
C     EUTAPE =READS AFWL FORMATTED TAPES                               C
C     GRID   =CREATES GRID OF POINTS TO BE INTERPOLATED                C
C     HONTH  =CREATES TIME HISTORY FILES FROM HONDO.BYU FILES          C
C     IDBVIP =PERFORMS BIVARIATE INTERPOLATION                         C
C     IDSFFT =PERFORMS SURFACE FITTING INTERPOLATION                   C
C     MESH   =READS QMESH.BYU FILE TO USE AS POINTS OF INTERPOLATION   C
C     MOFILE =CREATES FILES OF INTERPOLATED RESULTS FORMATTED FOR      C
C             MOVIE.BYU                                                C
C     MULTI  =SETS PARAMETERS FOR READING MULTIPLE FILES               C
C     PRINT  =PRINTS INTERPOLATED RESULTS                              C
C     SELECT =ACCEPTS SELECTION FROM MENUS                             C
C     SELNCP =ACCEPTS SELECTION OF NCP                                 C
C     THIST  =PRODUCES TIME HISTORY PLOTS                              C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  IMPORTANT VARIABLES                                                 C
C     IFLAG1 =OPTION NUMBER FROM OPTION LIST #2--DEVELOPMENT OF INTER- C
```

```
C              POLATION POINTS                                          C
C            1=QMESH                                                    C
C            2=GRID                                                     C
C            3=RANDOM                                                   C
C     IFLAG2 =OPTION NUMBER FROM OPTION LIST #3--DISPOSAL OF SOLUTION   C
C            1=TTY                                                      C
C            2=DISK FILE                                                C
C            3=MOVIE.BYU FILES                                          C
C     IFLAG3 =INDICATES IF TIME HISTORIES ARE TO BE CREATED             C
C            0=NO TIME HISTORIES                                        C
C            1=TIME HISTORIES                                           C
C     IFLAG4 =INDICATES IF OTHER INTERPOLATIONS ARE TO BE PERFORMED     C
C            0=ONLY ONE TYPE                                            C
C            1=BOTH TIME-HISTORY AND OTHER INTERPOLATIONS               C
C     IFLAG5 =INDICATES REPEAT OF OPTION LIST #3                        C
C            0=FIRST TIME                                               C
C            1=REPEAT                                                   C
C     COORDX =ARRAY OF X COORDINATES OF POINTS FOR WHICH TIME HIS-      C
C              TORIES ARE MADE                                          C
C     COORDY =ARRAY OF Y COORDINATES OF POINTS FOR WHICH TIME HIS-      C
C              TORIES ARE MADE                                          C
C     COORDZ =ARRAY OF INTERPOLATED VALUES USED TO CREATE TIME HIS-     C
C              TORIES                                                   C
C     XD     =ARRAY OF X COORDINATES OF DATA POINTS                     C
C     YD     =ARRAY OF Y COORDINATES OF DATA POINTS                     C
C     ZD     =ARRAY CONTAINING THE DATA POINT VALUES                    C
C     XI     =ARRAY OF X COORDINATES OF INTERPOLATED POINTS             C
C     YI     =ARRAY OF Y COORDINATES OF INTERPOLATED POINTS             C
C     ZI     =ARRAY OF INTERPOLATED VALUES                              C
C     XIS    =ARRAY OF X COORDINATES OF DATA POINTS                     C
C     YIS    =ARRAY OF Y COORDINATES OF DATA POINTS                     C
C     ZIS    =ARRAY CONTAINING THE DATA POINT VALUES                    C
C     N1,N2,N3,N4 =ARRAYS CONTAINING THE QMESH CONNECTIVITY ARRAY       C
C                                                                       C
C     IWKBDM =MAX0(31,27+NCP)*NDP+NIP                                   C
C     IWKSDM =MAX0(31,27+NCP)*NDP+NXI*NYI                               C
C     NCPDFT =DEFAULT VALUE OF NCP                                      C
C     NDP    =NUMBER OF DATA POINTS                                     C
C     NHIST  =NUMBER OF HISTORY FILES CREATED FROM INTERPOLATED DATA    C
C     NIP    =NUMBER OF INTERPOLATION POINTS--IRREGULARLY DISTRIBUTED   C
C     NPT    =NUMBER OF POINTS IN QMESH FILE                            C
C     NUMFIL =NUMBER OF CONSECUTIVE FILES TO BE READ AND INTERPOLATED   C
C              (NUMBER OF HISTORY TIME STEPS)                           C
C     NXI    =NUMBER OF INTERPOLATION GRID POINTS IN X COORDINATE       C
C     NYI    =NUMBER OF INTERPOLATION GRID POINTS IN Y COORDINATE       C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SYSTEM DEPENDENT VARIABLES                                           C
C     IDSK   =UNIT NUMBER OF DISK                                       C
C     ILUN   =UNIT NUMBER TO READ FROM TERMINAL                         C
C     LUN    =UNIT NUMBER TO WRITE ON TERMINAL                          C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  VARIABLE DIMENSIONS INFORMATION FOR EXPERT                           C
C     COORDX(NHIST), COORDY(NHIST), COORDZ(NHIST)                       C
C     HISTORY(NUMFIL,NHIST)                                             C
C     IWK(MAX0(IWKBDM,IWKSDM))                                          C
C     KOMT(DIMENSION OF COMMENT LINE IN QMESH FILE)                     C
C     N1(NPT),N2(NPT),N3(NPT),N4(NPT)                                   C
C     T(NUMFIL)                                                         C
C     THNAME(NUMFIL)                                                    C
C     WK(8*NDP)                                                         C
C     XI(MAX0(NXI,NIP))                                                 C
C     YI(MAX0(NYI,NIP))                                                 C
```

```
C       ZI(MAX0(NXI*NYI,NIP))                                           C
C       XIS(NIP), YIS(NIP), ZIS(NIP)                                    C
C                                                                       C
C***********************************************************************C
C                                                                       C
C       SYSTEM ROUTINES CALLED                                          C
C                                                                       C
C          DISSPLA                                                      C
C          ASSIGN, RENAME (CRAY VERSION)                                C
C                                                                       C
C***********************************************************************C
C                                                                       C
C       NOTE:  THE CRAY VERSION USES A FORMAT OF "A8" FOR ALL FILENAME  C
C              READ AND WRITE STATEMENTS.  THE VAX VERSION USES A FORMAT C
C              OF "A12".  REPLACE ALL FORMAT STATEMENTS ACCORDINGLY.    C
C                                                                       C
C***********************************************************************C
C                                                                       C
        DIMENSION XD(30),YD(30),ZD(30),
     *            XI(1000),YI(1000),ZI(1000),
     *            XIS(30),YIS(30),ZIS(30),
     *            COORDX(30),COORDY(30),COORDZ(30),
     *            HISTORY(100,30),THNAME(30),T(800),
     *            IWK(2641),WK(240),
     *            N1(1000),N2(1000),N3(1000),N4(1000),
     *            KOMT(8)
        COMMON/DATA/XD,YD,ZD,XI,YI,ZI,XIS,YIS,ZIS
        COMMON/UNIT/ILUN,LUN,IDSK
        COMMON NAMOLD                                                   CRAY
        DATA NCPDFT/12/
        DATA ILUN/5/
CVAX    DATA LUN/6/                                                     VAX
        DATA LUN/5/                                                     CRAY
CVAX    DATA IDSK/2/                                                    VAX
CVAX    CHARACTER*12 DNAME,QNAME,THNAME,XNAME,TYPE                      VAX
CVAX    CHARACTER*1 DRCTN,LETTER                                        VAX
        INTEGER DNAME,QNAME,THNAME,XNAME,TYPE                           CRAY
        INTEGER ANS1,ANS2,ANS3,ANS4,DRCTN                               CRAY
        LOGICAL MULTPL
        NAMOLD='DUMMY'                                                  CRAY
        IDSK=NAMOLD                                                     CRAY
        CALL ASSIGN(-1,IDSK,0)                                          CRAY
        WRITE(LUN,10)
  10    FORMAT(' ',//
     * T9'**********************************************************'/
     * T9'*                                                        *'/
     * T9'*  EEEEEEE  X     X  PPPPPP   EEEEEEE  RRRRRR  TTTTTTTTT  *'/
     * T9'*  E         X   X   P     P  E        R     R     T      *'/
     * T9'*  E          X X    P     P  E        R     R     T      *'/
     * T9'*  EEEEE       XX    PPPPPP   EEEEE    RRRRRR      T      *'/
     * T9'*  E          X X    P        E        R   R       T      *'/
     * T9'*  E         X   X   P        E        R    R      T      *'/
     * T9'*  EEEEEEE  X     X  P        EEEEEEE  R     R     T      *'/
     * T9'*                                                        *'/
     * T9'*       EXPERIMENTAL DATA INTERPOLATION PROGRAM          *'/
     * T9'*                                                        *'/
     * T9'**********************************************************')

 20     WRITE(LUN,30)
 30     FORMAT(' ',///' INPUT OF EXPERIMENTAL DATA POINTS:'//
     *                '   OPTIONS:'/
     *                '      1.  INPUT DATA AND CREATE FILE'/
     *                '      2.  READ DATA FILE'/
     *                '      3.  READ MULTIPLE DATA FILES'/
     *                '   SPECIAL OPTIONS:'/
     *                '      4.  READ E.U. TAPE AND CREATE DATA FILES'/
```

```
     *                 '        5.  CREATE TIME HISTORIES FROM HONDO ',
     *                               'DATA FILE'/
     *                 '        6.  PLOT TIME HISTORIES'/
     *                 '        7.  HELP'/)
      NUM=7
      MENU=1
      CALL SELECT(NUM,MENU,NSEL)
      GO TO (70,110,240,40,50,60) NSEL
      NSEL=IABS(NSEL)
      GO TO (20,20,590,20) NSEL
 40   CALL EUTAPE
      GO TO 20
 50   CALL HONTH
      GO TO 20
 60   CALL THIST(NSELT)
      GO TO (20,590,20) NSELT
 70   WRITE(LUN,80)
 80   FORMAT('$','NUMBER OF DATA POINTS: ')
      READ(ILUN,*,ERR=70) NDP
      IF(NDP.LT.4) THEN
        WRITE(LUN,90)
 90     FORMAT(' ','ERROR. NDP MUST BE GREATER THAN OR EQUAL TO 4.'/)
        GO TO 70
      ENDIF
C
C  ENTER THE DATA
C
      WRITE(LUN,100)
100   FORMAT(' ','ENTER THE DATA (X,Y,Z):'/)
      READ(ILUN,*) (XD(N),YD(N),ZD(N),N=1,NDP)
      GO TO 140
C
C  READ DATA FROM FILE
C
110   WRITE(LUN,120)
120   FORMAT('$','ENTER THE FILENAME.EXT ')
      READ(ILUN,130) XNAME
130   FORMAT(A12)
      CALL DISKN(XNAME)                                                   CRAY
CVAX  OPEN(UNIT=IDSK,FILE=XNAME,STATUS='OLD')                            VAX
      READ(IDSK,*),NDP
      READ(IDSK,*) (XD(N),YD(N),ZD(N),N=1,NDP)
CVAX  CLOSE(UNIT=IDSK)                                                   VAX
C
C  ECHO THE DATA TO ALLOW VERIFICATION
C
140   WRITE(LUN,150)
150   FORMAT(' ','DPN',7X,'X',14X,'Y',14X,'Z',/)
      WRITE(LUN,160) (N,XD(N),YD(N),ZD(N),N=1,NDP)
160   FORMAT(0PI3,1P3E15.5)
      WRITE(LUN,170)
170   FORMAT('$','CORRECTIONS? ')
      READ(ILUN,180) ANS1
180   FORMAT(A1)
      IF(ANS1.EQ.1HY) THEN
        WRITE(LUN,190)
190     FORMAT('$','HOW MANY? ')
        READ(ILUN,*) NCORR
        WRITE(LUN,200)
200     FORMAT(' ','ENTER THE CORRECTIONS (DATA POINT ',
     *  'NUMBER,X-COORD,Y-COORD,Z-COORD):'/' DPN,X,Y,Z')
        READ(ILUN,*) (NN,XD(NN),YD(NN),ZD(NN),N=1,NCORR)
        GO TO 140
      ENDIF
C
C  STORE DATA ON DISK
```

```
C
      WRITE(LUN,210)
210   FORMAT('$','STORE THE DATA? ')
      READ(ILUN,180) ANS2
      IF(ANS2.NE.1HY) GO TO 230
      WRITE(LUN,220)
220   FORMAT('$','ASSIGN FILENAME: ')
      READ(ILUN,130) DNAME
      IF(DNAME.EQ.'          ') DNAME=XNAME
      CALL DISKN(DNAME)                                              CRAY
CVAX  OPEN(UNIT=IDSK,FILE=DNAME,STATUS='UNKNOWN')                    VAX
      WRITE(IDSK,*),NDP
      WRITE(IDSK,*) (XD(N),YD(N),ZD(N),N=1,NDP)
CVAX  CLOSE(UNIT=IDSK)                                               VAX
230   IF(ANS2.NE.1HY) DNAME=XNAME
      NUMFIL=1
      MULTPL=.FALSE.
      IFLAG3=0
      GO TO 250
C
C  SET PARAMETERS FOR MULTIPLE FILE READ
C
240   MULTPL=.TRUE.
      IFLAG1=0
      CALL MULTI(NUMFIL,LETTER,NCOUNT,INCR,NHIST,COORDX,COORDY,THNAME,
     *           IFLAG3,IFLAG4,TYPE,DRCTN)
      NCNT=NCOUNT
      IF(IFLAG4.EQ.0) GO TO 350
250   WRITE(LUN,260)
260   FORMAT(' ',///' DEVELOPMENT OF INTERPOLATION POINTS:'/
     *            '        (POINTS AT WHICH INTERPOLATION WILL BE ',
     *                     'PERFORMED)'//
     *            '    OPTIONS:'/
     *            '        1.  READ QMESH DATA FILE'/
     *            '        2.  CREATE A REGULAR RECTANGULAR GRID'/
     *            '        3.  SPECIFY RANDOM POINTS TO BE ',
     *                         'INTERPOLATED'/
     *            '        4.  HELP')
      NUM=4
      MENU=2
      CALL SELECT(NUM,MENU,NSEL)
      IFLAG1=NSEL
      GO TO (270,280,290) NSEL
      NSEL=IABS(NSEL)
      GO TO (250,20,590,20) NSEL
270   CALL MESH(KOMT,XI,YI,N1,N2,N3,N4,NPT,NIP,QNAME)
      GO TO 320
280   CALL GRID(NXI,NYI,XI,YI)
      GO TO 320
C
C  SPECIFY RANDOM POINTS
C
290   WRITE(LUN,300)
300   FORMAT('$','HOW MANY POINTS? ')
      READ(ILUN,*) NIP
      WRITE(LUN,310)
310   FORMAT(' ','INPUT THE COORDINATES (X-COORD,Y-COORD)'/' X,Y')
      READ(ILUN,*) (XIS(N),YIS(N),N=1,NIP)
320   IFLAG5=0
330   NCOUNT=NCNT
      WRITE(LUN,340)
340   FORMAT(' ',///' SOLUTION:'/
     *            '    OPTIONS:'/
     *            '        1.  PRINT RESULTS ON TTY'/
     *            '        2.  CREATE DISK FILE OF RESULTS'/
     *            '        3.  CREATE MOVIE.BYU FILES'/
```

```
     *                   '         4.  1 & 2'/
     *                   '         5.  1 & 3'/
     *                   '         6.  2 & 3'/
     *                   '         7.  1,2, & 3'/
     *                   '         8.  CONTINUE'/
     *                   '         9.  HELP')
      NUM=9
      MENU=3
      CALL SELECT(NUM,MENU,NSEL)
      IFLAG2=NSEL
      IF(IFLAG2.EQ.8) GO TO 550
      NSEL=-NSEL
      GO TO (320,250,590,20) NSEL
      IF(IFLAG5.EQ.1) GO TO 400
350   DO 490 I=1,NUMFIL
         MD=1
         NUMBER=I
         IF(MULTPL) THEN
            CALL ENC(LETTER,NCOUNT,DNAME)
            CALL DISKN(DNAME)                                       CRAY
CVAX        OPEN(UNIT=IDSK,FILE=DNAME,STATUS='OLD')                 VAX
            READ(IDSK,*) NDP
            READ(IDSK,*) (XD(N),YD(N),ZD(N),N=1,NDP)
CVAX        CLOSE(UNIT=IDSK)                                        VAX
         ENDIF
C
C  SPECIFY NCP
C
         IF(NUMBER.EQ.1) THEN
            MENU=4
            CALL SELNCP(MENU,NDP,NCPDFT,NCP)
            IF(NCP.EQ.-1) GO TO 590
            WRITE(LUN,370) NCP
370         FORMAT(' ',//,' THE VALUE OF NCP IS ',I2,'  <------')
         ENDIF
         IERR=0
         IWKBDM=MAX0(31,27+NCP)*NDP+NIP
         IWKSDM=MAX0(31,27+NCP)*NDP+NXI*NYI
C
C  INTERPOLATE AT QMESH POINTS
C
         IF(IFLAG1.EQ.1) CALL IDBVIP(MD,NCP,NDP,XD,YD,ZD,NIP,XI,YI,ZI,
     *        IWK,IWKBDM,WK,IERR)
C
C  INTERPOLATE AT GRID POINTS
C
         IF(IFLAG1.EQ.2) CALL IDSFFT(MD,NCP,NDP,XD,YD,ZD,NXI,NYI,XI,YI,
     *        ZI,IWK,IWKSDM,WK,IERR)
C
C  INTERPOLATE AT SPECIFIED RANDOM POINTS
C
         IF(IFLAG1.EQ.3) CALL IDBVIP(MD,NCP,NDP,XD,YD,ZD,NIP,XIS,YIS,
     *        ZIS,IWK,IWKBDM,WK,IERR)
C
C  INTERPOLATE AT TIME HISTORY POINTS
C
         IF(IFLAG3.EQ.1) THEN
            IF(IFLAG4.EQ.1) MD=2
            CALL IDBVIP(MD,NCP,NDP,XD,YD,ZD,NHIST,COORDX,COORDY,COORDZ,
     *                 IWK,IWKBDM,WK,IERR)
            DO 380 J=1,NHIST
               HISTORY(I,J)=COORDZ(J)
380         CONTINUE
            T(I)=FLOAT(NCOUNT)/1000.
            IF(IFLAG4.EQ.0) IFLAG2=8
         ENDIF
```

```
C
C  ERROR ENCOUNTERED IN INTERPOLATION SUBROUTINES
C
         IF(IERR.EQ.1) THEN
            WRITE(LUN,390)
390         FORMAT(' ',' ERROR ENCOUNTERED IN INTERPOLATION SUBROU',
     *                 'TINES.'/
     *                '     THIS RUN HAS BEEN ABORTED.  MAKE NECESSARY'/
     *                '          ADJUSTMENTS AND RE-ENTER THE DATA.')
            GO TO 20
         ENDIF
400      IUNIT=IDSK
C
C  PRINT RESULTS AND/OR CREATE FILES
C
         GO TO (410,420,460,470,440,450,430,480) IFLAG2
410      IUNIT=LUN
420      CALL PRINT(LETTER,NCOUNT,INCR,IFLAG1,IUNIT,NDP,NCP,NIP,NXI,
     *      NYI,DNAME,QNAME,KOMT,MULTPL)
         GO TO 480
430      CALL PRINT(LETTER,NCOUNT,INCR,IFLAG1,IUNIT,NDP,NCP,NIP,NXI,
     *      NYI,DNAME,QNAME,KOMT,MULTPL)
440      IUNIT=LUN
450      CALL PRINT(LETTER,NCOUNT,INCR,IFLAG1,IUNIT,NDP,NCP,NIP,NXI,
     *      NYI,DNAME,QNAME,KOMT,MULTPL)
460      CALL MOFILE(IFLAG1,NXI,NYI,XI,YI,ZI,NPT,NIP,N1,N2,N3,N4,
     *      MULTPL,NCOUNT,NUMBER)
         GO TO 480
470      CALL PRINT(LETTER,NCOUNT,INCR,IFLAG1,IUNIT,NDP,NCP,NIP,NXI,
     *      NYI,DNAME,QNAME,KOMT,MULTPL)
         IUNIT=LUN
         CALL PRINT(LETTER,NCOUNT,INCR,IFLAG1,IUNIT,NDP,NCP,NIP,NXI,
     *      NYI,DNAME,QNAME,KOMT,MULTPL)
480      NCOUNT=NCOUNT+INCR
490   CONTINUE
C
C  CREATE TIME HISTORY FILES
C
      IF(IFLAG3.EQ.1) THEN
         DO 530 J=1,NHIST
            CALL DISKN(THNAME(J))                                  CRAY
CVAX        OPEN(UNIT=IDSK,FILE=THNAME(J),STATUS='NEW')            VAX
            WRITE(IDSK,500) THNAME(J)
            WRITE(IDSK,500) TYPE
            WRITE(IDSK,510) DRCTN
            WRITE(IDSK,520) NUMFIL
            WRITE(IDSK,*) (HISTORY(I,J),I=1,NUMFIL)
            WRITE(IDSK,*) (T(I),I=1,NUMFIL)
500         FORMAT(1X,A12)
510         FORMAT(1X,A1)
520         FORMAT(I5)
CVAX        CLOSE(UNIT=IDSK)                                       VAX
530      CONTINUE
      ENDIF
      WRITE(LUN,540)
540   FORMAT(' ',//' COMMAND COMPLETED')
C
C  IF NOT MULTIPLE FILES GO TO PREVIOUS OPTION LIST
C
      IF(.NOT.MULTPL) THEN
         IFLAG5=1
         GO TO 330
      ENDIF
550   WRITE(LUN,560)
560   FORMAT(' '///'  OPTIONS:'/
     *       '        1.  CREATE TIME HISTORIES FROM HONDO ',
```

```
     *                               'DATA FILES'/
     *              '         2.  PLOT TIME HISTORIES'/
     *              '         3.  SPECIFY NEW INTERPOLATION POINTS'/
     *              '         4.  DEFINE NEW PROBLEM (START)'/
     *              '         5.  TERMINATE'/
     *              '         6.  HELP')
      NUM=6
      MENU=5
      CALL SELECT(NUM,MENU,NSEL)
      GO TO (570,580,250,20,590) NSEL
      NSEL=IABS(NSEL)
      GO TO (550,330,590,20) NSEL
570   CALL HONTH
      GO TO 550
580   CALL THIST(NSELT)
      GO TO (550,590,20) NSELT
590   CONTINUE
      STOP
      END

CSSET SUBROUTINE CLOSE                                                   SSET
C                                                                       CSSET
C***********************************************************************CSSET
C                                                                       CSSET
C  SUBROUTINE CLOSE-CLOSES DISK ON THE SUPERSET                         CSSET
C                                                                       CSSET
C***********************************************************************CSSET
C                                                                       CSSET
C  SUBROUTINE CALLED BY THIST                                           CSSET
C                                                                       CSSET
C***********************************************************************CSSET
C                                                                       CSSET
C  SYSTEM SUBROUTINES CALLED: OPSYSTEM                                  CSSET
C                                                                       CSSET
C***********************************************************************CSSET
C                                                                       CSSET
CSSET CALL OPSYSTEM('#DEACTIVATE U=!BYU',18)                             SSET
CSSET RETURN                                                             SSET
CSSET END                                                                SSET

      SUBROUTINE DISKN                                                   CRAY
C                                                                       CCRAY
C***********************************************************************CCRAY
C                                                                       CCRAY
C  SUBROUTINE DISKN-ASSIGNS THE FILENAME TO DISK FILES FOR THE CRAY     CCRAY
C                  VERSION                                              CCRAY
C                                                                       CCRAY
C***********************************************************************CCRAY
C                                                                       CCRAY
C  SUBROUTINE CALLED BY ALL ROUTINES WHICH READ OR WRITE A DISK FILE    CCRAY
C                                                                       CCRAY
C***********************************************************************CCRAY
C                                                                       CCRAY
C  SYSTEM ROUTINES CALLED: RENAME                                       CCRAY
C                                                                       CCRAY
C***********************************************************************CCRAY
C                                                                       CCRAY
C  IMPORTANT VARIABLES:                                                 CCRAY
C     NAMNEW=NEW NAME TO BE ASSIGNED TO THE FILE OR THE FILE NAME OF    CCRAY
C            A FILE TO BE READ                                          CCRAY
C     NAMOLD=OLD NAME OF FILE PREVIOUSLY WRITTEN TO OR READ FROM        CCRAY
C     DUMMY =DUMMY FILE NAME USED AS AN INTERMEDIATE WHEN A FILE IS     CCRAY
C            FIRST READ FROM AND THEN WRITTEN TO (WHEN NAMNEW =         CCRAY
C            NAMOLD)                                                    CCRAY
```

```
C                                                                       CCRAY
C***********************************************************************CCRAY
C                                                                       CCRAY
      COMMON/UNIT/ILUN,LUN,IDSK                                          CRAY
      COMMON NAMOLD                                                      CRAY
      INTEGER DUMMY                                                      CRAY
      IF(NAMNEW.EQ.NAMOLD) THEN                                          CRAY
         DUMMY='DUMMY'                                                   CRAY
         CALL RENAME(DUMMY,NAMOLD)                                       CRAY
         NAMOLD=DUMMY                                                    CRAY
      ENDIF                                                              CRAY
      IDSK=NAMNEW                                                        CRAY
      CALL RENAME(NAMNEW,NAMOLD)                                         CRAY
      NAMOLD=NAMNEW                                                      CRAY
      RETURN                                                             CRAY
      END                                                                CRAY


      SUBROUTINE ENC(LETTER,NUMBER,NAME)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE ENC-ENCODES THE LETTER AND NUMBER SPECIFIERS TO FORM      C
C                 FILENAMES                                             C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE CALLED BY: EXPERT, MOFILE, PRINTG, PRINTQ, PRINTS         C
C                                                                       C
C***********************************************************************C
C                                                                       C
CVAX  CHARACTER*12 NAME                                                  VAX
CVAX  CHARACTER*1 LETTER                                                 VAX
CVAX  DATA NCODE/12/                                                     VAX
      DATA NCODE/8/                                                      CRAY
      IF(NUMBER.GE.10000) THEN
        ENCODE(NCODE,10,NAME) LETTER,NUMBER
 10     FORMAT(A1,I5)
        GO TO 60
      ENDIF
      IF(NUMBER.GE.1000) THEN
        ENCODE(NCODE,20,NAME) LETTER,NUMBER
 20     FORMAT(A1,I4)
        GO TO 60
      ENDIF
      IF(NUMBER.GE.100) THEN
        ENCODE(NCODE,30,NAME) LETTER,NUMBER
 30     FORMAT(A1,I3)
        GO TO 60
      ENDIF
      IF(NUMBER.GE.10) THEN
        ENCODE(NCODE,40,NAME) LETTER,NUMBER
 40     FORMAT(A1,I2)
        GO TO 60
      ENDIF
      ENCODE(NCODE,50,NAME) LETTER,NUMBER
 50   FORMAT(A1,I1)
 60   CONTINUE
      RETURN
      END


      SUBROUTINE EUTAPE
C                                                                       C
C***********************************************************************C
C                                                                       C
```

```
C  SUBROUTINE EUTAPE-READS A.F.W.L. FORMATTED TAPES TO OBTAIN THE        C
C                    EXPERIMENTAL DATA                                    C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  SUBROUTINE CALLED BY EXPERT                                            C
C                                                                         C
C                                                                         C
C*************************************************************************C
C                                                                         C
      WRITE(5,10)
 10   FORMAT(' ',/,T15,' ***E.U. TAPE READ IS NOT PRESENTLY
     * OPERABLE***'/)
      RETURN
      END


      SUBROUTINE EXTRM(FUNCT,NPT,FMAX,FMIN)
C                                                                         C
C*************************************************************************C
C                                                                         C
C  SUBROUTINE EXTRM-FINDS THE EXTREME (ABSOLUTE MAXIMUM POSITIVE AND      C
C                 AND NEGATIVE) VALUES OF AN ARRAY                        C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  SUBROUTINE CALLED BY THIST                                             C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  INPUT ARGUMENTS: FUNCT, NPT                                            C
C                                                                         C
C  OUTPUT ARGUMENTS: FMAX, FMIN                                           C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  ARRAY DIMENSIONS                                                       C
C     FUNCT(NPT) WHERE NPT EQUALS THE NUMBER OF POINTS IN THE ARRAY       C
C                                                                         C
C*************************************************************************C
C                                                                         C
      DIMENSION FUNCT(800)
      FMAX=FUNCT(1)
      FMIN=FUNCT(1)
      DO 10 I=2,NPT
         IF(FUNCT(I).GT.FMAX) FMAX=FUNCT(I)
         IF(FUNCT(I).LT.FMIN) FMIN=FUNCT(I)
 10   CONTINUE
      RETURN
      END


      SUBROUTINE GRID(NXI,NYI,XI,YI)
C
C*************************************************************************C
C                                                                         C
C  SUBROUTINE GRID-CREATES A GRID OF POINTS AT WHICH INTERPOLATION IS    C
C                 PERFORMED                                               C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  SUBROUTINE CALLED BY EXPERT                                            C
C                                                                         C
C                                                                         C
C*************************************************************************C
C                                                                         C
```

```
C  INPUT ARGUMENTS: NONE                                                  C
C                                                                        C
C  OUTPUT ARGUMENTS: NXI,NYI,XI,YI                                       C
C                                                                        C
C************************************************************************C
C                                                                        C
C  ARRAY DIMENSIONS                                                      C
C     XI(NXI) WHERE NXI IS THE NUMBER OF GRID VALUES IN THE X COORD      C
C     YI(NYI) WHERE NYI IS THE NUMBER OF GRID VALUES IN THE Y COORD      C
C                                                                        C
C************************************************************************C
C                                                                        C
      DIMENSION XI(31),YI(31)
      COMMON/UNIT/ILUN,LUN,IDSK
      WRITE(LUN,5)
   5  FORMAT(' ','FORMATION OF GRID SYSTEM:',/)
      WRITE(LUN,10)
  10  FORMAT('$','X-COORDINATE OF THE FIRST VERTICAL GRID ',
     * 'LINE: ')
      READ(ILUN,*) XFIRST
      WRITE(LUN,20)
  20  FORMAT('$','X-COORDINATE OF THE LAST VERTICAL GRID ',
     * 'LINE: ')
      READ(ILUN,*) XLAST
      WRITE(LUN,30)
  30  FORMAT('$','TOTAL NUMBER OF VERTICAL GRID LINES: ')
      READ(ILUN,*) NXI
      WRITE(LUN,40)
  40  FORMAT('$','Y-COORDINATE OF THE FIRST HORIZONTAL ',
     * 'GRID LINE: ')
      READ(ILUN,*) YFIRST
      WRITE(LUN,50)
  50  FORMAT('$','Y-COORDINATE OF THE LAST HORIZONTAL ',
     * 'GRID LINE: ')
      READ(ILUN,*) YLAST
      WRITE(LUN,60)
  60  FORMAT('$','TOTAL NUMBER OF HORIZONTAL GRID LINES: ')
      READ(ILUN,*) NYI
      NXX=NXI
      NYY=NYI
C
C  CALCULATE X INCREMENT AND X COORDINATES OF GRID
C
      DELTAX=(XLAST-XFIRST)/(FLOAT(NXX)-1.)
      XI(1)=XFIRST
      DO 70 I=2,NXI
         J=I-1
         XI(I)=XI(1)+FLOAT(J)*DELTAX
  70  CONTINUE
C
C  CALCULATE Y INCREMENT AND Y COORDINATE OF GRID
C
      DELTAY=(YLAST-YFIRST)/(FLOAT(NYY)-1.)
      YI(1)=YFIRST
      DO 80 I=2,NYI
         J=I-1
         YI(I)=YI(1)+FLOAT(J)*DELTAY
  80  CONTINUE
      RETURN
      END

      SUBROUTINE HELP(MENU,NUM)
C                                                                        C
C************************************************************************C
C                                                                        C
```

```
C  SUBROUTINE HELP-CONTAINS THE TEXT OF THE HELP MESSAGES                 C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  SUBROUTINE CALLED BY SELECT,SELNCP                                      C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  INPUT ARGUMENTS: MENU,LUN                                               C
C                                                                          C
C  OUTPUT ARGUMENTS: NONE                                                  C
C                                                                          C
C**************************************************************************C
C                                                                          C
      COMMON/UNIT/ILUN,LUN,IDSK
      INTEGER PAUSE
      GO TO (10,20,30,40,50,60) MENU
      GO TO 70
 10   WRITE(LUN,15)
 15   FORMAT(' ',/,' HELP MESSAGES FOR OPTION LIST #1:'//
     * ' OPTION 1 ALLOWS A NEW SET OF DATA TO BE INPUT.  IT MAY BE'/
     * '     STORED IF DESIRED.'//
     * ' OPTION 2 READS A DISK DATA FILE AND TYPES IT ON THE TTY,'/
     * '     ALLOWING CORRECTIONS TO BE MADE IF NECESSARY.  UPON'/
     * '     COMPLETION OF THIS COMMAND A NEW SET OF OPTIONS APPEARS.'//
     * ' OPTION 3 ALLOWS A SERIES OF CONSECUTIVE DISK FILES, SUCH AS'/
     * '     THOSE CREATED BY OPTION 4, TO BE READ.  THE FILE NAMES'/
     * '     MUST BEGIN WITH THE SAME LETTER AND BE NUMBERED CONSECU',
     *      'TIVELY.'//
     * ' OPTION 4 READS AN AFWL ENGINEERING UNITS TAPE AND CREATES'/
     * '     A DISK DATA FILE.  UPON COMPLETION OF THE COMMAND'/
     * '     OPTION LIST #1 REAPPEARS.  ADDITIONAL TAPES CAN BE READ,'/
     * '     OR ANY OF THE OTHER OPTIONS MAY BE SELECTED.'/
     * '     ***THE E.U. READ SUBROUTINE IS PRESENTLY INOPERABLE.***'//
     * ' OPTION 5 READS A HONDO.MOV FILE AND CREATES TIME HISTORY',
     *      'FILES.'//
     * ' OPTION 6 PLOTS ONE OR TWO TIME HISTORY FILES.'//
     * '$                  (CONTINUED.  PRESS <RETURN> TO CONTINUE)')
      READ(ILUN,90) PAUSE
      WRITE(LUN,95) NUM
      GO TO 100
 20   WRITE(LUN,25)
 25   FORMAT(' ',///,' HELP MESSAGES FOR OPTION LIST #2:'//
     * ' OPTION 1 READS A DISK DATA FILE CREATED BY THE QMESH'/
     * '     PROGRAM PACKAGE.  THIS FILE MUST HAVE BEEN PREVIOUSLY'/
     * '     CREATED.  THE FILE MUST NOT EXCEED 1000 NODES.  THE'/
     * '     FILE NAME IS REQUIRED AS INPUT.'//
     * ' OPTION 2 ALLOWS THE USER TO CREATE A REGULAR RECTANGULAR'/
     * '     GRID BY SPECIFYING THE COORDINATES OF THE FIRST AND'/
     * '     LAST GRID LINE IN BOTH AXES, AND THE TOTAL NUMBER OF'/
     * '     GRID LINES IN EACH DIRECTION.  THE PRODUCT OF THE NUM-'/
     * '     BER OF GRID LINES IN EACH DIRECTION MUST NOT EXCEED'/
     * '     1000 (NXI*NYI.LE.1000).'//
     * ' OPTION 3 ALLOWS THE USER TO SPECIFY POINTS OF SPECIAL'/
     * '     INTEREST TO BE INTERPOLATED.  THE PROGRAM IS NOT CAP-'/
     * '     ABLE, HOWEVER, OF USING THIS DATA TO CREATE FILES FOR'/
     * '     MOVIE.BYU.'//
     * '$                  (CONTINUED.  PRESS <RETURN> TO CONTINUE)')
      READ(ILUN,90) PAUSE
      WRITE(LUN,95) NUM
      GO TO 100
 30   WRITE(LUN,35)
 35   FORMAT(' ',///,' HELP MESSAGES FOR OPTION LIST #3:'//
     * ' FOR MULTIPLE FILES THIS OPTION LIST ONLY APPEARS ONCE.'/
     * ' FOR SINGLE FILES THIS OPTION REAPPEARS AFTER THE PREVIOUS'/
     * ' COMMAND IS COMPLETED.  (SEE OPTION 8)'//
```

```
     * ' OPTION 1 DISPLAYS THE RESULTS ON THE SCREEN.'//
     * ' OPTION 2 CREATES A DISK FILE OF THE RESULTS.'//
     * ' OPTION 3 CREATES FILES COMPATIBLE WITH MOVIE.BYU.  MOVIE'/
     * '    FILES CAN NOT BE CREATED IF THE INTERPOLATION POINTS WERE'/
     * '    RANDOMLY SPECIFIED (OPTION 3 OF OPTION LIST #2).'/
     * '    <RETURN> IN RESPONSE TO THE GEOMETRY FILENAME.EXT'/
     * '    SUPRESSES CREATION OF THE GEOMETRY FILE.'//
     * ' OPTION 4 THROUGH 7 PERFORMS COMBINATION OF ABOVE OPTIONS'/
     * '    AS INDICATED.'//
     * ' OPTION 8 TRANSFERS CONTROL TO THE FINAL SET OF OPTIONS'//
     * '$                     (CONTINUED.  PRESS <RETURN> TO CONTINUE)')
      READ(ILUN,90) PAUSE
      WRITE(LUN,95) NUM
      GO TO 100
40    WRITE(LUN,45) NUM,NUM
45    FORMAT(' ',/' HELP MESSAGES FOR NCP:'//
     * ' NCP IS THE NUMBER OF CLOSEST DATA POINTS USED TO INTERPOLATE'/
     * '   AT A GIVEN POINT.  THE OPTIMUM VALUE TO USE IS SOMEWHAT'/
     * '   DEPENDENT UPON THE DISTRIBUTION OF THE EXPERIMENTAL DATA'/
     * '   POINTS.'/
     * ' NCP MUST BE GREATER THAN OR EQUAL TO 2, AND LESS THAN THE'/
     * '   NUMBER OF DATA POINTS.  IT HAS BEEN ARBITRARILY LIMITED TO'/
     * '   A MAXIMUM OF 25.'//
     * ' SPECIAL OPTIONS:'//
     * ' <RETURN> CAUSES THE DEFAULT VALUE OF ',I2,' TO BE USED (SAME'/
     * '              AS D).'/
     * '    D     DEFAULT.  THE VALUE OF ',I2,' IS USED.'/
     * '    M     MAXIMUM.  THE MAXIMUM ALLOWABLE VALUE IS USED.  THIS'/
     * '              IS THE SMALLER OF 1 LESS THAN THE NUMBER OF DATA'/
     * '              POINTS AND 25.'/
     * '    H     HELP.  PRINTS THE HELP MESSAGE.'/
     * '    T     TERMINATE.  TERMINATES THE PROGRAM.'//
     * ' ENTER THE DESIRED VALUE OF NCP OR A SPECIAL OPTION.'/)
      GO TO 100
50    WRITE(LUN,55)
55    FORMAT(' ',///,' HELP MESSAGES FOR OPTION LIST #5'//
     * ' OPTION 1 READS A HONDO.MOV FILE AND CREATES TIME HISTORY',
     *   ' FILES.'//
     * ' OPTION 2 PLOTS TIME HISTORIES.'//
     * ' OPTION 3 TRANSFERS TO OPTION LIST #2, "DEVELOPMENT OF',/
     * '    INTERPOLATION POINTS".'//
     * ' OPTION 4 TRANSFERS TO THE START OF THE PROGRAM.'//
     * '$                (CONTINUED.  PRESS <RETURN> TO CONTINUE)')
      READ(ILUN,90) PAUSE
      WRITE(LUN,95) NUM
      GO TO 100
60    WRITE(LUN,65)
65    FORMAT(' ',///,' HELP MESSAGES FOR OPTION LIST #6'//
     * ' OPTION 1 PLOTS ONE CURVE PER GRAPH.'//
     * ' OPTION 2 PLOTS TWO CURVES OF THE SAME GENERAL TYPE.  ONLY'/
     * '    ONE Y-COORDINATE AXIS IS DRAWN.'//
     * ' OPTION 3 PLOTS TWO CURVES OF DIFFERENT TYPES OR OF WIDELY'/
     * '    DIFFERING RANGES.  TWO Y-COORDINATE AXES ARE DRAWN,'/
     * '    SCALED TO THE SEPARATE CURVES.'//
     * '$                (CONTINUED.  PRESS <RETURN> TO CONTINUE)')
      READ(ILUN,90) PAUSE
      WRITE(LUN,95) NUM
      GO TO 100
70    WRITE(LUN,75) MENU
75    FORMAT(' ','THERE ARE NO ''HELP'' MESSAGES FOR OPTION LIST
     * # ',I1/)
90    FORMAT(A1)
95    FORMAT(' ',/,'  SPECIAL OPTIONS:'//
     * '    H  PRINTS THE HELP MESSAGE (SAME AS OPTION #',I2,').'/
     * '    R  TRANSFERS CONTROL TO THE PREVIOUS SET OF OPTIONS.'/
     * '    S  TRANSFERS CONTROL TO THE BEGINNING OF THE PROGRAM.'/
```

```
     * '     T  TERMINATES THE PROGRAM.'/
     * '     <RETURN>  CAUSES THE OPTIONS TO BE RELISTED.'//
     * ' TYPE THE DESIRED OPTION NUMBER OR SPECIAL OPTION:')
 100  RETURN
      END


      SUBROUTINE HONTH
C                                                                          C
C**************************************************************************C
C                                                                          C
C  SUBROUTINE HONTH-CREATES TIME HISTORY FILES FROM HONDO.BYU FILES       C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  SUBROUTINE CALLED BY EXPERT                                             C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  SUBROUTINE CALLED: DISKN (CRAY)                                         C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  INPUT ARGUMENTS: NONE                                                   C
C                                                                          C
C  OUTPUT ARGUMENTS: NONE                                                  C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  IMPORTANT INTERNAL VARIABLES:                                           C
C     R       =ARRAY OF RADIAL COORDINATES                                 C
C     Z       =ARRAY OF VERTICAL COORDINATES                               C
C     U       =ARRAY OF DISPLACEMENTS                                      C
C     V       =ARRAY OF VELOCITIES                                         C
C     A       =ARRAY OF ACCELERATIONS                                      C
C     SIG     =ARRAY OF STRESSES                                           C
C     HISTORY=TEMPORARY ARRAY CONTAINING HISTORIES                         C
C     T       =ARRAY CONTAINING TIMES                                      C
C     IPLOP   =ARRAY CONTAINING PLOT OPTION COINCIDING WITH NONUMP         C
C     NONUMP  =ARRAY OF NODES FOR WHICH HISTORIES ARE TO BE CREATED        C
C     DESIG,DRCTN,TYPE=ALPHA ARRAYS DESCRIBING THE FILE                    C
C     ICOUNT  =NUMBER OF HISTORY TIMESTEPS                                 C
C     NUMNP   =NUMBER OF NODAL POINTS                                      C
C     NHIST   =NUMBER OF HISTORY FILES CREATED                             C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  ARRAY DIMENSIONS                                                        C
C     R(NUMNP),Z(NUMNP),U(NUMNP),V(NUMNP),A(NUMNP)                         C
C     SIG(4,NUMNP),HISTORY(ICOUNT,NHIST),T(ICOUNT)                         C
C     IPLOP(NHIST),NONUMP(NHIST),DESIG(NHIST),DRCTN(NHIST),TYPE(NHIST)     C
C     HED(20)                                                              C
C                                                                          C
C**************************************************************************C
C                                                                          C
      DIMENSION R(2500),Z(2500),U(2500),V(2500),A(2500),SIG(4,2500),
     *          NONUMP(30),HISTORY(800,30),T(800),IPLOP(30),HED(20),
     *          DRCTN(30),TYPE(30),DESIG(30)
CVAX  CHARACTER*12 THNAME,TYPE                                             VAX
CVAX  CHARACTER*1 DRCTN                                                    VAX
CVAX  CHARACTER*2 DESIG                                                    VAX
CVAX  CHARACTER*40 HNAME                                                   VAX
      INTEGER HNAME,THNAME,DESIG,DRCTN,TYPE                                CRAY
      COMMON/UNIT/ILUN,LUN,IDSK
      COMMON NAMOLD                                                        CRAY
      WRITE(LUN,10)
```

```
 10    FORMAT('$','HONDO FILENAME: ')
       READ(ILUN,20) HNAME
C20    FORMAT(A40)                                                        VAX
 20    FORMAT(A8)                                                         CRAY
       WRITE(LUN,30)
 30    FORMAT('$','MAXIMUM TIME: ')
       READ(ILUN,*) TMAX
C
C  READ HONDO FILE
C
       CALL DISKN(HNAME)                                                  CRAY
CVAX   OPEN(UNIT=IDSK,FILE=HNAME,STATUS='OLD',FORM='UNFORMATTED',ACCESS= VAX
CVAX  *'SEQUENTIAL')                                                      VAX
       READ(IDSK) HED,NUMEL,NUMNP
       READ(IDSK) (R(I),I=1,NUMNP),(Z(I),I=1,NUMNP),((JUNK,J=1,NUMEL),
      *  I=1,5)
       NEQ=2*NUMNP
C
C  LIST NODES AND COORDINATES ON TTY
C
       WRITE(LUN,40)
 40    FORMAT('$','DO YOU WANT THE NODE NUMBERS AND COORDINATES LISTED'
      *          ,' ON TTY?')
       READ(ILUN,50) ANS
 50    FORMAT(A1)
       IF(ANS.NE.1HY) GO TO 70
       WRITE(LUN,60) (I,R(I),Z(I),I=1,NUMNP)
 60    FORMAT(' ',' NODE',7X,'R',5X,'Z',/'NUMBER',4X,'COORD',5X,'COORD'/
      *  (I5,2F10.3,/))
C
C  SPECIFY NODES AND TYPE
C
 70    WRITE(LUN,80)
 80    FORMAT('$','NUMBER OF TIME HISTORIES TO BE CREATED: ')
       READ(ILUN,*) NHIST
       WRITE(LUN,90)
 90    FORMAT(' ',/' SPECIFY NODE NUMBER AND PLOT OPTION:')
       WRITE(LUN,100)
100    FORMAT(' ','   PLOT OPTIONS:'/
      *           '          1.  R DISPLACEMENT'/
      *           '          2.  Z DISPLACEMENT'/
      *           '          3.  R VELOCITY'/
      *           '          4.  Z VELOCITY'/
      *           '          5.  R ACCELERATION'/
      *           '          6.  Z ACCELERATION'/
      *           '          7.  PRESSURE'/
      *           '          8.  *RELIST PLOT OPTIONS*')
       DO 140 I=1,NHIST
105       WRITE(LUN,110) I
110       FORMAT(/,'$',I4,2X,'NODE NUMBER: ')
          READ(ILUN,*) NONUMP(I)
          WRITE(LUN,120)
120       FORMAT('$',6X,'PLOT OPTION: ')
          READ(ILUN,*,ERR=130) IPLOP(I)
          IF(IPLOP(I).GE.8) THEN
130          WRITE(LUN,100)
             GO TO 105
          ENDIF
C
C  THE TYPE(I) ALPHA STRINGS MUST BE REDUCED TO 8 CHARACTERS IN THE
C  CRAY VERSION.  THE VAX VERSION USES 12 CHARACTERS.
C
          IF(IPLOP(I).EQ.1) THEN
             DRCTN(I)='R'
             TYPE(I)='DISPLACEMENT'
             DESIG(I)='RD'
```

```
          ENDIF
          IF(IPLOP(I).EQ.2) THEN
             DRCTN(I)='Z'
             TYPE(I)='DISPLACEMENT'
             DESIG(I)='ZD'
          ENDIF
          IF(IPLOP(I).EQ.3) THEN
             DRCTN(I)='R'
             TYPE(I)='VELOCITY    '
             DESIG(I)='RV'
          ENDIF
          IF(IPLOP(I).EQ.4) THEN
             DRCTN(I)='Z'
             TYPE(I)='VELOCITY    '
             DESIG(I)='ZV'
          ENDIF
          IF(IPLOP(I).EQ.5) THEN
             DRCTN(I)='R'
             TYPE(I)='ACCELERATION'
             DESIG(I)='RA'
          ENDIF
          IF(IPLOP(I).EQ.6) THEN
             DRCTN(I)='Z'
             TYPE(I)='ACCELERATION'
             DESIG(I)='ZA'
          ENDIF
          IF(IPLOP(I).EQ.7) THEN
             DRCTN(I)=' '
             TYPE(I)='PRESSURE    '
             DESIG(I)='PP'
          ENDIF
140    CONTINUE
       ICOUNT=0
       WRITE(LUN,145)
145    FORMAT(' '/,' CREATING TIME HISTORIES')
C
C  CONTINUE READING HONDO FILE
C
150    ICOUNT=ICOUNT+1
          READ(IDSK,END=170) TT
          READ(IDSK) (U(I),I=1,NEQ),(V(I),I=1,NEQ),(A(I),I=1,NEQ)
          READ(IDSK) ((SIG(I,J),J=1,NUMEL),I=1,4)
          READ(IDSK) ((JUNK,J=1,NUMEL),I=1,4)
          DO 160 I=1,NHIST
              J=NONUMP(I)
              II=2*J
              IM=II-1
              IF(IPLOP(I).EQ.1) HISTORY(ICOUNT,I)=U(IM)
              IF(IPLOP(I).EQ.2) HISTORY(ICOUNT,I)=U(II)
              IF(IPLOP(I).EQ.3) HISTORY(ICOUNT,I)=V(IM)
              IF(IPLOP(I).EQ.4) HISTORY(ICOUNT,I)=V(II)
              IF(IPLOP(I).EQ.5) HISTORY(ICOUNT,I)=A(IM)
              IF(IPLOP(I).EQ.6) HISTORY(ICOUNT,I)=A(II)
              IF(IPLOP(I).EQ.7) HISTORY(ICOUNT,I)=-(SIG(1,J)+
     *        SIG(2,J)+SIG(3,J))/3.0
160       CONTINUE
          T(ICOUNT)=TT
       IF(TT.LE.TMAX) GO TO 150
       GO TO 180
170    ICOUNT=ICOUNT-1
180    CONTINUE
CVAX   CLOSE(UNIT=IDSK)                                              VAX
C
C  WRITE HISTORY FILES
C
       DO 240 J=1,NHIST
```

```
CVAX      NCODE=12                                                 VAX
          NCODE=8                                                  CRAY
          IF(NONUMP(J).GE.10000) THEN
             ENCODE(NCODE,190,THNAME) DESIG(J),NONUMP(J)
190          FORMAT('H',A2,I5)
             GO TO 199
          ENDIF
          IF(NONUMP(J).GE.1000) THEN
             ENCODE(NCODE,192,THNAME) DESIG(J),NONUMP(J)
192          FORMAT('H',A2,I4)
             GO TO 199
          ENDIF
          IF(NONUMP(J).GE.100) THEN
             ENCODE(NCODE,194,THNAME) DESIG(J),NONUMP(J)
194          FORMAT('H',A2,I3)
             GO TO 199
          ENDIF
          IF(NONUMP(J).GE.10) THEN
             ENCODE(NCODE,196,THNAME) DESIG(J),NONUMP(J)
196          FORMAT('H',A2,I2)
             GO TO 199
          ENDIF
          ENCODE(NCODE,198,THNAME) DESIG(J),NONUMP(J)
198       FORMAT('H',A2,I1)
199       CONTINUE
          CALL DISKN(THNAME)                                       CRAY
CVAX      OPEN(UNIT=IDSK,FILE=THNAME,STATUS='NEW')                 VAX
          WRITE(IDSK,200) THNAME
          WRITE(IDSK,200) TYPE(J)
          WRITE(IDSK,210) DRCTN(J)
          WRITE(IDSK,220) ICOUNT
          WRITE(IDSK,*) (HISTORY(I,J),I=1,ICOUNT)
          WRITE(IDSK,*) (T(I),I=1,ICOUNT)
CVAX      CLOSE(UNIT=IDSK)                                         VAX
200       FORMAT(1X,A12)
210       FORMAT(1X,A1)
220       FORMAT(I5)
          WRITE(LUN,230) THNAME
230       FORMAT(' ',/' HONDO TIME HISTORY FILE ',A12,' CREATED')
240     CONTINUE
        RETURN
        END

        SUBROUTINE MESH(KOMT,XI,YI,N1,N2,N3,N4,NPT,NJ,QNAME)
C                                                                      C
C**********************************************************************C
C                                                                      C
C   SUBROUTINE MESH-READS A QMESH.BYU FILE AND ASSIGNS THE MESH POINTS C
C                   AS POINTS OF INTERPOLATION                         C
C                                                                      C
C**********************************************************************C
C                                                                      C
C   SUBROUTINE CALLED BY EXPERT                                        C
C                                                                      C
C**********************************************************************C
C                                                                      C
C   SUBROUTINE CALLED: DISKN (CRAY)                                    C
C                                                                      C
C**********************************************************************C
C                                                                      C
C   INPUT ARGUMENTS: NONE                                              C
C                                                                      C
C   OUTPUT ARGUMENTS: ALL                                              C
C                                                                      C
C**********************************************************************C
```

```
C                                                                        C
C  IMPORTANT INTERNAL VARIABLES:                                         C
C      N1,N2,N3,N4=CONNECTIVITY ARRAY                                    C
C      NJ=NUMBER OF JOINTS IN THE QMESH FILE                             C
C      NPT=NUMBER OF POINTS IN THE CONNECTIVITY ARRAY                    C
C                                                                        C
C************************************************************************C
C                                                                        C
C  ARRAY DIMENSIONS                                                      C
C      KOMT(8)                                                           C
C      XI(NJ),YI(NJ)                                                     C
C      N1(NPT),N2(NPT),N3(NPT),N4(NPT),MAT(NPT)                          C
C                                                                        C
C************************************************************************C
C                                                                        C
      DIMENSION KOMT(8),XI(1000),YI(1000),N1(1000),N2(1000),
     *          N3(1000),N4(1000),MAT(1000)
CVAX  CHARACTER*12 QNAME                                                 VAX
      INTEGER QNAME                                                      CRAY
      COMMON/UNIT/ILUN,LUN,IDSK
      COMMON NAMOLD                                                      CRAY
      WRITE(LUN,10)
 10   FORMAT('$','ENTER THE FILENAME.EXT: ')
      READ(ILUN,20) QNAME
 20   FORMAT(A12)
      CALL DISKN(QNAME)                                                  CRAY
CVAX  OPEN(UNIT=IDSK,FILE=QNAME,STATUS='OLD',ACCESS='SEQUENTIAL',FORM=   VAX
CVAX * 'UNFORMATTED')                                                    VAX
      READ(IDSK) KOMT
      READ(IDSK) NPT,NJ,JUNK1,JUNK2
      READ(IDSK) (XI(N),YI(N),N=1,NJ)
      READ(IDSK) (N1(N),N2(N),N3(N),N4(N),MAT(N),N=1,NPT)
CVAX  CLOSE(UNIT=IDSK)                                                   VAX
      RETURN
      END


      SUBROUTINE MOFILE(IFLAG1,NXI,NYI,XI,YI,ZI,NPT,NIP,N1,N2,N3,N4,
     *   MULTPL,NCOUNT,NUMBER)
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE MOFILE-CREATES MOVIE.BYU FORMATTED FILES OF INTERPOLATED C
C                     RESULTS                                            C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE CALLED BY EXPERT                                           C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE CALLED: DISKN (CRAY)                                       C
C                                                                        C
C************************************************************************C
C                                                                        C
C  INPUT ARGUMENTS: ALL ARGUMENTS ARE INPUT                              C
C                   (NOTE: NPT IS ONLY DEFINED WHEN THE INTERPOLATION  C
C                    POINTS ARE DEFINED BY QMESH FILE)                   C
C                                                                        C
C  OUTPUT ARGUMENTS: NONE                                                C
C                                                                        C
C************************************************************************C
C                                                                        C
C  IMPORTANT INTERNAL VARIABLES:                                         C
C      IP=CONNECTIVITY ARRAY                                             C
C      NCON=NUMBER OF VALUES IN THE CONNECTIVITY ARRAY                   C
```

```
C                                                                        C
C************************************************************************C
C                                                                        C
C  ARRAY DIMENSIONS                                                      C
C      XI(MAX0(NXI,NIP))                                                 C
C      YI(MAX0(NYI,NIP))                                                 C
C      ZI(MAX0(NXI*NYI,NIP)                                              C
C      IP(MAX0(4*(NXI-1)*(NYI-1),4*NPT))                                 C
C      N1,N2,N3,N4(MAX0(NPT,(NXI-1)*(NYI-1))                             C
C                                                                        C
C************************************************************************C
C                                                                        C
      DIMENSION XI(1000),YI(1000),ZI(1000),IP(3600),
     *          N1(1000),N2(1000),N3(1000),N4(1000)
CVAX  CHARACTER*12 GNAME,MNAME                                          VAX
      INTEGER GNAME,MNAME                                               CRAY
      LOGICAL MULTPL
      COMMON/UNIT/ILUN,LUN,IDSK
      COMMON NAMOLD                                                     CRAY
      NP=1
      NTEST=0
      ZERO=0.0
      GO TO (80,10,130) IFLAG1
C
C  CREATE FILES FROM GRID INTERPOLATION POINTS
C
 10   NIP=NXI*NYI
      NPT=(NXI-1)*(NYI-1)
      NCON=4*NPT
      N=0
      NN=0
      NM=NXI-1
C
C  CREATE CONNECTIVITY ARRAY
C
      DO 20 I=1,NCON,4
         J=I+1
         K=I+2
         L=I+3
         N=N+1
         NN=NN+1
         IP(I)=N
         IP(J)=NXI+N
         IP(K)=NXI+N+1
         IP(L)=-(N+1)
         IF(NN.NE.NM) GO TO 20
         N=N+1
         NN=0
 20   CONTINUE
      IF(MULTPL.AND.NUMBER.NE.1) GO TO 50
C
C  WRITE GEOMETRY FILE
C
      WRITE(LUN,30)
 30   FORMAT('$','ASSIGN THE GEOMETRY FILENAME.EXT ')
      READ(ILUN,40) GNAME
 40   FORMAT(A12)
      IF(GNAME.EQ.'            ') GO TO 50
      CALL DISKN(GNAME)                                                 CRAY
CVAX  OPEN(UNIT=IDSK,FILE=GNAME,STATUS='UNKNOWN')                       VAX
      WRITE(IDSK,150) NP,NIP,NPT,NCON,NTEST
      WRITE(IDSK,150) NP,NPT
      WRITE(IDSK,160) ((XI(I),YI(J),ZERO,I=1,NXI),J=1,NYI)
      WRITE(IDSK,150) (IP(I),I=1,NCON)
CVAX  CLOSE(UNIT=IDSK)                                                  VAX
C
```

```
C  WRITE FUNCTION FILE
C
 50    IF(MULTPL) THEN
          CALL ENC('M',NCOUNT,MNAME)
       ELSE
          WRITE(LUN,70)
 70       FORMAT('$','ASSIGN THE FUNCTION FILENAME.EXT ')
          READ(ILUN,40) MNAME
       ENDIF
       CALL DISKN(MNAME)                                                  CRAY
CVAX   OPEN(UNIT=IDSK,FILE=MNAME,STATUS='UNKNOWN')                        VAX
       WRITE(IDSK,160) (ZI(I),I=1,NIP)
CVAX   CLOSE(UNIT=IDSK)                                                   VAX
       GO TO 170
C
C  CREATE FILES FROM QMESH INTERPOLATION POINTS
C
 80    NCON=4*NPT
       IF(MULTPL.AND.NUMBER.NE.1) GO TO 110
C
C  WRITE GEOMETRY FILE
C
       WRITE(LUN,90)
 90    FORMAT('$','ASSIGN THE GEOMETRY FILENAME.EXT ')
       READ(ILUN,100) GNAME
100    FORMAT(A12)
       IF(GNAME.EQ.'            ') GO TO 110
       CALL DISKN(GNAME)                                                  CRAY
CVAX   OPEN(UNIT=IDSK,FILE=GNAME,STATUS='UNKNOWN')                        VAX
       WRITE(IDSK,150) NP,NIP,NPT,NCON,NTEST
       WRITE(IDSK,150) NP,NPT
       WRITE(IDSK,160) (XI(I),YI(I),ZERO,I=1,NIP)
       WRITE(IDSK,150) (N1(I),N2(I),N3(I),-N4(I),I=1,NPT)
CVAX   CLOSE(UNIT=IDSK)                                                   VAX
C
C  WRITE FUNCTION FILE
C
110    IF(MULTPL) THEN
          CALL ENC('M',NCOUNT,MNAME)
       ELSE
          WRITE(LUN,120)
120       FORMAT('$','ASSIGN THE FUNCTION FILENAME.EXT ')
          READ(ILUN,100) MNAME
       ENDIF
       CALL DISKN(MNAME)                                                  CRAY
CVAX   OPEN(UNIT=IDSK,FILE=MNAME,STATUS='UNKNOWN')                        VAX
       WRITE(IDSK,160) (ZI(I),I=1,NIP)
CVAX   CLOSE(UNIT=IDSK)                                                   VAX
       GO TO 170
130    WRITE(LUN,140)
140    FORMAT(' ',/,T15,' MOVIE.BYU FILES CAN NOT BE FORMED.
      *  TYPE HELP.')
150    FORMAT(16I5)
160    FORMAT(6E12.5)
170    RETURN
       END


       SUBROUTINE MULTI(NUMFIL,LETTER,NCOUNT,INCR,NHIST,COORDX,
      *  COORDY,THNAME,IFLAG3,IFLAG4,TYPE,DRCTN)
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE MULTI-SETS PARAMETERS TO READ AND CREATE MULTIPLE FILES    C
C                                                                        C
C************************************************************************C
```

```
C                                                                         C
C  SUBROUTINE CALLED BY EXPERT                                            C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  INPUT ARGUMENTS: NONE                                                  C
C                                                                         C
C  OUTPUT ARGUMENTS: ALL                                                  C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  IMPORTANT INTERNAL VARIABLES:                                          C
C     COORDX=ARRAY OF X-COORDINATES OF POINTS FOR WHICH TIME HISTORIES C
C            ARE CREATED                                                  C
C     COORDY=ARRAY OF Y-COORDINATES OF POINTS FOR WHICH TIME HISTORIES C
C            ARE CREATED                                                  C
C     THNAME=CHARACTER ARRAY CONTAINING THE NAMES TO BE GIVEN TO THE      C
C            TIME HISTORY FILES                                           C
C     LETTER=FIRST LETTER IN DATA FILENAME                                C
C     NCOUNT=ONE TO FIVE DIGIT NUMBER IN DATA FILENAME                    C
C     INCR  =INCREMENT BETWEEN THE DATA FILE NAMES                        C
C     TYPE  =TYPE OF FILE (DISP,VEL,ACCEL,ETC.)                           C
C     DRCTN =DIRECTION (Z,R)                                              C
C     IFLAG3=INDICATES IF TIME HISTORIES ARE TO BE CREATED                C
C            0=NO TIME HISTORIES                                          C
C            1=TIME HISTORIES                                             C
C     IFLAG4=INDICATES IF OTHER INTERPOLATIONS ARE TO BE PERFORMED        C
C            0=ONLY ONE TYPE                                              C
C            1=BOTH TIME HISTORIES AND OTHER INTERPOLATIONS               C
C                                                                         C
C*************************************************************************C
C                                                                         C
C  ARRAY DIMENSIONS                                                       C
C     COORDX(NHIST), COORDY(NHIST)                                        C
C     THNAME(NHIST)                                                       C
C                                                                         C
C*************************************************************************C
C                                                                         C
      DIMENSION COORDX(30),COORDY(30),THNAME(30)
CVAX  CHARACTER*12 THNAME,TYPE                                             VAX
CVAX  CHARACTER*1 DRCTN,LETTER                                             VAX
      INTEGER THNAME,TYPE,DRCTN,ANS1,ANS2                                  CRAY
      COMMON/UNIT/ILUN,LUN,IDSK
C
C  MULTIPLE DATA FILE NAMES
C
      WRITE(LUN,10)
 10   FORMAT('$','NUMBER OF FILES: ')
      READ(ILUN,*) NUMFIL
C
C    VAX
C
C20   WRITE(LUN,30)                                                        VAX
C30   FORMAT('$','FILENAME OF FIRST FILE: ')                               VAX
CVAX  READ(ILUN,40,ERR=60) LETTER,NCOUNT                                   VAX
C40   FORMAT(A1,I5)                                                        VAX
C
C    CRAY
C
 20   WRITE(LUN,25)                                                        CRAY
 25   FORMAT('LETTER ASSOCIATED WITH FILENAME:')                           CRAY
      READ(ILUN,30) LETTER                                                 CRAY
 30   FORMAT(A1)                                                           CRAY
      WRITE(LUN,40)                                                        CRAY
 40   FORMAT('ONE- TO FIVE- DIGIT NUMBER ASSOCIATED WITH FIRST ',          CRAY
     *'FILENAME:')                                                         CRAY
```

```
      READ(ILUN,*) NCOUNT                                                CRAY
      WRITE(LUN,50)
 50   FORMAT('$','FILENAME INCREMENT: ')
      READ(ILUN,*) INCR
      GO TO 80
C60   WRITE(LUN,70)                                                      VAX
C70   FORMAT(' ','FILENAME INPUT ERROR.  MUST BE ONE LETTER FOLLOWED'/,  VAX
CVAX *           ' BY A ONE- TO FIVE- DIGIT NUMBER.  DO NOT ATTACH'/,    VAX
CVAX *           ' THE EXTENSION.'/)                                     VAX
CVAX  GO TO 20                                                           VAX
 80   WRITE(LUN,90)
 90   FORMAT('$','DO YOU WANT TIME HISTORIES? ')
      READ(ILUN,100) ANS1
100   FORMAT(A1)
      IFLAG4=1
C
C  DESIGNATION OF TIME HISTORY COORDINATES AND FILENAMES
C
      IF(ANS1.EQ.1HY) THEN
        IFLAG3=1
        WRITE(LUN,110)
110     FORMAT('$','NUMBER OF POINTS: ')
        READ(ILUN,*) NHIST
        DO 150 I=1,NHIST
           WRITE(LUN,120) I
120        FORMAT(/,'$',I4,2X,'INPUT COORDINATES (X,Y): ')
           READ(ILUN,*) COORDX(I),COORDY(I)
           WRITE(LUN,130)
130        FORMAT('$',6X,'ASSIGN TIME HISTORY FILENAME: ')
           READ(ILUN,140) THNAME(I)
140        FORMAT(A12)
150     CONTINUE
C
C  PLOT ENHANCEMENTS FOR TIME HISTORIES
C
160     WRITE(LUN,170)
170     FORMAT(' ',/' TYPE:'/
     *              '   1.  DISPLACEMENT'/
     *              '   2.  VELOCITY'/
     *              '   3.  ACCELERATION'/
     *              '   4.  PRESSURE'/
     *              '   5.  OTHER')
        NUM=6
        MENU=6
        CALL SELECT(NUM,MENU,NSEL)
        IF(NSEL.LT.1) GO TO 160
C
C  THE TYPE ALPHA STRINGS MUST BE REDUCED TO 8 CHARACTERS IN THE
C  CRAY VERSION.  THE VAX VERSION USES 12 CHARACTERS.
C
        IF(NSEL.EQ.1) TYPE='DISPLACEMENT'
        IF(NSEL.EQ.2) TYPE='VELOCITY    '
        IF(NSEL.EQ.3) TYPE='ACCELERATION'
        IF(NSEL.EQ.4) THEN
           TYPE='PRESSURE    '
           DRCTN=' '
           GO TO 230
        ENDIF
        IF(NSEL.EQ.5) THEN
           WRITE(LUN,180)
180        FORMAT('$','INPUT TYPE ')
           READ(ILUN,140) TYPE
        ENDIF
190     WRITE(LUN,200)
200     FORMAT(' ',/' DIRECTION'/
     *              '   1.  VERTICAL'/
```

```
     *                '   2.  RADIAL'/
     *                '   3.  OTHER')
        NUM=4
        MENU=6
        CALL SELECT(NUM,MENU,NSEL)
        IF(NSEL.LT.1) GO TO 190
        IF(NSEL.EQ.1) DRCTN='Z'
        IF(NSEL.EQ.2) DRCTN='R'
        IF(NSEL.EQ.3) THEN
           WRITE(LUN,210)
210        FORMAT('$','INPUT DIRECTION (ONE CHARACTER): ')
           READ(ILUN,220) DRCTN
220        FORMAT(A1)
        ENDIF
230     WRITE(LUN,240)
240     FORMAT('$','DO YOU WANT OTHER INTERPOLATIONS PERFORMED? ')
        READ(ILUN,100) ANS2
        IF(ANS2.NE.1HY) IFLAG4=0
      ELSE
        IFLAG3=0
      ENDIF
      RETURN
      END

CSSET SUBROUTINE OPEN(FNAME,IUNIT)                                     SSET
C                                                                     CSSET
C*********************************************************************CSSET
C                                                                     CSSET
C  SUBROUTINE OPEN-OPENS DISK ON THE SUPERSET                         CSSET
C                                                                     CSSET
C*********************************************************************CSSET
C                                                                     CSSET
C  SUBROUTINE CALLED BY THIST                                         CSSET
C                                                                     CSSET
C*********************************************************************CSSET
C                                                                     CSSET
C  SYSTEM SUBROUTINES CALLED: OPSYSTEM                                CSSET
C                                                                     CSSET
C*********************************************************************CSSET
C                                                                     CSSET
C  INPUT ARGUMENTS: FNAME                                             CSSET
C                                                                     CSSET
C  OUTPUT ARGUMENTS: IUNIT                                            CSSET
C                                                                     CSSET
C*********************************************************************CSSET
C                                                                     CSSET
C  IMPORTANT INTERNAL VARIABLES:                                      CSSET
C      FNAME   = NAME OF FILE                                         CSSET
C      IUNIT   = DISK NUMBER                                          CSSET
C      LSTRING = CHARACTER STRING FOR OPENING DISK                    CSSET
C                                                                     CSSET
C*********************************************************************CSSET
C                                                                     CSSET
C  ARRAY DIMENSIONS                                                   CSSET
C      FNAME(2)                                                       CSSET
C      LSTRING(10)                                                    CSSET
C                                                                     CSSET
C*********************************************************************CSSET
C                                                                     CSSET
CSSET COMMON/UNIT/ILUN,LUN,IDSK                                        SSET
CSSET INTEGER LSTRING(10)                                              SSET
CSSET INTEGER FNAME(2)                                                 SSET
CSSET IERROR=1                                                         SSET
CSSET IUNIT='!BYU'                                                     SSET
CSSET ENCODE(60,10,LSTRING) FNAME                                      SSET
```

```
C10   FORMAT('#ACTIVATE U=!BYU,E=',2A)                                  SSET
CSSET CALL OPSYSTEM(LSTRING,60)                                         SSET
CSSET RETURN                                                            SSET
CSSET END                                                               SSET


      SUBROUTINE PRINT(LETTER,NCOUNT,INCR,IFLAG1,IUNIT,NDP,NCP,
     *    NIP,NXI,NYI,DNAME,QNAME,KOMT,MULTPL)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE PRINT-CALLS APPROPRIATE PRINT SUBROUTINE ACCORDING TO     C
C                 THE TYPE OF INTERPOLATION POINTS                      C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE CALLED BY EXPERT                                          C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINES CALLED: PRINTG,PRINTQ,PRINTS                             C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  INPUT ARGUMENTS: ALL                                                 C
C                                                                       C
C  OUTPUT ARGUMENTS: NONE                                               C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  ARRAY DIMENSIONS                                                     C
C      XD,YD,ZD(NDP)                                                    C
C      XI(MAX0(NXI,NIP)                                                 C
C      YI(MAX0(NYI,NIP)                                                 C
C      ZI(MAX0(NXI*NYI,NIP)                                             C
C      XIS,YIS,ZIS(NIPS) WHERE NIPS = THE NUMBER OF SPECIFIED RANDOM    C
C        POINTS                                                         C
C                                                                       C
C***********************************************************************C
C                                                                       C
      DIMENSION XD(30),YD(30),ZD(30),XI(1000),YI(1000),ZI(1000),
     *          XIS(30),YIS(30),ZIS(30)
      COMMON/DATA/XD,YD,ZD,XI,YI,ZI,XIS,YIS,ZIS
CVAX  CHARACTER*12 DNAME,QNAME                                          VAX
      INTEGER DNAME,QNAME                                               CRAY
      LOGICAL MULTPL
      GO TO (10,20,30) IFLAG1
 10   CALL PRINTQ(IUNIT,NDP,XD,YD,ZD,NIP,XI,YI,ZI,NCP,QNAME,DNAME,
     *   KOMT,MULTPL,NCOUNT)
      GO TO 40
 20   CALL PRINTG(IUNIT,NDP,XD,YD,ZD,NXI,NYI,XI,YI,ZI,NCP,
     *   DNAME,MULTPL,NCOUNT)
      GO TO 40
 30   CALL PRINTS(IUNIT,NDP,XD,YD,ZD,NIP,XIS,YIS,ZIS,NCP,DNAME,
     *   MULTPL,NCOUNT)
 40   CONTINUE
      RETURN
      END


      SUBROUTINE PRINTG(IUNIT,NDP,XD,YD,ZD,NXI,NYI,XI,YI,
     *                  ZI,NCP,DNAME,MULTPL,NCOUNT)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE PRINTG-PRINTS THE RESULTS ON TTY OR DISK FOR INTERPOL-    C
```

```
C                      ATION POINTS SPECIFIED BY GRID                  C
C                                                                      C
C**********************************************************************C
C                                                                      C
C   SUBROUTINE CALLED BY PRINT                                         C
C                                                                      C
C**********************************************************************C
C                                                                      C
C   SUBROUTINE CALLED: DISKN (CRAY)                                    C
C                                                                      C
C**********************************************************************C
C                                                                      C
C   INPUT ARGUMENTS: ALL                                               C
C                                                                      C
C   OUTPUT ARGUMENTS: NONE                                             C
C                                                                      C
C**********************************************************************C
C                                                                      C
      DIMENSION  XD(NDP),YD(NDP),ZD(NDP),XI(NXI),YI(NYI),ZI(NXI*NYI)
      COMMON/UNIT/ILUN,LUN,IDSK
      COMMON NAMOLD                                                     CRAY
CVAX  CHARACTER*12 DNAME,RNAME                                          VAX
      INTEGER DNAME,RNAME                                               CRAY
      LOGICAL MULTPL
      IF(IUNIT.EQ.LUN) GO TO 40
      IF(.NOT.MULTPL) THEN
        WRITE(LUN,10)
 10     FORMAT(' ',/'$ASSIGN THE FILENAME.EXT: ')
        READ(ILUN,20) RNAME
 20     FORMAT(A12)
      ELSE
        CALL ENC('R',NCOUNT,RNAME)
      ENDIF
      CALL DISKN(RNAME)                                                 CRAY
      IUNIT=IDSK                                                        CRAY
CVAX  OPEN(UNIT=IUNIT,FILE=RNAME,STATUS='NEW')                          VAX
 40   IF(DNAME.NE.'            ') WRITE(IUNIT,50) DNAME
 50   FORMAT(' ',/,' INTERPOLATED DATA CORRESPONDING TO DATA FILE ',
     * A12/)
      NZI=NXI*NYI
      WRITE(IUNIT,60) NCP,NDP,NZI
 60   FORMAT(' ','NCP=',I3,/,' NUMBER OF DATA POINTS=',I3,/,
     * ' NUMBER OF INTERPOLATION POINTS=',I4/)
      WRITE(IUNIT,70)
 70   FORMAT(' ','INPUT DATA POINTS:'/,8X,'X',14X,'Y',14X,'Z')
      WRITE(IUNIT,80) (XD(N),YD(N),ZD(N),N=1,NDP)
 80   FORMAT(1P3E15.5)
      WRITE(IUNIT,90)
 90   FORMAT(' ',///,' INTERPOLATED VALUES AT GRID POINTS:'/,9X,'X',
     * 14X,'Y',14X,'Z')
      DO 110 I=1,NXI
         DO 110 J=1,NYI
            K=(J-1)*NXI+I
            WRITE(IUNIT,100) XI(I),YI(J),ZI(K)
100         FORMAT(1P3E15.5)
110    CONTINUE
CVAX  IF(IUNIT.NE.LUN) CLOSE(UNIT=IUNIT)                                VAX
      RETURN
      END


      SUBROUTINE PRINTQ(IUNIT,NDP,XD,YD,ZD,NIP,XI,YI,ZI,NCP,QNAME,DNAME,
     * KOMT,MULTPL,NCOUNT)
C                                                                      C
C**********************************************************************C
C                                                                      C
```

```
C  SUBROUTINE PRINTQ-PRINTS THE RESULTS ON TTY OR DISK FOR INTERPOL-    C
C                    ATION POINTS SPECIFIED BY QMESH FILE               C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE CALLED BY PRINT                                           C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE CALLED: DISKN (CRAY)                                      C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  INPUT ARGUMENTS: ALL                                                 C
C                                                                       C
C  OUTPUT ARGUMENTS: NONE                                               C
C                                                                       C
C***********************************************************************C
C                                                                       C
      DIMENSION XD(NDP),YD(NDP),ZD(NDP),XI(NIP),YI(NIP),ZI(NIP),KOMT(8)
      COMMON/UNIT/ILUN,LUN,IDSK
      COMMON NAMOLD                                                     CRAY
      LOGICAL MULTPL
CVAX  CHARACTER*12 DNAME,QNAME,RNAME                                    VAX
      INTEGER DNAME,QNAME,RNAME                                         CRAY
      IF(IUNIT.EQ.LUN) GO TO 40
      IF(.NOT.MULTPL) THEN
        WRITE(LUN,10)
 10     FORMAT(' ',/'$ASSIGN THE FILENAME.EXT: ')
        READ(ILUN,20) RNAME
 20     FORMAT(A12)
      ELSE
        CALL ENC('R',NCOUNT,RNAME)
      ENDIF
      CALL DISKN(RNAME)                                                 CRAY
      IUNIT=IDSK                                                        CRAY
CVAX  OPEN(UNIT=IUNIT,FILE=RNAME,STATUS='NEW')                          VAX
 40   WRITE(IUNIT,50) QNAME
 50   FORMAT(' ',/,' INTERPOLATED DATA CORRESPONDING TO QMESH FILE ',
     * A12)
      IF(DNAME.NE.'            ') WRITE(IUNIT,60) DNAME
 60   FORMAT(' ',T34,'AND DATA FILE ',A12/)
      WRITE(IUNIT,70) KOMT
 70   FORMAT(' ',T5,'<< ',8A4,' >>'/)
      WRITE(IUNIT,80) NCP,NDP,NIP
 80   FORMAT(' ','NCP=',I3,/,' NUMBER OF DATA POINTS=',I3,/,
     * ' NUMBER OF INTERPOLATION POINTS=',I4/)
      WRITE(IUNIT,90)
 90   FORMAT(' ','INPUT DATA POINTS:'/,8X,'X',14X,'Y',14X,'Z')
      WRITE(IUNIT,100) (XD(N),YD(N),ZD(N),N=1,NDP)
100   FORMAT(1P3E15.5)
      WRITE(IUNIT,110)
110   FORMAT(' ',///,' INTERPOLATED VALUES AT MESH POINTS:'/,8X,'X',
     * 14X,'Y',14X,'Z')
      WRITE(IUNIT,100) (XI(N),YI(N),ZI(N),N=1,NIP)
CVAX  IF(IUNIT.NE.LUN) CLOSE(UNIT=IUNIT)                                VAX
      RETURN
      END


      SUBROUTINE PRINTS(IUNIT,NDP,XD,YD,ZD,NIP,XIS,YIS,ZIS,NCP,
     * DNAME,MULTPL,NCOUNT)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE PRINTS-PRINTS THE RESULTS ON TTY OR DISK FOR INTERPOL-    C
```

```
C                     ATION POINTS SPECIFIED BY RANDOM POINTS            C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE CALLED BY PRINT                                            C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE CALLED: DISKN (CRAY)                                       C
C                                                                        C
C************************************************************************C
C                                                                        C
C  INPUT ARGUMENTS: ALL                                                  C
C                                                                        C
C  OUTPUT ARGUMENTS: NONE                                                C
C                                                                        C
C************************************************************************C
C                                                                        C
      DIMENSION XD(NDP),YD(NDP),ZD(NDP),XIS(NIP),YIS(NIP),ZIS(NIP)
      COMMON/UNIT/ILUN,LUN,IDSK
      COMMON NAMOLD                                                       CRAY
      LOGICAL MULTPL
CVAX  CHARACTER*12 DNAME,RNAME                                            VAX
      INTEGER DNAME,RNAME                                                 CRAY
      IF(IUNIT.EQ.LUN) GO TO 40
      IF(.NOT.MULTPL) THEN
        WRITE(LUN,10)
 10     FORMAT(' ',/'$ASSIGN THE FILENAME.EXT: ')
        READ(ILUN,20) RNAME
 20     FORMAT(A12)
      ELSE
        CALL ENC('R',NCOUNT,RNAME)
      ENDIF
      CALL DISKN(RNAME)                                                   CRAY
      IUNIT=IDSK                                                          CRAY
CVAX  OPEN(UNIT=IUNIT,FILE=RNAME,STATUS='NEW')                            VAX
 40   IF(DNAME.NE.'            ') WRITE(IUNIT,50) DNAME
 50   FORMAT(' ',/,' INTERPOLATED DATA CORRESPONDING TO DATA FILE ',
     * A12/)
      WRITE(IUNIT,60) NCP,NDP,NIP
 60   FORMAT(' ','NCP=',I3,/,' NUMBER OF DATA POINTS=',I3,/,
     * ' NUMBER OF INTERPOLATION POINTS=',I4/)
      WRITE(IUNIT,70)
 70   FORMAT(' ','INPUT DATA POINTS:'/,8X,'X',14X,'Y',14X,'Z')
      WRITE(IUNIT,80) (XD(N),YD(N),ZD(N),N=1,NDP)
 80   FORMAT(1P3E15.5)
      WRITE(IUNIT,90)
 90   FORMAT(' ',///,' INTERPOLATED VALUES AT SPECIFIED POINTS:'/,8X,
     * 'X',14X,'Y',14X,'Z')
      WRITE(IUNIT,100) (XIS(N),YIS(N),ZIS(N),N=1,NIP)
100   FORMAT(1P3E15.5)
CVAX  IF(IUNIT.NE.LUN) CLOSE(UNIT=IUNIT)                                  VAX
      RETURN
      END


      SUBROUTINE SELECT(NUM,MENU,NSEL)
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE SELECT-ACCEPTS SELECTION FROM MAIN MENUS                   C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE CALLED BY EXPERT, THIST                                    C
C                                                                        C
```

```
C*********************************************************************C
C                                                                     C
C  SUBROUTINES CALLED: HELP                                           C
C                                                                     C
C*********************************************************************C
C                                                                     C
C  INPUT ARGUMENTS: NUM,MENU                                          C
C                                                                     C
C  OUTPUT ARGUMENTS: NSEL                                             C
C                                                                     C
C*********************************************************************C
C                                                                     C
C  IMPORTANT INTERNAL VARIABLES:                                      C
C     NUM =NUMBER OF OPTIONS IN MENU                                  C
C     MENU=OPTION LIST NUMBER                                         C
C     NSEL=SELECTION MADE                                             C
C                                                                     C
C*********************************************************************C
C                                                                     C
C  ARRAY DIMENSIONS                                                   C
C     ICHAR(15)                                                       C
C                                                                     C
C*********************************************************************C
C                                                                     C
      DIMENSION ICHAR(15)
      DATA ICHAR/'1','2','3','4','5','6','7','8','9','0',
     *'H',' ','R','T','S'/
      COMMON/UNIT/ILUN,LUN,IDSK
 5    WRITE(LUN,10)
 10   FORMAT('$','SELECTION: ')
      READ(ILUN,20,ERR=5) IOPT
 20   FORMAT(A1)
      DO 30 NSEL=1,15
        IF (IOPT.EQ.ICHAR(NSEL)) GO TO 50
 30   CONTINUE
      WRITE(LUN,40) IOPT
 40   FORMAT(' ',/T6,' **** ',A1,' IS AN INVALID RESPONSE ****')
      NSEL=12
 50   IF(NSEL.GT.10) GO TO 60
      IF(NSEL.GT.NUM) GO TO 30
      IF(NSEL.LT.NUM) GO TO 70
 60   IF(NSEL.EQ.11.OR.NSEL.EQ.NUM) THEN
         CALL HELP(MENU,NUM)
         GO TO 5
      ELSE
         NSEL=11-NSEL
      ENDIF
 70   CONTINUE
      RETURN
      END

      SUBROUTINE SELNCP(MENU,NDP,NCPDFT,NCP)
C                                                                     C
C*********************************************************************C
C                                                                     C
C  SUBROUTINE SELNCP-ACCEPTS SELECTION OF NCP BY READING AS AN ALPHA- C
C                    STRING AND CONVERTING TO AN INTEGER              C
C                                                                     C
C*********************************************************************C
C                                                                     C
C  SUBROUTINE CALLED BY EXPERT                                        C
C                                                                     C
C*********************************************************************C
C                                                                     C
C  SUBROUTINES CALLED: HELP                                           C
```

```
C                                                                      C
C**********************************************************************C
C                                                                      C
C  INPUT ARGUMENTS: MENU,NDP,NCPDFT                                    C
C                                                                      C
C  OUTPUT ARGUMENTS: NCP                                               C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  IMPORTANT INTERNAL VARIABLES:                                       C
C     IFLAG =INDICATES VALIDITY OF THE RESPONSE                        C
C            0=VALID                                                   C
C            1=INVALID                                                 C
C     NCPDFT=DEFAULT VALUE OF NCP (NUMBER OF CLOSEST POINTS USED FOR   C
C            EACH INTERPOLATION                                        C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  ARRAY DIMENSIONS                                                    C
C     ICHAR(15)                                                        C
C                                                                      C
C**********************************************************************C
C                                                                      C
      DIMENSION ICHAR(15)
      DATA ICHAR/'1','2','3','4','5','6','7','8','9','0',
     * ' ','D','M','H','T'/
      COMMON/UNIT/ILUN,LUN,IDSK
 10   WRITE(LUN,20)
 20   FORMAT(' ',/' NUMBER OF CLOSEST DATA POINTS (NCP) TO BE USED'/
     *   '$ FOR INTERPOLATION OF EACH POINT (OR TYPE HELP): ')
      READ(ILUN,30,ERR=10) IOPT1,IOPT2
 30   FORMAT(2A1)
      IFLAG=0
      DO 40 I=1,15
        IF(IOPT1.EQ.ICHAR(I)) THEN
          II=I-11
          GO TO (110,120,130,180) II
          GO TO 50
        ENDIF
 40   CONTINUE
      IFLAG=1
 50   DO 60 J=1,15
        IF(IOPT2.EQ.ICHAR(J)) GO TO 70
 60   CONTINUE
      IFLAG=1
 70   IF(IFLAG.EQ.1) GO TO 160
      IF(I.EQ.10) I=0
      IF(J.EQ.10) J=0
      IF(I.LT.10.AND.J.LT.10) GO TO 90
      IF(I.LT.10.AND.J.EQ.11) GO TO 100
      I=I-10
      GO TO (80,110,120,130,180) I
      GO TO 160
 80   J=J-10
      GO TO (110,110,120,130,180) J
      IOPT1=1H^
C
C  NCP = SELECTED VALUE
C
      GO TO 160
 90   NCP=I*10+J
      GO TO 140
100   NCP=I
      GO TO 140
C
C  NCP = DEFAULT VALUE
```

```
C
110    NCP=MIN0(NCPDFT,NDP-1)
       GO TO 140
C
C  NCP = MAXIMUM ALLOWABLE VALUE
C
120    NCP=MIN0(25,NDP-1)
       GO TO 140
130    CALL HELP(MENU,NCPDFT)
       GO TO 10
140    IF(NCP.LT.2.OR.NCP.GT.NDP-1.OR.NCP.GT.25) THEN
         WRITE(LUN,150)
150      FORMAT(' ',/T10,'**** NCP RANGE ERROR.  TYPE HELP. ****'/)
         GO TO 10
       ELSE
         GO TO 190
       ENDIF
160    WRITE(LUN,170) IOPT1,IOPT2
170    FORMAT(' ',/T10,' **** ',2A1,' IS AN INVALID RESPONSE ****'/)
       GO TO 10
180    NCP=-1
190    CONTINUE
       RETURN
       END

       SUBROUTINE THIST(NSELT)
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE THIST-CALLS THE DISSPLA SUBROUTINES THAT CREATE PLOTS     C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINE CALLED BY EXPERT                                           C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINES CALLED: EXTRM, SELECT                                     C
C                      OPEN, CLOSE (SUPERSET)                            C
C                      DISKN (CRAY)                                      C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SYSTEM SUBROUTINES: DISSPLA                                           C
C                                                                        C
C************************************************************************C
C                                                                        C
C  INPUT ARGUMENTS: NONE                                                 C
C                                                                        C
C  OUTPUT ARGUMENTS: NSELT                                               C
C                                                                        C
C************************************************************************C
C                                                                        C
C  IMPORTANT INTERNAL VARIABLES:                                         C
C     CURVE1 = ARRAY CONTAINING FUNCTION VALUES                          C
C     CURVE2 = ARRAY CONTAINING FUNCTION VALUES                          C
C     TIME1  = ARRAY CONTAINING TIME STEPS                               C
C     TIME2  = ARRAY CONTAINING TIME STEPS                               C
C     NPT1   = NUMBER OF TIME STEPS IN TIME1                             C
C     NPT2   = NUMBER OF TIME STEPS IN TIME2                             C
C                                                                        C
C************************************************************************C
C                                                                        C
C  ARRAY DIMENSIONS                                                      C
C     CURVE1,TIME1(NPT1)                                                 C
```

```
C       CURVE2,TIME2(NPT2)                                                     C
C                                                                              C
C******************************************************************************C
C                                                                              C
        DIMENSION CURVE1(2500),CURVE2(2500),TIME1(2500),TIME2(2500)
        INTEGER FILE1(2),FILE2(2),LSTRG1(4),LSTRG2(4),TYPE1(2),TYPE2(2)
        INTEGER DRCTN1,DRCTN2,SPACE,IPKRAY(50)
        COMMON/UNIT/ILUN,LUN,IDSK
        COMMON NAMOLD                                                          CRAY
        DATA SPACE/' '/
 10     WRITE(LUN,20)
 20     FORMAT(' '/' OPTIONS:'/
       1         '   1.  ONE CURVE'/
       2         '   2.  TWO CURVES--SAME TYPE'/
       3         '   3.  TWO CURVES--DIFFERENT TYPES'/
       4         '   4.  HELP')
        NUM=4
        MENU=6
        CALL SELECT(NUM,MENU,NSEL)
        NSELT=1
        IFLAG=NSEL
        GO TO (30,50,50) IFLAG
        NSELT=-NSEL
        GO TO (10,300,300,300) NSELT
 30     WRITE(LUN,40)
 40     FORMAT(' ','ENTER FILENAME TO BE PLOTTED:')
        READ(ILUN,70) FILE1
        GO TO 80
 50     WRITE(LUN,60)
 60     FORMAT(' ','ENTER FILENAMES TO BE PLOTTED:')
        READ(ILUN,70) FILE1
        READ(ILUN,70) FILE2
 70     FORMAT(2A4)
 80     CONTINUE
C
C  READ FIRST FILE
C
        CALL DISKN(FILE1)                                                      CRAY
        IUNIT=IDSK                                                             CRAY
CSSET   CALL OPEN(FILE1,IUNIT)                                                 SSET
 90     READ(IUNIT,100) FILE1
100     FORMAT(1X,2A4)
        WRITE(LUN,110) FILE1
110     FORMAT(4X,'FILE1= ',2A4)
        READ(IUNIT,120) TYPE1
120     FORMAT(1X,2A6)
        WRITE(LUN,130) TYPE1
130     FORMAT(4X,'TYPE1= ',2A6)
        READ(IUNIT,140) DRCTN1
140     FORMAT(1X,A1)
        WRITE(LUN,150) DRCTN1
150     FORMAT(4X,'DRCTN1= ',A1)
        READ(IUNIT,160) NPT1
160     FORMAT(I5)
        WRITE(LUN,170)NPT1
170     FORMAT(4X,'NPT1= ',I5)
CSSET   READ(IUNIT,180)  (CURVE1(I),I=1,NPT1)                                  SSET
CSSET   READ(IUNIT,180)  (TIME1(I),I=1,NPT1)                                   SSET
C180     FORMAT(5U)                                                            SSET
        READ(IUNIT,*)  (CURVE1(I),I=1,NPT1)                                    CRAY
        READ(IUNIT,*)  (TIME1(I),I=1,NPT1)                                     CRAY
CSSET   CALL CLOSE                                                             SSET
        ENCODE(23,190,LSTRG1) DRCTN1,SPACE,TYPE1,SPACE,FILE1
190     FORMAT(A1,A1,2A6,A1,2A4)
        CALL EXTRM(CURVE1,NPT1,YMAX1,YMIN1)
        IF(IFLAG.EQ.1) GO TO 250
```

```
C
C  READ SECOND FILE
C
      CALL DISKN(FILE2)                                                 CRAY
      IUNIT=IDSK                                                        CRAY
CSSET CALL OPEN(FILE2,IUNIT)                                            SSET
      READ(IUNIT,100) FILE2
      WRITE(LUN,200) FILE2
200   FORMAT(4X,'FILE2= ',2A4)
      READ(IUNIT,120) TYPE2
      WRITE(LUN,210) TYPE2
210   FORMAT(4X,'TYPE2= ',2A6)
      READ(IUNIT,140) DRCTN2
      WRITE(LUN,220) DRCTN2
220   FORMAT(4X,'DRCTN2= ',A1)
      READ(IUNIT,160) NPT2
      WRITE(LUN,230) NPT2
230   FORMAT(4X,'NPT2= ',I5)
CSSET READ(IUNIT,180) (CURVE2(I),I=1,NPT2)                              SSET
CSSET READ(IUNIT,180) (TIME2(I),I=1,NPT2)                               SSET
      READ(IUNIT,*) (CURVE2(I),I=1,NPT2)                                CRAY
      READ(IUNIT,*) (TIME2(I),I=1,NPT2)                                 CRAY
CSSET CALL CLOSE                                                        SSET
      ENCODE(23,190,LSTRG2) DRCTN2,SPACE,TYPE2,SPACE,FILE2
      CALL EXTRM(CURVE2,NPT2,YMAX2,YMIN2)
      IF(IFLAG.EQ.3) GO TO 240
C
C  DETERMINE RANGE OF AXES--TWO FILES
C
      YMAX=AMAX1(YMAX1,YMAX2)
      YMAX=YMAX+.05*ABS(YMAX)
      YORIG=AMIN1(YMIN1,YMIN2)
      YORIG=YORIG-.05*ABS(YORIG)
240   XMAX=AMAX1(TIME1(NPT1),TIME2(NPT2))
      IF(IFLAG.EQ.2) GO TO 260
250   YMAX=YMAX1+.05*ABS(YMAX1)
      YORIG=YMIN1-.05*ABS(YMIN1)
      IF(IFLAG.EQ.3) GO TO 260
      XMAX=TIME1(NPT1)
C
C  CALL DISSPLA ROUTINES
C
260   CALL BGNPL(1)
      CALL PROMPT(0,0)
      CALL TITLE(' ',-1,'TIME (SEC)',10,TYPE1,12,9.0,7.5)
      CALL XTICKS(5)
      CALL YTICKS(5)
      CALL HEIGHT(.20)
      CALL GRAF(0,'SCALE',XMAX,YORIG,'SCALE',YMAX)
      CALL RESET('HEIGHT')
      MAXLIN=LINEST(IPKRAY,50,40)
      CALL LINES(LSTRG1,IPKRAY,1)
      CALL MARKER(4)
      CALL CURVE(TIME1,CURVE1,NPT1,4)
      NLINES=1
      GO TO (290,280,270) IFLAG
C
C  DETERMINE RANGE OF AXIS--ONE FILE
C
270   YMAX=YMAX2+.05*ABS(YMAX2)
      YORIG=YMIN2-.05*ABS(YMIN2)
      CALL HEIGHT(.20)
      CALL YGRAXS(YORIG,'SCALE',YMAX,7.5,TYPE2,-12,9.0,0.0)
      CALL RESET('HEIGHT')
280   CALL LINES(LSTRG2,IPKRAY,2)
      CALL CHNDOT
```

```
      CALL MARKER(5)
      CALL CURVE(TIME2,CURVE2,NPT2,4)
      CALL RESET('CHNDOT')
      NLINES=2
C
C  LEGEND
C
290   CALL LEGEND(IPKRAY,NLINES,1.7,0.6)
      XL=XMESS(LSTRG1,22)
      X1=1.5
      Y1=0.45
      X2=X1+XL+0.8
      Y2=1.25
      CALL STRTPT(X1,Y1)
      CALL CONNPT(X1,Y2)
      CALL CONNPT(X2,Y2)
      CALL CONNPT(X2,Y1)
      CALL CONNPT(X1,Y1)
      CALL ENDPL(0)
      CALL DONEPL
300   NSELT=NSELT-1
      RETURN
      END

CSSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET  SSET
C                                                                       SSET
C     PROGRAM THTEST                                                    SSET
C                                                                      CSSET
C**********************************************************************CSSET
C                                                                      CSSET
C  PROGRAM THTEST-THIS PROGRAM IS A DRIVER FOR SUBROUTINE THIST FOR    CSSET
C                 THE SUPERSET.  THE DISSPLA ROUTINES ARE NOT AVAIL-   CSSET
C                 ABLE ON THE VAX, SO SUBROUTINE THIST MUST BE         CSSET
C                 EXECUTED ON THE SUPERSET.  NEED TO LINK THTEST,      CSSET
C                 THIST, EXTRM, CLOSE, OPEN, AND SELECT.               CSSET
C                                                                      CSSET
C**********************************************************************CSSET
C                                                                      CSSET
C  SUBROUTINES CALLED: THIST                                           CSSET
C                                                                      CSSET
C**********************************************************************CSSET
C                                                                      CSSET
C     DATA ILUN/5/,LUN/6/                                               SSET
C     COMMON/UNIT/ILUN,LUN,IDSK                                         SSET
C     CALL THIST(NSELT)                                                 SSET
C     STOP                                                              SSET
C     END                                                               SSET
CSSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET SSET  SSET
```

```
C***********************************************************************C
C                                                                       C
C                 INTERPOLATION SUBROUTINE PACKAGE                      C
C                                                                       C
C         THE FOLLOWING SUBROUTINES PERFORM THE INTERPOLATION ON        C
C     THE EXPERIMENTAL DATA.  THE PACKAGE CONSISTS OF EIGHT SUB-        C
C     ROUTINES AND A FUNCTION.  TWO SUBROUTINES, IDBVIP AND IDSFFT,     C
C     ARE THE MAIN SUBROUTINES.  THE OTHERS ARE CALLED BY THESE         C
C     TWO.  THEY WERE WRITTEN BY HIROSHI AKIMA AND ARE INCLUDED AS      C
C     PART OF BOTH THE A.C.M. LIBRARY AND I.M.S.L. LIBRARY.  FOR        C
C     REFERENCE SEE A.C.M. TRANSACTIONS ON MATHEMATICAL SOFTWARE,       C
C     VOL. 4, NO. 2, JUNE 1978, PAGES 148-162.  SOME MODIFICATIONS      C
C     WERE MADE BY L. ALLEN ADAMS FOR USE WITH THE PROGRAM 'EXPERT'.    C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINES:                                                         C
C     IDBVIP =PERFORMS BIVARIATE INTERPOLATION WHEN THE PROJECTIONS     C
C             OF THE DATA POINTS IN THE X-Y PLANE ARE IRREGULARLY       C
C             DISTRIBUTED IN THE PLANE.                                 C
C     IDSFFT =PERFORMS SMOOTH FITTING WHEN THE PROJECTIONS OF THE       C
C             DATA POINTS IN THE X-Y PLANE ARE IRREGULARLY DISTRI-      C
C             BUTED IN THE PLANE.                                       C
C     IDCLDP =SELECTS SEVERAL DATA POINTS THAT ARE CLOSEST TO EACH      C
C             OF THE DATA POINTS.                                       C
C     IDGRID =ORGANIZES GRID POINTS FOR SURFACE FITTING.                C
C     IDLCTN =DETERMINES TO WHAT TRIANGLE A POINT BELONGS.              C
C     IDPDRV =ESTIMATES PARTIAL DERIVATIVES AT DATA POINTS.             C
C     IDPTIP =PERFORMS PUNCTUAL INTERPOLATION.                          C
C     IDTANG =PERFORMS TRIANGULATION.                                   C
C                                                                       C
C  FUNCTION:                                                            C
C     IDXCHG =DETERMINES NECESSITY OF EXCHANGE OF TRIANGLES.            C
C                                                                       C
C***********************************************************************C
C                                                                       C
C                                                                       C


      SUBROUTINE  IDBVIP(MD,NCP,NDP,XD,YD,ZD,NIP,XI,YI,ZI,
     1                   IWK,IWKBDM,WK,IERR)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE IDBVIP- PERFORMS BIVARIATE INTERPOLATION WHEN THE PRO-    C
C                     JECTIONS OF THE DATA POINTS IN THE X-Y PLANE      C
C                     ARE IRREGULARLY DISTRIBUTED IN THE PLANE.         C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE CALLED BY EXPERT                                          C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINES CALLED:  IDCLDP, IDLCTN, IDPDRV, IDPTIP, IDTANG          C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  INPUT PARAMETERS:                                                    C
C     MD  = MODE OF COMPUTATION (MUST BE 1, 2, OR 3),                   C
C         = 1 FOR NEW NCP AND/OR NEW XD-YD,                             C
C         = 2 FOR OLD NCP, OLD XD-YD, NEW XI-YI,                        C
C         = 3 FOR OLD NCP, OLD XD-YD, OLD XI-YI,                        C
C     NCP = NUMBER OF ADDITIONAL DATA POINTS USED FOR ESTI-             C
```

```
C           MATING PARTIAL DERIVATIVES AT EACH DATA POINT                    C
C           (MUST BE 2 OR GREATER, BUT SMALLER THAN NDP),                    C
C     NDP = NUMBER OF DATA POINTS (MUST BE 4 OR GREATER),                    C
C     XD  = ARRAY OF DIMENSION NDP CONTAINING THE X                          C
C           COORDINATES OF THE DATA POINTS,                                  C
C     YD  = ARRAY OF DIMENSION NDP CONTAINING THE Y                          C
C           COORDINATES OF THE DATA POINTS,                                  C
C     ZD  = ARRAY OF DIMENSION NDP CONTAINING THE Z                          C
C           COORDINATES OF THE DATA POINTS,                                  C
C     NIP = NUMBER OF OUTPUT POINTS AT WHICH INTERPOLATION                   C
C           IS TO BE PERFORMED (MUST BE 1 OR GREATER),                       C
C     XI  = ARRAY OF DIMENSION NIP CONTAINING THE X                          C
C           COORDINATES OF THE OUTPUT POINTS,                                C
C     YI  = ARRAY OF DIMENSION NIP CONTAINING THE Y                          C
C           COORDINATES OF THE OUTPUT POINTS.                                C
C     IWKBDM = DIMENSION OF ARRAY IWK.                                       C
C                                                                            C
C****************************************************************************C
C                                                                            C
C  OUTPUT PARAMETERS:                                                        C
C     ZI  = ARRAY OF DIMENSION NIP WHERE INTERPOLATED Z                      C
C           VALUES ARE TO BE STORED.                                         C
C     IERR= INDICATES ERROR CONDITION.                                       C
C           0=NO ERROR                                                       C
C           1=ERROR                                                          C
C                                                                            C
C****************************************************************************C
C                                                                            C
C  OTHER PARAMETERS:                                                         C
C     IWK = INTEGER ARRAY OF DIMENSION                                       C
C              MAX0(31,27+NCP)*NDP+NIP                                       C
C           USED INTERNALLY AS A WORK AREA,                                  C
C     WK  = ARRAY OF DIMENSION 8*NDP USED INTERNALLY AS A                    C
C           WORK AREA.                                                       C
C                                                                            C
C****************************************************************************C
C                                                                            C
C   NOTE:                                                                    C
C      THE VERY FIRST CALL TO THIS SUBROUTINE AND THE CALL WITH A NEW        C
C      NCP VALUE, A NEW NDP VALUE, AND/OR NEW CONTENTS OF THE XD AND         C
C      YD ARRAYS MUST BE MADE WITH MD=1.  THE CALL WITH MD=2 MUST BE         C
C      PRECEDED BY ANOTHER CALL WITH THE SAME NCP AND NDP VALUES AND         C
C      WITH THE SAME CONTENTS OF THE XD AND YD ARRAYS.  THE CALL WITH        C
C      MD=3 MUST BE PRECEDED BY ANOTHER CALL WITH THE SAME NCP, NDP,         C
C      AND NIP VALUES AND WITH THE SAME CONTENTS OF THE XD, YD, XI,          C
C      AND YI ARRAYS.  BETWEEN THE CALL WITH MD=2 OR MD=3 AND ITS            C
C      PRECEDING CALL, THE IWK AND WK ARRAYS MUST NOT BE DISTURBED.          C
C      USE OF A VALUE BETWEEN 3 AND 5 (INCLUSIVE) FOR NCP IS RECOM-          C
C      MENDED UNLESS THERE ARE EVIDENCES THAT DICTATE OTHERWISE.             C
C                                                                            C
C****************************************************************************C
C                                                                            C
C DECLARATION STATEMENTS
      DIMENSION   XD(NDP),YD(NDP),ZD(NDP),XI(NIP),YI(NIP),
     1            ZI(NIP),IWK(IWKBDM),WK(8*NDP)
      COMMON/IDLC/ITIPV,DMMY1(13)
      COMMON/IDPI/ITPV,DMMY(27)
      COMMON/UNIT/ILUN,LUN,IDSK
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES.
C (FOR MD=1,2,3)
   10 MD0=MD
      NCP0=NCP
      NDP0=NDP
      NIP0=NIP
C ERROR CHECK.  (FOR MD=1,2,3)
   20 IF(MD0.LT.1.OR.MD0.GT.3)          GO TO 90
```

```
      IF(NCP0.LT.2.OR.NCP0.GE.NDP0)        GO TO 90
      IF(NDP0.LT.4)                        GO TO 90
      IF(NIP0.LT.1)                        GO TO 90
      IF(MD0.GE.2)         GO TO 21
      IWK(1)=NCP0
      IWK(2)=NDP0
      GO TO 22
   21 NCPPV=IWK(1)
      NDPPV=IWK(2)
      IF(NCP0.NE.NCPPV)    GO TO 90
      IF(NDP0.NE.NDPPV)    GO TO 90
   22 IF(MD0.GE.3)         GO TO 23
      IWK(3)=NIP
      GO TO 30
   23 NIPPV=IWK(3)
      IF(NIP0.NE.NIPPV)    GO TO 90
C ALLOCATION OF STORAGE AREAS IN THE IWK ARRAY.  (FOR MD=1,2,3)
   30 JWIPT=16
      JWIWL=6*NDP0+1
      JWIWK=JWIWL
      JWIPL=24*NDP0+1
      JWIWP=30*NDP0+1
      JWIPC=27*NDP0+1
      JWIT0=MAX0(31,27+NCP0)*NDP0
C TRIANGULATES THE X-Y PLANE.  (FOR MD=1)
   40 IF(MD0.GT.1)   GO TO 41
      CALL IDTANG(NDP0,XD,YD,NT,IWK(JWIPT),NL,IWK(JWIPL),
     1            IWK(JWIWL),IWK(JWIWP),WK,IERR)
      IWK(5)=NT
      IWK(6)=NL
      IF(NT.EQ.0)    RETURN
      GO TO 50
 41   NT=IWK(5)
      NL=IWK(6)
C DETERMINES NCP POINTS CLOSEST TO EACH DATA POINT.  (FOR MD=1)
   50 IF(MD0.GT.1)   GO TO 60
      CALL IDCLDP(NDP0,XD,YD,NCP0,IWK(JWIPC),IERR)
      IF(IWK(JWIPC).EQ.0)       RETURN
C LOCATES ALL POINTS AT WHICH INTERPOLATION IS TO BE PERFORMED.
C (FOR MD=1,2)
   60 IF(MD0.EQ.3)   GO TO 70
      ITIPV=0
      JWIT=JWIT0
      DO 61  IIP=1,NIP0
        JWIT=JWIT+1
        CALL IDLCTN(NDP0,XD,YD,NT,IWK(JWIPT),NL,IWK(JWIPL),
     1             XI(IIP),YI(IIP),IWK(JWIT),IWK(JWIWK),WK)
   61 CONTINUE
C ESTIMATES PARTIAL DERIVATIVES AT ALL DATA POINTS.
C (FOR MD=1,2,3)
   70 CALL IDPDRV(NDP0,XD,YD,ZD,NCP0,IWK(JWIPC),WK)
C INTERPOLATES THE ZI VALUES.  (FOR MD=1,2,3)
   80 ITPV=0
      JWIT=JWIT0
      DO 81  IIP=1,NIP0
        JWIT=JWIT+1
        CALL IDPTIP(XD,YD,ZD,NDP,NT,IWK(JWIPT),NL,IWK(JWIPL),WK,
     1               IWK(JWIT),XI(IIP),YI(IIP),ZI(IIP))
   81 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) MD0,NCP0,NDP0,NIP0
      IERR=1
      RETURN
C FORMAT STATEMENT FOR ERROR MESSAGE
 2090 FORMAT(1X/41H ***   IMPROPER INPUT PARAMETER VALUE(S)./
```

```
     1    7H   MD =,I4,10X,5HNCP =,I6,10X,5HNDP =,I6,
     2    10X,5HNIP =,I6/
     3    35H ERROR DETECTED IN ROUTINE   IDBVIP/)
      END

      SUBROUTINE  IDCLDP(NDP,XD,YD,NCP,IPC,IERR)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE IDCLDP-SELECTS SEVERAL DATA POINTS THAT ARE CLOSEST TO    C
C                    EACH OF THE DATA POINTS                            C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE CALLED BY IDBVIP, IDSFFT                                  C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  INPUT PARAMETERS:                                                    C
C      NDP = NUMBER OF DATA POINTS,                                     C
C      XD,YD = ARRAYS OF DIMENSION NDP CONTAINING THE X AND Y           C
C              COORDINATES OF THE DATA POINTS,                          C
C      NCP = NUMBER OF DATA POINTS CLOSEST TO EACH DATA                 C
C              POINTS.                                                  C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  OUTPUT PARAMETER:                                                    C
C      IPC = INTEGER ARRAY OF DIMENSION NCP*NDP, WHERE THE              C
C              POINT NUMBERS OF NCP DATA POINTS CLOSEST TO              C
C              EACH OF THE NDP DATA POINTS ARE TO BE STORED.            C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  NOTE:                                                                C
C       THIS SUBROUTINE ARBITRARILY SETS A RESTRICTION THAT NCP MUST    C
C       NOT EXCEED 25.                                                  C
C                                                                       C
C***********************************************************************C
C                                                                       C
C DECLARATION STATEMENTS
      DIMENSION   XD(NDP),YD(NDP),IPC(NCP*NDP)
      DIMENSION   DSQ0(25),IPC0(25)
      COMMON/UNIT/ILUN,LUN,IDSK
      DATA  NCPMX/25/
C STATEMENT FUNCTION
      DSQF(U1,V1,U2,V2)=(U2-U1)**2+(V2-V1)**2
C PRELIMINARY PROCESSING
   10 NDP0=NDP
      NCP0=NCP
      IF(NDP0.LT.2)  GO TO 90
      IF(NCP0.LT.1.OR.NCP0.GT.NCPMX.OR.NCP0.GE.NDP0)    GO TO 90
C CALCULATION
   20 DO 59  IP1=1,NDP0
C - SELECTS NCP POINTS.
        X1=XD(IP1)
        Y1=YD(IP1)
        J1=0
        DSQMX=0.0
        DO 22  IP2=1,NDP0
          IF(IP2.EQ.IP1)  GO TO 22
          DSQI=DSQF(X1,Y1,XD(IP2),YD(IP2))
          J1=J1+1
          DSQ0(J1)=DSQI
          IPC0(J1)=IP2
```

```
          IF(DSQI.LE.DSQMX)     GO TO 21
          DSQMX=DSQI
          JMX=J1
   21     IF(J1.GE.NCP0)  GO TO 23
   22   CONTINUE
   23   IP2MN=IP2+1
        IF(IP2MN.GT.NDP0)       GO TO 30
        DO 25  IP2=IP2MN,NDP0
          IF(IP2.EQ.IP1)  GO TO 25
          DSQI=DSQF(X1,Y1,XD(IP2),YD(IP2))
          IF(DSQI.GE.DSQMX)     GO TO 25
          DSQ0(JMX)=DSQI
          IPC0(JMX)=IP2
          DSQMX=0.0
          DO 24  J1=1,NCP0
            IF(DSQ0(J1).LE.DSQMX)    GO TO 24
            DSQMX=DSQ0(J1)
            JMX=J1
   24     CONTINUE
   25   CONTINUE
C - CHECKS IF ALL THE NCP+1 POINTS ARE COLLINEAR.
   30   IP2=IPC0(1)
        DX12=XD(IP2)-X1
        DY12=YD(IP2)-Y1
        DO 31  J3=2,NCP0
          IP3=IPC0(J3)
          DX13=XD(IP3)-X1
          DY13=YD(IP3)-Y1
          IF((DY13*DX12-DX13*DY12).NE.0.0)     GO TO 50
   31   CONTINUE
C - SEARCHES FOR THE CLOSEST NONCOLLINEAR POINT.
   40   NCLPT=0
        DO 43  IP3=1,NDP0
          IF(IP3.EQ.IP1)        GO TO 43
          DO 41  J4=1,NCP0
            IF(IP3.EQ.IPC0(J4))      GO TO 43
   41     CONTINUE
          DX13=XD(IP3)-X1
          DY13=YD(IP3)-Y1
          IF((DY13*DX12-DX13*DY12).EQ.0.0)     GO TO 43
          DSQI=DSQF(X1,Y1,XD(IP3),YD(IP3))
          IF(NCLPT.EQ.0)        GO TO 42
          IF(DSQI.GE.DSQMN)     GO TO 43
   42     NCLPT=1
          DSQMN=DSQI
          IP3MN=IP3
   43   CONTINUE
        IF(NCLPT.EQ.0)    GO TO 91
        DSQMX=DSQMN
        IPC0(JMX)=IP3MN
C - REPLACES THE LOCAL ARRAY FOR THE OUTPUT ARRAY.
   50   J1=(IP1-1)*NCP0
        DO 51  J2=1,NCP0
          J1=J1+1
          IPC(J1)=IPC0(J2)
   51   CONTINUE
   59 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090)
      GO TO 92
   91 WRITE (LUN,2091)
   92 WRITE (LUN,2092)   NDP0,NCP0
      IPC(1)=0
      IERR=1
      RETURN
```

```
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/41H ***   IMPROPER INPUT PARAMETER VALUE(S).)
 2091 FORMAT(1X/33H ***   ALL COLLINEAR DATA POINTS.)
 2092 FORMAT(8H   NDP =,I5,5X,5HNCP =,I5/
     1   35H ERROR DETECTED IN ROUTINE   IDCLDP/)
      END

      SUBROUTINE IDGRID(XD, YD, NDP, NT, IPT, NL, IPL, NXI, NYI, XI,
     *  YI, NGP, IGP)
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE IDGRID-ORGANIZES GRID POINTS FOR SURFACE FITTING BY      C
C                    SORTING THEM IN ASCENDING ORDER OF TRIANGLE       C
C                    NUMBERS AND OF THE BORDER LINE SEGMENT NUMBER.    C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE CALLED BY IDSFFT                                         C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  INPUT PARAMETERS:                                                   C
C     XD,YD = ARRAYS OF DIMENSION NDP CONTAINING THE X AND Y           C
C             COORDINATES OF THE DATA POINTS, WHERE NDP IS THE         C
C             NUMBER OF THE DATA POINTS,                               C
C     NT  = NUMBER OF TRIANGLES,                                       C
C     IPT = INTEGER ARRAY OF DIMENSION 3*NT CONTAINING THE             C
C           POINT NUMBERS OF THE VERTEXES OF THE TRIANGLES,            C
C     NL  = NUMBER OF BORDER LINE SEGMENTS,                            C
C     IPL = INTEGER ARRAY OF DIMENSION 3*NL CONTAINING THE             C
C           POINT NUMBERS OF THE END POINTS OF THE BORDER              C
C           LINE SEGMENTS AND THEIR RESPECTIVE TRIANGLE                C
C           NUMBERS,                                                   C
C     NXI = NUMBER OF GRID POINTS IN THE X COORDINATE,                 C
C     NYI = NUMBER OF GRID POINTS IN THE Y COORDINATE,                 C
C     XI,YI = ARRAYS OF DIMENSION NXI AND NYI CONTAINING                C
C           THE X AND Y COORDINATES OF THE GRID POINTS,                C
C           RESPECTIVELY.                                              C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  OUTPUT PARAMETERS:                                                  C
C     NGP = INTEGER ARRAY OF DIMENSION 2*(NT+2*NL) WHERE THE           C
C           NUMBER OF GRID POINTS THAT BELONG TO EACH OF THE           C
C           TRIANGLES OR OF THE BORDER LINE SEGMENTS ARE TO            C
C           BE STORED,                                                 C
C     IGP = INTEGER ARRAY OF DIMENSION NXI*NYI WHERE THE               C
C           GRID POINT NUMBERS ARE TO BE STORED IN ASCENDING           C
C           ORDER OF THE TRIANGLE NUMBER AND THE BORDER LINE           C
C           SEGMENT NUMBER.                                            C
C                                                                      C
C**********************************************************************C
C                                                                      C
C DECLARATION STATEMENTS
      DIMENSION XD(NDP), YD(NDP), IPT(3*NT), IPL(3*NL), XI(NXI),
     *  YI(NYI), NGP(2*NT+4*NL), IGP(NXI*NYI)
C STATEMENT FUNCTIONS
      SIDE(U1,V1,U2,V2,U3,V3) = (U1-U3)*(V2-V3) - (V1-V3)*(U2-U3)
      SPDT(U1,V1,U2,V2,U3,V3) = (U1-U2)*(U3-U2) + (V1-V2)*(V3-V2)
C PRELIMINARY PROCESSING
      NT0 = NT
      NL0 = NL
      NXI0 = NXI
      NYI0 = NYI
```

```
      NXINYI = NXI0*NYI0
      XIMN = AMIN1(XI(1),XI(NXI0))
      XIMX = AMAX1(XI(1),XI(NXI0))
      YIMN = AMIN1(YI(1),YI(NYI0))
      YIMX = AMAX1(YI(1),YI(NYI0))
C DETERMINES GRID POINTS INSIDE THE DATA AREA.
      JNGP0 = 0
      JNGP1 = 2*(NT0+2*NL0) + 1
      JIGP0 = 0
      JIGP1 = NXINYI + 1
      DO 160 IT0=1,NT0
        NGP0 = 0
        NGP1 = 0
        IT0T3 = IT0*3
        IP1 = IPT(IT0T3-2)
        IP2 = IPT(IT0T3-1)
        IP3 = IPT(IT0T3)
        X1 = XD(IP1)
        Y1 = YD(IP1)
        X2 = XD(IP2)
        Y2 = YD(IP2)
        X3 = XD(IP3)
        Y3 = YD(IP3)
        XMN = AMIN1(X1,X2,X3)
        XMX = AMAX1(X1,X2,X3)
        YMN = AMIN1(Y1,Y2,Y3)
        YMX = AMAX1(Y1,Y2,Y3)
        INSD = 0
        DO 20 IXI=1,NXI0
          IF (XI(IXI).GE.XMN .AND. XI(IXI).LE.XMX) GO TO 10
          IF (INSD.EQ.0) GO TO 20
          IXIMX = IXI - 1
          GO TO 30
   10     IF (INSD.EQ.1) GO TO 20
          INSD = 1
          IXIMN = IXI
   20   CONTINUE
        IF (INSD.EQ.0) GO TO 150
        IXIMX = NXI0
   30   DO 140 IYI=1,NYI0
          YII = YI(IYI)
          IF (YII.LT.YMN .OR. YII.GT.YMX) GO TO 140
          DO 130 IXI=IXIMN,IXIMX
            XII = XI(IXI)
            L = 0
            IF (SIDE(X1,Y1,X2,Y2,XII,YII)) 130, 40, 50
   40       L = 1
   50       IF (SIDE(X2,Y2,X3,Y3,XII,YII)) 130, 60, 70
   60       L = 1
   70       IF (SIDE(X3,Y3,X1,Y1,XII,YII)) 130, 80, 90
   80       L = 1
   90       IZI = NXI0*(IYI-1) + IXI
            IF (L.EQ.1) GO TO 100
            NGP0 = NGP0 + 1
            JIGP0 = JIGP0 + 1
            IGP(JIGP0) = IZI
            GO TO 130
  100       IF (JIGP1.GT.NXINYI) GO TO 120
            DO 110 JIGP1I=JIGP1,NXINYI
              IF (IZI.EQ.IGP(JIGP1I)) GO TO 130
  110       CONTINUE
  120       NGP1 = NGP1 + 1
            JIGP1 = JIGP1 - 1
            IGP(JIGP1) = IZI
  130     CONTINUE
  140   CONTINUE
```

```
  150     JNGP0 = JNGP0 + 1
          NGP(JNGP0) = NGP0
          JNGP1 = JNGP1 - 1
          NGP(JNGP1) = NGP1
  160 CONTINUE
C DETERMINES GRID POINTS OUTSIDE THE DATA AREA.
C - IN SEMI-INFINITE RECTANGULAR AREA.
      DO 450 IL0=1,NL0
          NGP0 = 0
          NGP1 = 0
          IL0T3 = IL0*3
          IP1 = IPL(IL0T3-2)
          IP2 = IPL(IL0T3-1)
          X1 = XD(IP1)
          Y1 = YD(IP1)
          X2 = XD(IP2)
          Y2 = YD(IP2)
          XMN = XIMN
          XMX = XIMX
          YMN = YIMN
          YMX = YIMX
          IF (Y2.GE.Y1) XMN = AMIN1(X1,X2)
          IF (Y2.LE.Y1) XMX = AMAX1(X1,X2)
          IF (X2.LE.X1) YMN = AMIN1(Y1,Y2)
          IF (X2.GE.X1) YMX = AMAX1(Y1,Y2)
          INSD = 0
          DO 180 IXI=1,NXI0
            IF (XI(IXI).GE.XMN .AND. XI(IXI).LE.XMX) GO TO 170
            IF (INSD.EQ.0) GO TO 180
            IXIMX = IXI - 1
            GO TO 190
  170       IF (INSD.EQ.1) GO TO 180
            INSD = 1
            IXIMN = IXI
  180     CONTINUE
          IF (INSD.EQ.0) GO TO 310
          IXIMX = NXI0
  190     DO 300 IYI=1,NYI0
            YII = YI(IYI)
            IF (YII.LT.YMN .OR. YII.GT.YMX) GO TO 300
            DO 290 IXI=IXIMN,IXIMX
              XII = XI(IXI)
              L = 0
              IF (SIDE(X1,Y1,X2,Y2,XII,YII)) 210, 200, 290
  200         L = 1
  210         IF (SPDT(X2,Y2,X1,Y1,XII,YII)) 290, 220, 230
  220         L = 1
  230         IF (SPDT(X1,Y1,X2,Y2,XII,YII)) 290, 240, 250
  240         L = 1
  250         IZI = NXI0*(IYI-1) + IXI
              IF (L.EQ.1) GO TO 260
              NGP0 = NGP0 + 1
              JIGP0 = JIGP0 + 1
              IGP(JIGP0) = IZI
              GO TO 290
  260         IF (JIGP1.GT.NXINYI) GO TO 280
              DO 270 JIGP1I=JIGP1,NXINYI
                IF (IZI.EQ.IGP(JIGP1I)) GO TO 290
  270         CONTINUE
  280         NGP1 = NGP1 + 1
              JIGP1 = JIGP1 - 1
              IGP(JIGP1) = IZI
  290       CONTINUE
  300     CONTINUE
  310     JNGP0 = JNGP0 + 1
          NGP(JNGP0) = NGP0
```

```
      JNGP1 = JNGP1 - 1
      NGP(JNGP1) = NGP1
C - IN SEMI-INFINITE TRIANGULAR AREA.
      NGP0 = 0
      NGP1 = 0
      ILP1 = MOD(IL0,NL0) + 1
      ILP1T3 = ILP1*3
      IP3 = IPL(ILP1T3-1)
      X3 = XD(IP3)
      Y3 = YD(IP3)
      XMN = XIMN
      XMX = XIMX
      YMN = YIMN
      YMX = YIMX
      IF (Y3.GE.Y2 .AND. Y2.GE.Y1) XMN = X2
      IF (Y3.LE.Y2 .AND. Y2.LE.Y1) XMX = X2
      IF (X3.LE.X2 .AND. X2.LE.X1) YMN = Y2
      IF (X3.GE.X2 .AND. X2.GE.X1) YMX = Y2
      INSD = 0
      DO 330 IXI=1,NXI0
        IF (XI(IXI).GE.XMN .AND. XI(IXI).LE.XMX) GO TO 320
        IF (INSD.EQ.0) GO TO 330
        IXIMX = IXI - 1
        GO TO 340
  320   IF (INSD.EQ.1) GO TO 330
        INSD = 1
        IXIMN = IXI
  330 CONTINUE
      IF (INSD.EQ.0) GO TO 440
      IXIMX = NXI0
  340 DO 430 IYI=1,NYI0
        YII = YI(IYI)
        IF (YII.LT.YMN .OR. YII.GT.YMX) GO TO 430
        DO 420 IXI=IXIMN,IXIMX
          XII = XI(IXI)
          L = 0
          IF (SPDT(X1,Y1,X2,Y2,XII,YII)) 360, 350, 420
  350     L = 1
  360     IF (SPDT(X3,Y3,X2,Y2,XII,YII)) 380, 370, 420
  370     L = 1
  380     IZI = NXI0*(IYI-1) + IXI
          IF (L.EQ.1) GO TO 390
          NGP0 = NGP0 + 1
          JIGP0 = JIGP0 + 1
          IGP(JIGP0) = IZI
          GO TO 420
  390     IF (JIGP1.GT.NXINYI) GO TO 410
          DO 400 JIGP1I=JIGP1,NXINYI
            IF (IZI.EQ.IGP(JIGP1I)) GO TO 420
  400     CONTINUE
  410     NGP1 = NGP1 + 1
          JIGP1 = JIGP1 - 1
          IGP(JIGP1) = IZI
  420   CONTINUE
  430 CONTINUE
  440 JNGP0 = JNGP0 + 1
      NGP(JNGP0) = NGP0
      JNGP1 = JNGP1 - 1
      NGP(JNGP1) = NGP1
  450 CONTINUE
      RETURN
      END

      SUBROUTINE IDLCTN(NDP, XD, YD, NT, IPT, NL, IPL, XII, YII, ITI,
     *  IWK, WK)
```

```
C                                                                          C
C**************************************************************************C
C                                                                          C
C  SUBROUTINE IDLCTN-LOCATES A POINT, I.E., DETERMINES TO WHAT TRI-        C
C                    ANGLE A GIVEN POINT (XII,YII) BELONGS.  WHEN THE      C
C                    GIVEN POINT DOES NOT LIE INSIDE THE DATA AREA,        C
C                    THIS SUBROUTINE DETERMINES THE BORDER LINE SEG-       C
C                    MENT WHEN THE POINT LIES IN AN OUTSIDE RECTAN-        C
C                    GULAR AREA, AND TWO BORDER LINE SEGMENTS WHEN THE     C
C                    POINT LIES IN AN OUTSIDE TRIANGULAR AREA.             C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  SUBROUTINE CALLED BY IDBVIP                                             C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  INPUT PARAMETERS:                                                       C
C     NDP = NUMBER OF DATA POINTS,                                         C
C     XD,YD = ARRAYS OF DIMENSION NDP CONTAINING THE X AND Y               C
C           COORDINATES OF THE DATA POINTS,                                C
C     NT  = NUMBER OF TRIANGLES,                                           C
C     IPT = INTEGER ARRAY OF DIMENSION 3*NT CONTAINING THE                 C
C           POINT NUMBERS OF THE VERTEXES OF THE TRIANGLES,                C
C     NL  = NUMBER OF BORDER LINE SEGMENTS,                                C
C     IPL = INTEGER ARRAY OF DIMENSION 3*NL CONTAINING THE                 C
C           POINT NUMBERS OF THE END POINTS OF THE BORDER                  C
C           LINE SEGMENTS AND THEIR RESPECTIVE TRIANGLE                    C
C           NUMBERS,                                                       C
C     XII,YII = X AND Y COORDINATES OF THE POINT TO BE                     C
C           LOCATED.                                                       C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  OUTPUT PARAMETER:                                                       C
C     ITI = TRIANGLE NUMBER, WHEN THE POINT IS INSIDE THE                  C
C           DATA AREA, OR                                                  C
C           TWO BORDER LINE SEGMENT NUMBERS, IL1 AND IL2,                  C
C           CODED TO IL1*(NT+NL)+IL2, WHEN THE POINT IS                    C
C           OUTSIDE THE DATA AREA.                                         C
C                                                                          C
C**************************************************************************C
C                                                                          C
C  OTHER PARAMETERS:                                                       C
C     IWK = INTEGER ARRAY OF DIMENSION 18*NDP USED INTER-                  C
C           NALLY AS A WORK AREA,                                          C
C     WK  = ARRAY OF DIMENSION 8*NDP USED INTERNALLY AS A                  C
C           WORK AREA.                                                     C
C                                                                          C
C**************************************************************************C
C                                                                          C
C DECLARATION STATEMENTS
      DIMENSION XD(NDP), YD(NDP), IPT(3*NT), IPL(3*NL), IWK(18*NDP),
     *  WK(8*NDP)
      DIMENSION  IDSC(9)
      COMMON /IDLC/ ITIPV,XS1,XS2,YS1,YS2,NTSC(9)
C STATEMENT FUNCTIONS
      SIDE(U1,V1,U2,V2,U3,V3) = (U1-U3)*(V2-V3) - (V1-V3)*(U2-U3)
      SPDT(U1,V1,U2,V2,U3,V3) = (U1-U2)*(U3-U2) + (V1-V2)*(V3-V2)
C PRELIMINARY PROCESSING
      NDP0 = NDP
      NT0 = NT
      NL0 = NL
      NTL = NT0 + NL0
      X0 = XII
      Y0 = YII
```

```
C PROCESSING FOR A NEW SET OF DATA POINTS
      IF (ITIPV.NE.0) GO TO 80
C - DIVIDES THE X-Y PLANE INTO NINE RECTANGULAR SECTIONS.
      XMN = XD(1)
      XMX = XMN
      YMN = YD(1)
      YMX = YMN
      DO 10 IDP=2,NDP0
        XI = XD(IDP)
        YI = YD(IDP)
        XMN = AMIN1(XI,XMN)
        XMX = AMAX1(XI,XMX)
        YMN = AMIN1(YI,YMN)
        YMX = AMAX1(YI,YMX)
   10 CONTINUE
      XS1 = (XMN+XMN+XMX)/3.0
      XS2 = (XMN+XMX+XMX)/3.0
      YS1 = (YMN+YMN+YMX)/3.0
      YS2 = (YMN+YMX+YMX)/3.0
C - DETERMINES AND STORES IN THE IWK ARRAY TRIANGLE NUMBERS OF
C - THE TRIANGLES ASSOCIATED WITH EACH OF THE NINE SECTIONS.
      DO 20 ISC=1,9
        NTSC(ISC) = 0
        IDSC(ISC) = 0
   20 CONTINUE
      IT0T3 = 0
      JWK = 0
      DO 70 IT0=1,NT0
        IT0T3 = IT0T3 + 3
        I1 = IPT(IT0T3-2)
        I2 = IPT(IT0T3-1)
        I3 = IPT(IT0T3)
        XMN = AMIN1(XD(I1),XD(I2),XD(I3))
        XMX = AMAX1(XD(I1),XD(I2),XD(I3))
        YMN = AMIN1(YD(I1),YD(I2),YD(I3))
        YMX = AMAX1(YD(I1),YD(I2),YD(I3))
        IF (YMN.GT.YS1) GO TO 30
        IF (XMN.LE.XS1) IDSC(1) = 1
        IF (XMX.GE.XS1 .AND. XMN.LE.XS2) IDSC(2) = 1
        IF (XMX.GE.XS2) IDSC(3) = 1
   30   IF (YMX.LT.YS1 .OR. YMN.GT.YS2) GO TO 40
        IF (XMN.LE.XS1) IDSC(4) = 1
        IF (XMX.GE.XS1 .AND. XMN.LE.XS2) IDSC(5) = 1
        IF (XMX.GE.XS2) IDSC(6) = 1
   40   IF (YMX.LT.YS2) GO TO 50
        IF (XMN.LE.XS1) IDSC(7) = 1
        IF (XMX.GE.XS1 .AND. XMN.LE.XS2) IDSC(8) = 1
        IF (XMX.GE.XS2) IDSC(9) = 1
   50   DO 60 ISC=1,9
          IF (IDSC(ISC).EQ.0) GO TO 60
          JIWK = 9*NTSC(ISC) + ISC
          IWK(JIWK) = IT0
          NTSC(ISC) = NTSC(ISC) + 1
          IDSC(ISC) = 0
   60   CONTINUE
C - STORES IN THE WK ARRAY THE MINIMUM AND MAXIMUM OF THE X AND
C - Y COORDINATE VALUES FOR EACH OF THE TRIANGLE.
        JWK = JWK + 4
        WK(JWK-3) = XMN
        WK(JWK-2) = XMX
        WK(JWK-1) = YMN
        WK(JWK) = YMX
   70 CONTINUE
      GO TO 110
C CHECKS IF IN THE SAME TRIANGLE AS PREVIOUS.
   80 IT0 = ITIPV
```

```
      IF (IT0.GT.NT0) GO TO 90
      IT0T3 = IT0*3
      IP1 = IPT(IT0T3-2)
      X1 = XD(IP1)
      Y1 = YD(IP1)
      IP2 = IPT(IT0T3-1)
      X2 = XD(IP2)
      Y2 = YD(IP2)
      IF (SIDE(X1,Y1,X2,Y2,X0,Y0).LT.0.0) GO TO 110
      IP3 = IPT(IT0T3)
      X3 = XD(IP3)
      Y3 = YD(IP3)
      IF (SIDE(X2,Y2,X3,Y3,X0,Y0).LT.0.0) GO TO 110
      IF (SIDE(X3,Y3,X1,Y1,X0,Y0).LT.0.0) GO TO 110
      GO TO 170
C CHECKS IF ON THE SAME BORDER LINE SEGMENT.
   90 IL1 = IT0/NTL
      IL2 = IT0 - IL1*NTL
      IL1T3 = IL1*3
      IP1 = IPL(IL1T3-2)
      X1 = XD(IP1)
      Y1 = YD(IP1)
      IP2 = IPL(IL1T3-1)
      X2 = XD(IP2)
      Y2 = YD(IP2)
      IF (IL2.NE.IL1) GO TO 100
      IF (SPDT(X1,Y1,X2,Y2,X0,Y0).LT.0.0) GO TO 110
      IF (SPDT(X2,Y2,X1,Y1,X0,Y0).LT.0.0) GO TO 110
      IF (SIDE(X1,Y1,X2,Y2,X0,Y0).GT.0.0) GO TO 110
      GO TO 170
C CHECKS IF BETWEEN THE SAME TWO BORDER LINE SEGMENTS.
  100 IF (SPDT(X1,Y1,X2,Y2,X0,Y0).GT.0.0) GO TO 110
      IP3 = IPL(3*IL2-1)
      X3 = XD(IP3)
      Y3 = YD(IP3)
      IF (SPDT(X3,Y3,X2,Y2,X0,Y0).LE.0.0) GO TO 170
C LOCATES INSIDE THE DATA AREA.
C - DETERMINES THE SECTION IN WHICH THE POINT IN QUESTION LIES.
  110 ISC = 1
      IF (X0.GE.XS1) ISC = ISC + 1
      IF (X0.GE.XS2) ISC = ISC + 1
      IF (Y0.GE.YS1) ISC = ISC + 3
      IF (Y0.GE.YS2) ISC = ISC + 3
C - SEARCHES THROUGH THE TRIANGLES ASSOCIATED WITH THE SECTION.
      NTSCI = NTSC(ISC)
      IF (NTSCI.LE.0) GO TO 130
      JIWK = -9 + ISC
      DO 120 ITSC=1,NTSCI
        JIWK = JIWK + 9
        IT0 = IWK(JIWK)
        JWK = IT0*4
        IF (X0.LT.WK(JWK-3)) GO TO 120
        IF (X0.GT.WK(JWK-2)) GO TO 120
        IF (Y0.LT.WK(JWK-1)) GO TO 120
        IF (Y0.GT.WK(JWK)) GO TO 120
        IT0T3 = IT0*3
        IP1 = IPT(IT0T3-2)
        X1 = XD(IP1)
        Y1 = YD(IP1)
        IP2 = IPT(IT0T3-1)
        X2 = XD(IP2)
        Y2 = YD(IP2)
        IF (SIDE(X1,Y1,X2,Y2,X0,Y0).LT.0.0) GO TO 120
        IP3 = IPT(IT0T3)
        X3 = XD(IP3)
        Y3 = YD(IP3)
```

```
      IF (SIDE(X2,Y2,X3,Y3,X0,Y0).LT.0.0) GO TO 120
      IF (SIDE(X3,Y3,X1,Y1,X0,Y0).LT.0.0) GO TO 120
      GO TO 170
  120 CONTINUE
C LOCATES OUTSIDE THE DATA AREA.
  130 DO 150 IL1=1,NL0
      IL1T3 = IL1*3
      IP1 = IPL(IL1T3-2)
      X1 = XD(IP1)
      Y1 = YD(IP1)
      IP2 = IPL(IL1T3-1)
      X2 = XD(IP2)
      Y2 = YD(IP2)
      IF (SPDT(X2,Y2,X1,Y1,X0,Y0).LT.0.0) GO TO 150
      IF (SPDT(X1,Y1,X2,Y2,X0,Y0).LT.0.0) GO TO 140
      IF (SIDE(X1,Y1,X2,Y2,X0,Y0).GT.0.0) GO TO 150
      IL2 = IL1
      GO TO 160
  140 IL2 = MOD(IL1,NL0) + 1
      IP3 = IPL(3*IL2-1)
      X3 = XD(IP3)
      Y3 = YD(IP3)
      IF (SPDT(X3,Y3,X2,Y2,X0,Y0).LE.0.0) GO TO 160
  150 CONTINUE
      IT0 = 1
      GO TO 170
  160 IT0 = IL1*NTL + IL2
C NORMAL EXIT
  170 ITI = IT0
      ITIPV = IT0
      RETURN
      END

      SUBROUTINE  IDPDRV(NDP,XD,YD,ZD,NCP,IPC,PD)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE IDPDRV-ESTIMATES PARTIAL DERIVATIVES OF THE FIRST AND     C
C                    SECOND ORDER AT THE DATA POINTS.                   C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE CALLED BY IDBVIP, IDSFFT                                  C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  INPUT PARAMETERS:                                                    C
C     NDP = NUMBER OF DATA POINTS,                                      C
C     XD,YD,ZD = ARRAYS OF DIMENSION NDP CONTAINING THE X,              C
C           Y, AND Z COORDINATES OF THE DATA POINTS,                    C
C     NCP = NUMBER OF ADDITIONAL DATA POINTS USED FOR ESTI-             C
C           MATING PARTIAL DERIVATIVES AT EACH DATA POINT,              C
C     IPC = INTEGER ARRAY OF DIMENSION NCP*NDP CONTAINING               C
C           THE POINT NUMBERS OF NCP DATA POINTS CLOSEST TO             C
C           EACH OF THE NDP DATA POINTS.                                C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  OUTPUT PARAMETER:                                                    C
C     PD  = ARRAY OF DIMENSION 5*NDP, WHERE THE ESTIMATED               C
C           ZX, ZY, ZXX, ZXY, AND ZYY VALUES AT THE DATA                C
C           POINTS ARE TO BE STORED.                                    C
C                                                                       C
C***********************************************************************C
C                                                                       C
```

```
C DECLARATION STATEMENTS
      DIMENSION   XD(NDP),YD(NDP),ZD(NDP),IPC(NCP*NDP),PD(5*NDP)
      REAL        NMX,NMY,NMZ,NMXX,NMXY,NMYX,NMYY
C PRELIMINARY PROCESSING
   10 NDP0=NDP
      NCP0=NCP
      NCPM1=NCP0-1
C ESTIMATION OF ZX AND ZY
  20  DO 24  IP0=1,NDP0
        X0=XD(IP0)
        Y0=YD(IP0)
        Z0=ZD(IP0)
        NMX=0.0
        NMY=0.0
        NMZ=0.0
        JIPC0=NCP0*(IP0-1)
        DO 23  IC1=1,NCPM1
          JIPC=JIPC0+IC1
          IPI=IPC(JIPC)
          DX1=XD(IPI)-X0
          DY1=YD(IPI)-Y0
          DZ1=ZD(IPI)-Z0
          IC2MN=IC1+1
          DO 22  IC2=IC2MN,NCP0
            JIPC=JIPC0+IC2
            IPI=IPC(JIPC)
            DX2=XD(IPI)-X0
            DY2=YD(IPI)-Y0
            DNMZ=DX1*DY2-DY1*DX2
            IF(DNMZ.EQ.0.0)     GO TO 22
            DZ2=ZD(IPI)-Z0
            DNMX=DY1*DZ2-DZ1*DY2
            DNMY=DZ1*DX2-DX1*DZ2
            IF(DNMZ.GE.0.0)     GO TO 21
            DNMX=-DNMX
            DNMY=-DNMY
            DNMZ=-DNMZ
   21       NMX=NMX+DNMX
            NMY=NMY+DNMY
            NMZ=NMZ+DNMZ
   22     CONTINUE
   23   CONTINUE
        JPD0=5*IP0
        PD(JPD0-4)=-NMX/NMZ
        PD(JPD0-3)=-NMY/NMZ
   24 CONTINUE
C ESTIMATION OF ZXX, ZXY, AND ZYY
   30 DO 34  IP0=1,NDP0
        JPD0=JPD0+5
        X0=XD(IP0)
        JPD0=5*IP0
        Y0=YD(IP0)
        ZX0=PD(JPD0-4)
        ZY0=PD(JPD0-3)
        NMXX=0.0
        NMXY=0.0
        NMYX=0.0
        NMYY=0.0
        NMZ =0.0
        JIPC0=NCP0*(IP0-1)
        DO 33  IC1=1,NCPM1
          JIPC=JIPC0+IC1
          IPI=IPC(JIPC)
          DX1=XD(IPI)-X0
          DY1=YD(IPI)-Y0
          JPD=5*IPI
```

```
          DZX1=PD(JPD-4)-ZX0
          DZY1=PD(JPD-3)-ZY0
          IC2MN=IC1+1
          DO 32  IC2=IC2MN,NCP0
            JIPC=JIPC0+IC2
            IPI=IPC(JIPC)
            DX2=XD(IPI)-X0
            DY2=YD(IPI)-Y0
            DNMZ =DX1*DY2 -DY1*DX2
            IF(DNMZ.EQ.0.0)     GO TO 32
            JPD=5*IPI
            DZX2=PD(JPD-4)-ZX0
            DZY2=PD(JPD-3)-ZY0
            DNMXX=DY1*DZX2-DZX1*DY2
            DNMXY=DZX1*DX2-DX1*DZX2
            DNMYX=DY1*DZY2-DZY1*DY2
            DNMYY=DZY1*DX2-DX1*DZY2
            IF(DNMZ.GE.0.0)     GO TO 31
            DNMXX=-DNMXX
            DNMXY=-DNMXY
            DNMYX=-DNMYX
            DNMYY=-DNMYY
            DNMZ =-DNMZ
   31       NMXX=NMXX+DNMXX
            NMXY=NMXY+DNMXY
            NMYX=NMYX+DNMYX
            NMYY=NMYY+DNMYY
            NMZ =NMZ +DNMZ
   32     CONTINUE
   33   CONTINUE
        PD(JPD0-2)=-NMXX/NMZ
        PD(JPD0-1)=-(NMXY+NMYX)/(2.0*NMZ)
        PD(JPD0)   =-NMYY/NMZ
   34 CONTINUE
      RETURN
      END

      SUBROUTINE  IDPTIP(XD,YD,ZD,NDP,NT,IPT,NL,IPL,PDD,ITI,XII,
     1                   YII,ZII)
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE IDPTIP-PERFORMS PUNCTUAL INTERPOLATION OR EXTRAPOLATION, C
C                    I.E., DETERMINES THE Z VALUE AT A POINT.          C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE CALLED BY IDBVIP, IDSFFT                                 C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  INPUT PARAMETERS:                                                   C
C     XD,YD,ZD = ARRAYS OF DIMENSION NDP CONTAINING THE X,             C
C            Y, AND Z COORDINATES OF THE DATA POINTS, WHERE            C
C            NDP IS THE NUMBER OF THE DATA POINTS,                     C
C     NT  = NUMBER OF TRIANGLES,                                       C
C     IPT = INTEGER ARRAY OF DIMENSION 3*NT CONTAINING THE             C
C            POINT NUMBERS OF THE VERTEXES OF THE TRIANGLES,           C
C     NL  = NUMBER OF BORDER LINE SEGMENTS,                            C
C     IPL = INTEGER ARRAY OF DIMENSION 3*NL CONTAINING THE             C
C            POINT NUMBERS OF THE END POINTS OF THE BORDER             C
C            LINE SEGMENTS AND THEIR RESPECTIVE TRIANGLE               C
C            NUMBERS,                                                  C
C     PDD = ARRAY OF DIMENSION 5*NDP CONTAINING THE PARTIAL            C
C            DERIVATIVES AT THE DATA POINTS,                           C
```

```
C     ITI = TRIANGLE NUMBER OF THE TRIANGLE IN WHICH LIES             C
C           THE POINT FOR WHICH INTERPOLATION IS TO BE                C
C           PERFORMED,                                                C
C     XII,YII = X AND Y COORDINATES OF THE POINT FOR WHICH            C
C           INTERPOLATION IS TO BE PERFORMED.                         C
C                                                                     C
C**********************************************************************C
C                                                                     C
C  OUTPUT PARAMETER:                                                  C
C     ZII = INTERPOLATED Z VALUE.                                     C
C                                                                     C
C**********************************************************************C
C                                                                     C
C DECLARATION STATEMENTS
      DIMENSION   XD(NDP),YD(NDP),ZD(NDP),IPT(3*NT),IPL(3*NL),
     1            PDD(5*NDP)
      COMMON/IDPI/ITPV,X0,Y0,AP,BP,CP,DP,
     1            P00,P10,P20,P30,P40,P50,P01,P11,P21,P31,P41,
     2            P02,P12,P22,P32,P03,P13,P23,P04,P14,P05
      DIMENSION   X(3),Y(3),Z(3),PD(15),
     1            ZU(3),ZV(3),ZUU(3),ZUV(3),ZVV(3)
      REAL        LU,LV
      EQUIVALENCE (P5,P50)
C PRELIMINARY PROCESSING
   10 IT0=ITI
      NTL=NT+NL
      IF(IT0.LE.NTL)      GO TO 20
      IL1=IT0/NTL
      IL2=IT0-IL1*NTL
      IF(IL1.EQ.IL2)      GO TO 40
      GO TO 60
C CALCULATION OF ZII BY INTERPOLATION.
C CHECKS IF THE NECESSARY COEFFICIENTS HAVE BEEN CALCULATED.
   20 IF(IT0.EQ.ITPV)     GO TO 30
C LOADS COORDINATE AND PARTIAL DERIVATIVE VALUES AT THE
C VERTEXES.
   21 JIPT=3*(IT0-1)
      JPD=0
      DO 23  I=1,3
        JIPT=JIPT+1
        IDP=IPT(JIPT)
        X(I)=XD(IDP)
        Y(I)=YD(IDP)
        Z(I)=ZD(IDP)
        JPDD=5*(IDP-1)
        DO 22  KPD=1,5
          JPD=JPD+1
          JPDD=JPDD+1
          PD(JPD)=PDD(JPDD)
   22   CONTINUE
   23 CONTINUE
C DETERMINES THE COEFFICIENTS FOR THE COORDINATE SYSTEM
C TRANSFORMATION FROM THE X-Y SYSTEM TO THE U-V SYSTEM
C AND VICE VERSA.
   24 X0=X(1)
      Y0=Y(1)
      A=X(2)-X0
      B=X(3)-X0
      C=Y(2)-Y0
      D=Y(3)-Y0
      AD=A*D
      BC=B*C
      DLT=AD-BC
      AP= D/DLT
      BP=-B/DLT
      CP=-C/DLT
```

```
      DP= A/DLT
C CONVERTS THE PARTIAL DERIVATIVES AT THE VERTEXES OF THE
C TRIANGLE FOR THE U-V COORDINATE SYSTEM.
   25 AA=A*A
      ACT2=2.0*A*C
      CC=C*C
      AB=A*B
      ADBC=AD+BC
      CD=C*D
      BB=B*B
      BDT2=2.0*B*D
      DD=D*D
      DO 26  I=1,3
        JPD=5*I
        ZU(I)=A*PD(JPD-4)+C*PD(JPD-3)
        ZV(I)=B*PD(JPD-4)+D*PD(JPD-3)
        ZUU(I)=AA*PD(JPD-2)+ACT2*PD(JPD-1)+CC*PD(JPD)
        ZUV(I)=AB*PD(JPD-2)+ADBC*PD(JPD-1)+CD*PD(JPD)
        ZVV(I)=BB*PD(JPD-2)+BDT2*PD(JPD-1)+DD*PD(JPD)
   26 CONTINUE
C CALCULATES THE COEFFICIENTS OF THE POLYNOMIAL.
   27 P00=Z(1)
      P10=ZU(1)
      P01=ZV(1)
      P20=0.5*ZUU(1)
      P11=ZUV(1)
      P02=0.5*ZVV(1)
      H1=Z(2)-P00-P10-P20
      H2=ZU(2)-P10-ZUU(1)
      H3=ZUU(2)-ZUU(1)
      P30= 10.0*H1-4.0*H2+0.5*H3
      P40=-15.0*H1+7.0*H2    -H3
      P50=  6.0*H1-3.0*H2+0.5*H3
      H1=Z(3)-P00-P01-P02
      H2=ZV(3)-P01-ZVV(1)
      H3=ZVV(3)-ZVV(1)
      P03= 10.0*H1-4.0*H2+0.5*H3
      P04=-15.0*H1+7.0*H2    -H3
      P05=  6.0*H1-3.0*H2+0.5*H3
      LU=SQRT(AA+CC)
      LV=SQRT(BB+DD)
      THXU=ATAN2(C,A)
      THUV=ATAN2(D,B)-THXU
      CSUV=COS(THUV)
      P41=5.0*LV*CSUV/LU*P50
      P14=5.0*LU*CSUV/LV*P05
      H1=ZV(2)-P01-P11-P41
      H2=ZUV(2)-P11-4.0*P41
      P21= 3.0*H1-H2
      P31=-2.0*H1+H2
      H1=ZU(3)-P10-P11-P14
      H2=ZUV(3)-P11-4.0*P14
      P12= 3.0*H1-H2
      P13=-2.0*H1+H2
      THUS=ATAN2(D-C,B-A)-THXU
      THSV=THUV-THUS
      AA= SIN(THSV)/LU
      BB=-COS(THSV)/LU
      CC= SIN(THUS)/LV
      DD= COS(THUS)/LV
      AC=AA*CC
      AD=AA*DD
      BC=BB*CC
      G1=AA*AC*(3.0*BC+2.0*AD)
      G2=CC*AC*(3.0*AD+2.0*BC)
      H1=-AA*AA*AA*(5.0*AA*BB*P50+(4.0*BC+AD)*P41)
```

```
     1   -CC*CC*CC*(5.0*CC*DD*P05+(4.0*AD+BC)*P14)
      H2=0.5*ZVV(2)-P02-P12
      H3=0.5*ZUU(3)-P20-P21
      P22=(G1*H2+G2*H3-H1)/(G1+G2)
      P32=H2-P22
      P23=H3-P22
      ITPV=IT0
C CONVERTS XII AND YII TO U-V SYSTEM.
   30 DX=XII-X0
      DY=YII-Y0
      U=AP*DX+BP*DY
      V=CP*DX+DP*DY
C EVALUATES THE POLYNOMIAL.
   31 P0=P00+V*(P01+V*(P02+V*(P03+V*(P04+V*P05))))
      P1=P10+V*(P11+V*(P12+V*(P13+V*P14)))
      P2=P20+V*(P21+V*(P22+V*P23))
      P3=P30+V*(P31+V*P32)
      P4=P40+V*P41
      ZII=P0+U*(P1+U*(P2+U*(P3+U*(P4+U*P5))))
      RETURN
C CALCULATION OF ZII BY EXTRAPOLATION IN THE RECTANGLE.
C CHECKS IF THE NECESSARY COEFFICIENTS HAVE BEEN CALCULATED.
   40 IF(IT0.EQ.ITPV)      GO TO 50
C LOADS COORDINATE AND PARTIAL DERIVATIVE VALUES AT THE END
C POINTS OF THE BORDER LINE SEGMENT.
   41 JIPL=3*(IL1-1)
      JPD=0
      DO 43  I=1,2
        JIPL=JIPL+1
        IDP=IPL(JIPL)
        X(I)=XD(IDP)
        Y(I)=YD(IDP)
        Z(I)=ZD(IDP)
        JPDD=5*(IDP-1)
        DO 42  KPD=1,5
          JPD=JPD+1
          JPDD=JPDD+1
          PD(JPD)=PDD(JPDD)
   42   CONTINUE
   43 CONTINUE
C DETERMINES THE COEFFICIENTS FOR THE COORDINATE SYSTEM
C TRANSFORMATION FROM THE X-Y SYSTEM TO THE U-V SYSTEM
C AND VICE VERSA.
   44 X0=X(1)
      Y0=Y(1)
      A=Y(2)-Y(1)
      B=X(2)-X(1)
      C=-B
      D=A
      AD=A*D
      BC=B*C
      DLT=AD-BC
      AP= D/DLT
      BP=-B/DLT
      CP=-BP
      DP= AP
C CONVERTS THE PARTIAL DERIVATIVES AT THE END POINTS OF THE
C BORDER LINE SEGMENT FOR THE U-V COORDINATE SYSTEM.
   45 AA=A*A
      ACT2=2.0*A*C
      CC=C*C
      AB=A*B
      ADBC=AD+BC
      CD=C*D
      BB=B*B
      BDT2=2.0*B*D
```

```
      DD=D*D
      DO 46  I=1,2
        JPD=5*I
        ZU(I)=A*PD(JPD-4)+C*PD(JPD-3)
        ZV(I)=B*PD(JPD-4)+D*PD(JPD-3)
        ZUU(I)=AA*PD(JPD-2)+ACT2*PD(JPD-1)+CC*PD(JPD)
        ZUV(I)=AB*PD(JPD-2)+ADBC*PD(JPD-1)+CD*PD(JPD)
        ZVV(I)=BB*PD(JPD-2)+BDT2*PD(JPD-1)+DD*PD(JPD)
   46 CONTINUE
C CALCULATES THE COEFFICIENTS OF THE POLYNOMIAL.
   47 P00=Z(1)
      P10=ZU(1)
      P01=ZV(1)
      P20=0.5*ZUU(1)
      P11=ZUV(1)
      P02=0.5*ZVV(1)
      H1=Z(2)-P00-P01-P02
      H2=ZV(2)-P01-ZVV(1)
      H3=ZVV(2)-ZVV(1)
      P03= 10.0*H1-4.0*H2+0.5*H3
      P04=-15.0*H1+7.0*H2    -H3
      P05=  6.0*H1-3.0*H2+0.5*H3
      H1=ZU(2)-P10-P11
      H2=ZUV(2)-P11
      P12= 3.0*H1-H2
      P13=-2.0*H1+H2
      P21=0.0
      P23=-ZUU(2)+ZUU(1)
      P22=-1.5*P23
      ITPV=IT0
C CONVERTS XII AND YII TO U-V SYSTEM.
   50 DX=XII-X0
      DY=YII-Y0
      U=AP*DX+BP*DY
      V=CP*DX+DP*DY
C EVALUATES THE POLYNOMIAL.
   51 P0=P00+V*(P01+V*(P02+V*(P03+V*(P04+V*P05))))
      P1=P10+V*(P11+V*(P12+V*P13))
      P2=P20+V*(P21+V*(P22+V*P23))
      ZII=P0+U*(P1+U*P2)
      RETURN
C CALCULATION OF ZII BY EXTRAPOLATION IN THE TRIANGLE.
C CHECKS IF THE NECESSARY COEFFICIENTS HAVE BEEN CALCULATED.
   60 IF(IT0.EQ.ITPV)      GO TO 70
C LOADS COORDINATE AND PARTIAL DERIVATIVE VALUES AT THE VERTEX
C OF THE TRIANGLE.
   61 JIPL=3*IL2-2
      IDP=IPL(JIPL)
      X0=XD(IDP)
      Y0=YD(IDP)
      Z0=ZD(IDP)
      JPDD=5*(IDP-1)
      DO 62  KPD=1,5
        JPDD=JPDD+1
        PD(KPD)=PDD(JPDD)
   62 CONTINUE
C CALCULATES THE COEFFICIENTS OF THE POLYNOMIAL.
   67 P00=Z0
      P10=PD(1)
      P01=PD(2)
      P20=0.5*PD(3)
      P11=PD(4)
      P02=0.5*PD(5)
      ITPV=IT0
C CONVERTS XII AND YII TO U-V SYSTEM.
   70 U=XII-X0
```

```
      V=YII-Y0
C EVALUATES THE POLYNOMIAL.
   71 P0=P00+V*(P01+V*P02)
      P1=P10+V*P11
      ZII=P0+U*(P1+U*P20)
      RETURN
      END

      SUBROUTINE  IDSFFT(MD,NCP,NDP,XD,YD,ZD,NXI,NYI,XI,YI,ZI,
     1                   IWK,IWKSDM,WK,IERR)
C                                                                    C
C*********************************************************************C
C                                                                    C
C  SUBROUTINE IDSFFT-PERFORMS SMOOTH SURFACE FITTING WHEN THE PRO-   C
C                    JECTIONS OF THE DATA POINTS IN THE X-Y PLANE ARE C
C                    IRREGULARLY DISTRIBUTED IN THE PLANE            C
C                                                                    C
C*********************************************************************C
C                                                                    C
C  SUBROUTINE CALLED BY EXPERT                                       C
C                                                                    C
C*********************************************************************C
C                                                                    C
C  SUBROUTINES CALLED:  IDCLDP, IDGRID, IDPDRV, IDPTIP, IDTANG        C
C                                                                    C
C*********************************************************************C
C                                                                    C
C  INPUT PARAMETERS:                                                 C
C     MD  = MODE OF COMPUTATION (MUST BE 1, 2, OR 3),                C
C         = 1 FOR NEW NCP AND/OR NEW XD-YD,                          C
C         = 2 FOR OLD NCP, OLD XD-YD, NEW XI-YI,                     C
C         = 3 FOR OLD NCP, OLD XD-YD, OLD XI-YI,                     C
C     NCP = NUMBER OF ADDITIONAL DATA POINTS USED FOR ESTI-          C
C           MATING PARTIAL DERIVATIVES AT EACH DATA POINT            C
C           (MUST BE 2 OR GREATER, BUT SMALLER THAN NDP),            C
C     NDP = NUMBER OF DATA POINTS (MUST BE 4 OR GREATER),            C
C     XD  = ARRAY OF DIMENSION NDP CONTAINING THE X                  C
C           COORDINATES OF THE DATA POINTS,                          C
C     YD  = ARRAY OF DIMENSION NDP CONTAINING THE Y                  C
C           COORDINATES OF THE DATA POINTS,                          C
C     ZD  = ARRAY OF DIMENSION NDP CONTAINING THE Z                  C
C           COORDINATES OF THE DATA POINTS,                          C
C     NXI = NUMBER OF OUTPUT GRID POINTS IN THE X COORDINATE         C
C           (MUST BE 1 OR GREATER),                                  C
C     NYI = NUMBER OF OUTPUT GRID POINTS IN THE Y COORDINATE         C
C           (MUST BE 1 OR GREATER),                                  C
C     XI  = ARRAY OF DIMENSION NXI CONTAINING THE X                  C
C           COORDINATES OF THE OUTPUT GRID POINTS,                   C
C     YI  = ARRAY OF DIMENSION NYI CONTAINING THE Y                  C
C           COORDINATES OF THE OUTPUT GRID POINTS.                   C
C     IWKSDM = DIMENSION OF ARRAY IWK                                C
C                                                                    C
C*********************************************************************C
C                                                                    C
C  OUTPUT PARAMETERS:                                                C
C     ZI  = DOUBLY-DIMENSIONED ARRAY OF DIMENSION (NXI,NYI),         C
C           WHERE THE INTERPOLATED Z VALUES AT THE OUTPUT            C
C           GRID POINTS ARE TO BE STORED.                            C
C     IERR= INDICATES ERROR CONDITION.                               C
C           0=NO ERROR                                               C
C           1=ERROR                                                  C
C                                                                    C
C*********************************************************************C
C                                                                    C
C  OTHER PARAMETERS:                                                 C
```

```
C     IWK = INTEGER ARRAY OF DIMENSION                                    C
C               MAX0(31,27+NCP)*NDP+NXI*NYI                               C
C             USED INTERNALLY AS A WORK AREA,                             C
C     WK  = ARRAY OF DIMENSION 5*NDP USED INTERNALLY AS A                 C
C             WORK AREA.                                                  C
C                                                                         C
C*************************************************************************C
C                                                                         C
C   NOTE:                                                                 C
C     THE VERY FIRST CALL TO THIS SUBROUTINE AND THE CALL WITH A NEW      C
C     NCP VALUE, A NEW NDP VALUE, AND/OR NEW CONTENTS OF THE XD AND       C
C     YD ARRAYS MUST BE MADE WITH MD=1.  THE CALL WITH MD=2 MUST BE       C
C     PRECEDED BY ANOTHER CALL WITH THE SAME NCP AND NDP VALUES AND       C
C     WITH THE SAME CONTENTS OF THE XD AND YD ARRAYS.  THE CALL WITH      C
C     MD=3 MUST BE PRECEDED BY ANOTHER CALL WITH THE SAME NCP, NDP,       C
C     NXI, AND NYI VALUES AND WITH THE SAME CONTENTS OF THE XD, YD,       C
C     XI, AND YI ARRAYS.  BETWEEN THE CALL WITH MD=2 OR MD=3 AND ITS      C
C     PRECEDING CALL, THE IWK AND WK ARRAYS MUST NOT BE DISTURBED.        C
C     USE OF A VALUE BETWEEN 3 AND 5 (INCLUSIVE) FOR NCP IS RECOM-        C
C     MENDED UNLESS THERE ARE EVIDENCES THAT DICTATE OTHERWISE.           C
C                                                                         C
C*************************************************************************C
C                                                                         C
C DECLARATION STATEMENTS
      DIMENSION   XD(NDP),YD(NDP),ZD(NDP),XI(NXI),YI(NYI),
     1            ZI(NXI*NYI),IWK(IWKSDM),WK(5*NDP)
      COMMON/IDPI/ITPV,DMMY(27)
      COMMON/UNIT/ILUN,LUN,IDSK
C SETTING OF SOME INPUT PARAMETERS TO LOCAL VARIABLES.
C (FOR MD=1,2,3)
   10 MD0=MD
      NCP0=NCP
      NDP0=NDP
      NXI0=NXI
      NYI0=NYI
C ERROR CHECK.  (FOR MD=1,2,3)
   20 IF(MD0.LT.1.OR.MD0.GT.3)             GO TO 90
      IF(NCP0.LT.2.OR.NCP0.GE.NDP0)        GO TO 90
      IF(NDP0.LT.4)                        GO TO 90
      IF(NXI0.LT.1.OR.NYI0.LT.1)           GO TO 90
      IF(MD0.GE.2)          GO TO 21
      IWK(1)=NCP0
      IWK(2)=NDP0
      GO TO 22
   21 NCPPV=IWK(1)
      NDPPV=IWK(2)
      IF(NCP0.NE.NCPPV)   GO TO 90
      IF(NDP0.NE.NDPPV)   GO TO 90
   22 IF(MD0.GE.3)        GO TO 23
      IWK(3)=NXI0
      IWK(4)=NYI0
      GO TO 30
   23 NXIPV=IWK(3)
      NYIPV=IWK(4)
      IF(NXI0.NE.NXIPV)   GO TO 90
      IF(NYI0.NE.NYIPV)   GO TO 90
C ALLOCATION OF STORAGE AREAS IN THE IWK ARRAY.  (FOR MD=1,2,3)
   30 JWIPT=16
      JWIWL=6*NDP0+1
      JWNGP0=JWIWL-1
      JWIPL=24*NDP0+1
      JWIWP=30*NDP0+1
      JWIPC=27*NDP0+1
      JWIGP0=MAX0(31,27+NCP0)*NDP0
C TRIANGULATES THE X-Y PLANE.  (FOR MD=1)
   40 IF(MD0.GT.1)   GO TO 41
```

```
      CALL IDTANG(NDP0,XD,YD,NT,IWK(JWIPT),NL,IWK(JWIPL),
     1            IWK(JWIWL),IWK(JWIWP),WK,IERR)
      IWK(5)=NT
      IWK(6)=NL
      IF(NT.EQ.0)    RETURN
      GO TO 50
   41 NT=IWK(5)
      NL=IWK(6)
C DETERMINES NCP POINTS CLOSEST TO EACH DATA POINT.  (FOR MD=1)
   50 IF(MD0.GT.1)   GO TO 60
      CALL IDCLDP(NDP0,XD,YD,NCP0,IWK(JWIPC),IERR)
      IF(IWK(JWIPC).EQ.0)      RETURN
C SORTS OUTPUT GRID POINTS IN ASCENDING ORDER OF THE TRIANGLE
C NUMBER AND THE BORDER LINE SEGMENT NUMBER.  (FOR MD=1,2)
   60 IF(MD0.EQ.3)   GO TO 70
      CALL IDGRID(XD,YD,NDP,NT,IWK(JWIPT),NL,IWK(JWIPL),NXI0,NYI0,
     1            XI,YI,IWK(JWNGP0+1),IWK(JWIGP0+1))
C ESTIMATES PARTIAL DERIVATIVES AT ALL DATA POINTS.
C (FOR MD=1,2,3)
   70 CALL IDPDRV(NDP0,XD,YD,ZD,NCP0,IWK(JWIPC),WK)
C INTERPOLATES THE ZI VALUES.  (FOR MD=1,2,3)
   80 ITPV=0
      JIG0MX=0
      JIG1MN=NXI0*NYI0+1
      NNGP=NT+2*NL
      DO 89  JNGP=1,NNGP
        ITI=JNGP
        IF(JNGP.LE.NT)    GO TO 81
        IL1=(JNGP-NT+1)/2
        IL2=(JNGP-NT+2)/2
        IF(IL2.GT.NL)     IL2=1
        ITI=IL1*(NT+NL)+IL2
   81   JWNGP=JWNGP0+JNGP
        NGP0=IWK(JWNGP)
        IF(NGP0.EQ.0)     GO TO 86
        JIG0MN=JIG0MX+1
        JIG0MX=JIG0MX+NGP0
        DO 82  JIGP=JIG0MN,JIG0MX
          JWIGP=JWIGP0+JIGP
          IZI=IWK(JWIGP)
          IYI=(IZI-1)/NXI0+1
          IXI=IZI-NXI0*(IYI-1)
          CALL IDPTIP(XD,YD,ZD,NDP,NT,IWK(JWIPT),NL,IWK(JWIPL),WK,
     1                ITI,XI(IXI),YI(IYI),ZI(IZI))
   82   CONTINUE
   86   JWNGP=JWNGP0+2*NNGP+1-JNGP
        NGP1=IWK(JWNGP)
        IF(NGP1.EQ.0)     GO TO 89
        JIG1MX=JIG1MN-1
        JIG1MN=JIG1MN-NGP1
        DO 87  JIGP=JIG1MN,JIG1MX
          JWIGP=JWIGP0+JIGP
          IZI=IWK(JWIGP)
          IYI=(IZI-1)/NXI0+1
          IXI=IZI-NXI0*(IYI-1)
          CALL IDPTIP(XD,YD,ZD,NDP,NT,IWK(JWIPT),NL,IWK(JWIPL),WK,
     1                ITI,XI(IXI),YI(IYI),ZI(IZI))
   87   CONTINUE
   89 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) MD0,NCP0,NDP0,NXI0,NYI0
      IERR=1
      RETURN
C FORMAT STATEMENT FOR ERROR MESSAGE
 2090 FORMAT(1X/41H ***   IMPROPER INPUT PARAMETER VALUE(S)./
```

```
     1    7H   MD =,I4,10X,5HNCP =,I6,10X,5HNDP =,I6,
     2    10X,5HNHXI =,I6,10X,5HNYI =,I6/
     3    35H ERROR DETECTED IN ROUTINE   IDSFFT/)
      END

      SUBROUTINE  IDTANG(NDP,XD,YD,NT,IPT,NL,IPL,IWL,IWP,WK,IERR)
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE IDTANG-PERFORMS TRIANGULATION.  IT DIVIDES THE X-Y
C                    PLANE INTO A NUMBER OF TRIANGLES ACCORDING TO     C
C                    GIVEN DATA POINTS IN THE PLANE, DETERMINES LINE   C
C                    SEGMENTS THAT FORM THE BORDER OF DATA AREA, AND   C
C                    DETERMINES THE TRIANGLE NUMBERS CORRESPONDING TO  C
C                    THE BORDER LINE SEGMENTS. AT COMPLETION, POINT    C
C                    NUMBERS OF THE VERTEXES OF EACH TRIANGLE ARE      C
C                    LISTED COUNTER-CLOCKWISE.  POINT NUMBERS OF THE   C
C                    END POINTS OF EACH BORDER LINE SEGMENT ARE LISTED C
C                    COUNTER-CLOCKWISE, LISTING ORDER OF THE LINE      C
C                    SEGMENTS BEING COUNTER-CLOCKWISE.                 C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE CALLED BY IDBVIP, IDSSFT                                 C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  FUNCTION CALLED: IDXCHG                                             C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  INPUT PARAMETERS:                                                   C
C     NDP = NUMBER OF DATA POINTS,                                     C
C     XD  = ARRAY OF DIMENSION NDP CONTAINING THE                      C
C           X COORDINATES OF THE DATA POINTS,                          C
C     YD  = ARRAY OF DIMENSION NDP CONTAINING THE                      C
C           Y COORDINATES OF THE DATA POINTS.                          C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  OUTPUT PARAMETERS:                                                  C
C     NT  = NUMBER OF TRIANGLES,                                       C
C     IPT = INTEGER ARRAY OF DIMENSION 6*NDP-15, WHERE THE             C
C           POINT NUMBERS OF THE VERTEXES OF THE (IT)TH                C
C           TRIANGLE ARE TO BE STORED AS THE (3*IT-2)ND,               C
C           (3*IT-1)ST, AND (3*IT)TH ELEMENTS,                         C
C           IT=1,2,...,NT,                                             C
C     NL  = NUMBER OF BORDER LINE SEGMENTS,                            C
C     IPL = INTEGER ARRAY OF DIMENSION 6*NDP, WHERE THE                C
C           POINT NUMBERS OF THE END POINTS OF THE (IL)TH              C
C           BORDER LINE SEGMENT AND ITS RESPECTIVE TRIANGLE            C
C           NUMBER ARE TO BE STORED AS THE (3*IL-2)ND,                 C
C           (3*IL-1)ST, AND (3*IL)TH ELEMENTS,                         C
C           IL=1,2,..., NL.                                            C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  OTHER PARAMETERS:                                                   C
C     IWL = INTEGER ARRAY OF DIMENSION 18*NDP USED                     C
C           INTERNALLY AS A WORK AREA,                                 C
C     IWP = INTEGER ARRAY OF DIMENSION NDP USED                        C
C           INTERNALLY AS A WORK AREA,                                 C
C     WK  = ARRAY OF DIMENSION NDP USED INTERNALLY AS A                C
C           WORK AREA.                                                 C
C                                                                      C
```

```
C***********************************************************************C
C                                                                       C
C DECLARATION STATEMENTS
      DIMENSION    XD(NDP),YD(NDP),IPT(6*NDP-15),IPL(6*NDP),
     1             IWL(18*NDP),IWP(NDP),WK(NDP)
      DIMENSION    ITF(2)
      COMMON/UNIT/ILUN,LUN,IDSK
      DATA  RATIO/1.0E-6/, NREP/100/
C STATEMENT FUNCTIONS
      DSQF(U1,V1,U2,V2)=(U2-U1)**2+(V2-V1)**2
      SIDE(U1,V1,U2,V2,U3,V3)=(V3-V1)*(U2-U1)-(U3-U1)*(V2-V1)
C PRELIMINARY PROCESSING
   10 NDP0=NDP
      NDPM1=NDP0-1
      IF(NDP0.LT.4)        GO TO 90
C DETERMINES THE CLOSEST PAIR OF DATA POINTS AND THEIR MIDPOINT.
   20 DSQMN=DSQF(XD(1),YD(1),XD(2),YD(2))
      IPMN1=1
      IPMN2=2
      DO 22  IP1=1,NDPM1
        X1=XD(IP1)
        Y1=YD(IP1)
        IP1P1=IP1+1
        DO 21  IP2=IP1P1,NDP0
          DSQI=DSQF(X1,Y1,XD(IP2),YD(IP2))
          IF(DSQI.EQ.0.0)      GO TO 91
          IF(DSQI.GE.DSQMN)    GO TO 21
          DSQMN=DSQI
          IPMN1=IP1
          IPMN2=IP2
   21   CONTINUE
   22 CONTINUE
      DSQ12=DSQMN
      XDMP=(XD(IPMN1)+XD(IPMN2))/2.0
      YDMP=(YD(IPMN1)+YD(IPMN2))/2.0
C SORTS THE OTHER (NDP-2) DATA POINTS IN ASCENDING ORDER OF
C DISTANCE FROM THE MIDPOINT AND STORES THE SORTED DATA POINT
C NUMBERS IN THE IWP ARRAY.
   30 JP1=2
      DO 31  IP1=1,NDP0
        IF(IP1.EQ.IPMN1.OR.IP1.EQ.IPMN2)      GO TO 31
        JP1=JP1+1
        IWP(JP1)=IP1
        WK(JP1)=DSQF(XDMP,YDMP,XD(IP1),YD(IP1))
   31 CONTINUE
      DO 33  JP1=3,NDPM1
        DSQMN=WK(JP1)
        JPMN=JP1
        DO 32  JP2=JP1,NDP0
          IF(WK(JP2).GE.DSQMN)      GO TO 32
          DSQMN=WK(JP2)
          JPMN=JP2
   32   CONTINUE
        ITS=IWP(JP1)
        IWP(JP1)=IWP(JPMN)
        IWP(JPMN)=ITS
        WK(JPMN)=WK(JP1)
   33 CONTINUE
C IF NECESSARY, MODIFIES THE ORDERING IN SUCH A WAY THAT THE
C FIRST THREE DATA POINTS ARE NOT COLLINEAR.
   35 AR=DSQ12*RATIO
      X1=XD(IPMN1)
      Y1=YD(IPMN1)
      DX21=XD(IPMN2)-X1
      DY21=YD(IPMN2)-Y1
      DO 36  JP=3,NDP0
```

```
          IP=IWP(JP)
          IF(ABS((YD(IP)-Y1)*DX21-(XD(IP)-X1)*DY21).GT.AR)
     1                 GO TO 37
   36 CONTINUE
      GO TO 92
   37 IF(JP.EQ.3)     GO TO 40
      JPMX=JP
      JP=JPMX+1
      DO 38  JPC=4,JPMX
        JP=JP-1
        IWP(JP)=IWP(JP-1)
   38 CONTINUE
      IWP(3)=IP
C FORMS THE FIRST TRIANGLE.  STORES POINT NUMBERS OF THE VER-
C TEXES OF THE TRIANGLE IN THE IPT ARRAY, AND STORES POINT NUM-
C BERS OF THE BORDER LINE SEGMENTS AND THE TRIANGLE NUMBER IN
C THE IPL ARRAY.
   40 IP1=IPMN1
      IP2=IPMN2
      IP3=IWP(3)
      IF(SIDE(XD(IP1),YD(IP1),XD(IP2),YD(IP2),XD(IP3),YD(IP3))
     1     .GE.0.0)        GO TO 41
      IP1=IPMN2
      IP2=IPMN1
   41 NT0=1
      NTT3=3
      IPT(1)=IP1
      IPT(2)=IP2
      IPT(3)=IP3
      NL0=3
      NLT3=9
      IPL(1)=IP1
      IPL(2)=IP2
      IPL(3)=1
      IPL(4)=IP2
      IPL(5)=IP3
      IPL(6)=1
      IPL(7)=IP3
      IPL(8)=IP1
      IPL(9)=1
C ADDS THE REMAINING (NDP-3) DATA POINTS, ONE BY ONE.
   50 DO 79  JP1=4,NDP0
        IP1=IWP(JP1)
        X1=XD(IP1)
        Y1=YD(IP1)
C - DETERMINES THE VISIBLE BORDER LINE SEGMENTS.
        IP2=IPL(1)
        JPMN=1
        DXMN=XD(IP2)-X1
        DYMN=YD(IP2)-Y1
        DSQMN=DXMN**2+DYMN**2
        ARMN=DSQMN*RATIO
        JPMX=1
        DXMX=DXMN
        DYMX=DYMN
        DSQMX=DSQMN
        ARMX=ARMN
        DO 52  JP2=2,NL0
          IP2=IPL(3*JP2-2)
          DX=XD(IP2)-X1
          DY=YD(IP2)-Y1
          AR=DY*DXMN-DX*DYMN
          IF(AR.GT.ARMN)        GO TO 51
          DSQI=DX**2+DY**2
          IF(AR.GE.(-ARMN).AND.DSQI.GE.DSQMN)       GO TO 51
          JPMN=JP2
```

```
            DXMN=DX
            DYMN=DY
            DSQMN=DSQI
            ARMN=DSQMN*RATIO
   51       AR=DY*DXMX-DX*DYMX
            IF(AR.LT.(-ARMX))    GO TO 52
            DSQI=DX**2+DY**2
            IF(AR.LE.ARMX.AND.DSQI.GE.DSQMX)    GO TO 52
            JPMX=JP2
            DXMX=DX
            DYMX=DY
            DSQMX=DSQI
            ARMX=DSQMX*RATIO
   52     CONTINUE
          IF(JPMX.LT.JPMN)  JPMX=JPMX+NL0
          NSH=JPMN-1
          IF(NSH.LE.0)       GO TO 60
C - SHIFTS (ROTATES) THE IPL ARRAY TO HAVE THE INVISIBLE BORDER
C - LINE SEGMENTS CONTAINED IN THE FIRST PART OF THE IPL ARRAY.
          NSHT3=NSH*3
          DO 53  JP2T3=3,NSHT3,3
            JP3T3=JP2T3+NLT3
            IPL(JP3T3-2)=IPL(JP2T3-2)
            IPL(JP3T3-1)=IPL(JP2T3-1)
            IPL(JP3T3)  =IPL(JP2T3)
   53     CONTINUE
          DO 54  JP2T3=3,NLT3,3
            JP3T3=JP2T3+NSHT3
            IPL(JP2T3-2)=IPL(JP3T3-2)
            IPL(JP2T3-1)=IPL(JP3T3-1)
            IPL(JP2T3)  =IPL(JP3T3)
   54     CONTINUE
          JPMX=JPMX-NSH
C - ADDS TRIANGLES TO THE IPT ARRAY, UPDATES BORDER LINE
C - SEGMENTS IN THE IPL ARRAY, AND SETS FLAGS FOR THE BORDER
C - LINE SEGMENTS TO BE REEXAMINED IN THE IWL ARRAY.
   60     JWL=0
          DO 64  JP2=JPMX,NL0
            JP2T3=JP2*3
            IPL1=IPL(JP2T3-2)
            IPL2=IPL(JP2T3-1)
            IT  =IPL(JP2T3)
C - - ADDS A TRIANGLE TO THE IPT ARRAY.
            NT0=NT0+1
            NTT3=NTT3+3
            IPT(NTT3-2)=IPL2
            IPT(NTT3-1)=IPL1
            IPT(NTT3)  =IP1
C - - UPDATES BORDER LINE SEGMENTS IN THE IPL ARRAY.
            IF(JP2.NE.JPMX)      GO TO 61
            IPL(JP2T3-1)=IP1
            IPL(JP2T3)  =NT0
   61       IF(JP2.NE.NL0)       GO TO 62
            NLN=JPMX+1
            NLNT3=NLN*3
            IPL(NLNT3-2)=IP1
            IPL(NLNT3-1)=IPL(1)
            IPL(NLNT3)  =NT0
C - - DETERMINES THE VERTEX THAT DOES NOT LIE ON THE BORDER
C - - LINE SEGMENTS.
   62       ITT3=IT*3
            IPTI=IPT(ITT3-2)
            IF(IPTI.NE.IPL1.AND.IPTI.NE.IPL2)   GO TO 63
            IPTI=IPT(ITT3-1)
            IF(IPTI.NE.IPL1.AND.IPTI.NE.IPL2)   GO TO 63
            IPTI=IPT(ITT3)
```

```
C - - CHECKS IF THE EXCHANGE IS NECESSARY.
   63     IF(IDXCHG(XD,YD,IP1,IPTI,IPL1,IPL2).EQ.0)      GO TO 64
C - - MODIFIES THE IPT ARRAY WHEN NECESSARY.
          IPT(ITT3-2)=IPTI
          IPT(ITT3-1)=IPL1
          IPT(ITT3)  =IP1
          IPT(NTT3-1)=IPTI
          IF(JP2.EQ.JPMX)       IPL(JP2T3)=IT
          IF(JP2.EQ.NL0.AND.IPL(3).EQ.IT)      IPL(3)=NT0
C - - SETS FLAGS IN THE IWL ARRAY.
          JWL=JWL+4
          IWL(JWL-3)=IPL1
          IWL(JWL-2)=IPTI
          IWL(JWL-1)=IPTI
          IWL(JWL)  =IPL2
   64   CONTINUE
        NL0=NLN
        NLT3=NLNT3
        NLF=JWL/2
        IF(NLF.EQ.0)       GO TO 79
C - IMPROVES TRIANGULATION.
   70   NTT3P3=NTT3+3
        DO 78  IREP=1,NREP
          DO 76  ILF=1,NLF
            ILFT2=ILF*2
            IPL1=IWL(ILFT2-1)
            IPL2=IWL(ILFT2)
C - - LOCATES IN THE IPT ARRAY TWO TRIANGLES ON BOTH SIDES OF
C - - THE FLAGGED LINE SEGMENT.
            NTF=0
            DO 71  ITT3R=3,NTT3,3
              ITT3=NTT3P3-ITT3R
              IPT1=IPT(ITT3-2)
              IPT2=IPT(ITT3-1)
              IPT3=IPT(ITT3)
              IF(IPL1.NE.IPT1.AND.IPL1.NE.IPT2.AND.
     1           IPL1.NE.IPT3)       GO TO 71
              IF(IPL2.NE.IPT1.AND.IPL2.NE.IPT2.AND.
     1           IPL2.NE.IPT3)       GO TO 71
              NTF=NTF+1
              ITF(NTF)=ITT3/3
              IF(NTF.EQ.2)      GO TO 72
   71       CONTINUE
            IF(NTF.LT.2)        GO TO 76
C - - DETERMINES THE VERTEXES OF THE TRIANGLES THAT DO NOT LIE
C - - ON THE LINE SEGMENT.
   72       IT1T3=ITF(1)*3
            IPTI1=IPT(IT1T3-2)
            IF(IPTI1.NE.IPL1.AND.IPTI1.NE.IPL2)     GO TO 73
            IPTI1=IPT(IT1T3-1)
            IF(IPTI1.NE.IPL1.AND.IPTI1.NE.IPL2)     GO TO 73
            IPTI1=IPT(IT1T3)
   73       IT2T3=ITF(2)*3
            IPTI2=IPT(IT2T3-2)
            IF(IPTI2.NE.IPL1.AND.IPTI2.NE.IPL2)     GO TO 74
            IPTI2=IPT(IT2T3-1)
            IF(IPTI2.NE.IPL1.AND.IPTI2.NE.IPL2)     GO TO 74
            IPTI2=IPT(IT2T3)
C - - CHECKS IF THE EXCHANGE IS NECESSARY.
   74       IF(IDXCHG(XD,YD,IPTI1,IPTI2,IPL1,IPL2).EQ.0)
     1         GO TO 76
C - - MODIFIES THE IPT ARRAY WHEN NECESSARY.
            IPT(IT1T3-2)=IPTI1
            IPT(IT1T3-1)=IPTI2
            IPT(IT1T3)  =IPL1
            IPT(IT2T3-2)=IPTI2
```

```
            IPT(IT2T3-1)=IPTI1
            IPT(IT2T3)  =IPL2
C - - SETS NEW FLAGS.
            JWL=JWL+8
            IWL(JWL-7)=IPL1
            IWL(JWL-6)=IPTI1
            IWL(JWL-5)=IPTI1
            IWL(JWL-4)=IPL2
            IWL(JWL-3)=IPL2
            IWL(JWL-2)=IPTI2
            IWL(JWL-1)=IPTI2
            IWL(JWL)  =IPL1
            DO 75  JLT3=3,NLT3,3
              IPLJ1=IPL(JLT3-2)
              IPLJ2=IPL(JLT3-1)
              IF((IPLJ1.EQ.IPL1.AND.IPLJ2.EQ.IPTI2).OR.
     1           (IPLJ2.EQ.IPL1.AND.IPLJ1.EQ.IPTI2))
     2                        IPL(JLT3)=ITF(1)
              IF((IPLJ1.EQ.IPL2.AND.IPLJ2.EQ.IPTI1).OR.
     1           (IPLJ2.EQ.IPL2.AND.IPLJ1.EQ.IPTI1))
     2                        IPL(JLT3)=ITF(2)
   75       CONTINUE
   76     CONTINUE
          NLFC=NLF
          NLF=JWL/2
          IF(NLF.EQ.NLFC)       GO TO 79
C - - RESETS THE IWL ARRAY FOR THE NEXT ROUND.
          JWL=0
          JWL1MN=(NLFC+1)*2
          NLFT2=NLF*2
          DO 77  JWL1=JWL1MN,NLFT2,2
            JWL=JWL+2
            IWL(JWL-1)=IWL(JWL1-1)
            IWL(JWL)  =IWL(JWL1)
   77     CONTINUE
          NLF=JWL/2
   78   CONTINUE
   79 CONTINUE
C REARRANGES THE IPT ARRAY SO THAT THE VERTEXES OF EACH TRIANGLE
C ARE LISTED COUNTER-CLOCKWISE.
   80 DO 81  ITT3=3,NTT3,3
        IP1=IPT(ITT3-2)
        IP2=IPT(ITT3-1)
        IP3=IPT(ITT3)
        IF(SIDE(XD(IP1),YD(IP1),XD(IP2),YD(IP2),XD(IP3),YD(IP3))
     1       .GE.0.0)      GO TO 81
        IPT(ITT3-2)=IP2
        IPT(ITT3-1)=IP1
   81 CONTINUE
      NT=NT0
      NL=NL0
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090)  NDP0
      GO TO 93
   91 WRITE (LUN,2091)  NDP0,IP1,IP2,X1,Y1
      GO TO 93
   92 WRITE (LUN,2092)  NDP0
   93 WRITE (LUN,2093)
      NT=0
      IERR=1
      RETURN
C FORMAT STATEMENTS
 2090 FORMAT(1X/23H ***   NDP LESS THAN 4./8H   NDP =,I5)
 2091 FORMAT(1X/29H ***   IDENTICAL DATA POINTS./
     1   8H   NDP =,I5,5X,5HIP1 =,I5,5X,5HIP2 =,I5,
```

```
    2   5X,4HXD =,E12.4,5X,4HYD =,E12.4)
 2092 FORMAT(1X/33H ***   ALL COLLINEAR DATA POINTS./
    1   8H   NDP =,I5)
 2093 FORMAT(35H ERROR DETECTED IN ROUTINE   IDTANG/)
      END

      FUNCTION  IDXCHG(X,Y,I1,I2,I3,I4)
C                                                                     C
C*********************************************************************C
C                                                                     C
C   FUNCTION IDXCHG-DETERMINES WHETHER OR NOT THE EXCHANGE OF TWO     C
C                   TRIANGLES IS NECESSARY ON THE BASIS OF MAX-MIN-   C
C                   ANGLE CRITERION BY C. L. LAWSON.                  C
C                                                                     C
C*********************************************************************C
C                                                                     C
C   FUNCTION CALLED BY IDTANG                                         C
C                                                                     C
C*********************************************************************C
C                                                                     C
C  INPUT PARAMETERS:                                                  C
C     X,Y = ARRAYS CONTAINING THE COORDINATES OF THE DATA             C
C           POINTS,                                                   C
C     I1,I2,I3,I4 = POINT NUMBERS OF FOUR POINTS P1, P2,              C
C           P3, AND P4 THAT FORM A QUADRILATERAL WITH P3              C
C           AND P4 CONNECTED DIAGONALLY.                              C
C                                                                     e
C*********************************************************************C
C                                                                     C
C  OUTPUT PARAMETERS:                                                 C
C     THIS FUNCTION RETURNS AN INTEGER VALUE 1 (ONE) WHEN AN EX-      C
C     CHANGE IS NECESSARY, AND 0 (ZERO) OTHERWISE.                    C
C                                                                     C
C*********************************************************************C
C                                                                     C
C DECLARATION STATEMENTS
      DIMENSION   X(100),Y(100)
      EQUIVALENCE (C2SQ,C1SQ),(A3SQ,B2SQ),(B3SQ,A1SQ),
     1            (A4SQ,B1SQ),(B4SQ,A2SQ),(C4SQ,C3SQ)
C PRELIMINARY PROCESSING
   10 X1=X(I1)
      Y1=Y(I1)
      X2=X(I2)
      Y2=Y(I2)
      X3=X(I3)
      Y3=Y(I3)
      X4=X(I4)
      Y4=Y(I4)
C CALCULATION
   20 IDX=0
      U3=(Y2-Y3)*(X1-X3)-(X2-X3)*(Y1-Y3)
      U4=(Y1-Y4)*(X2-X4)-(X1-X4)*(Y2-Y4)
      IF(U3*U4.LE.0.0)    GO TO 30
      U1=(Y3-Y1)*(X4-X1)-(X3-X1)*(Y4-Y1)
      U2=(Y4-Y2)*(X3-X2)-(X4-X2)*(Y3-Y2)
      A1SQ=(X1-X3)**2+(Y1-Y3)**2
      B1SQ=(X4-X1)**2+(Y4-Y1)**2
      C1SQ=(X3-X4)**2+(Y3-Y4)**2
      A2SQ=(X2-X4)**2+(Y2-Y4)**2
      B2SQ=(X3-X2)**2+(Y3-Y2)**2
      C3SQ=(X2-X1)**2+(Y2-Y1)**2
      S1SQ=U1*U1/(C1SQ*AMAX1(A1SQ,B1SQ))
      S2SQ=U2*U2/(C2SQ*AMAX1(A2SQ,B2SQ))
      S3SQ=U3*U3/(C3SQ*AMAX1(A3SQ,B3SQ))
      S4SQ=U4*U4/(C4SQ*AMAX1(A4SQ,B4SQ))
```

```
      IF(AMIN1(S1SQ,S2SQ).LT.AMIN1(S3SQ,S4SQ))      IDX=1
   30 IDXCHG=IDX
      RETURN
      END
```

APPENDIX B

EXPERT User's Manual

E X P E R T

EXPERIMENTAL DATA INTERPOLATION PROGRAM

An Interactive Computer Program for the Display of

Experimental and Field Test Data

USER'S MANUAL

Department of Civil Engineering
Brigham Young University
May 1983

EXPERT
User's Manual

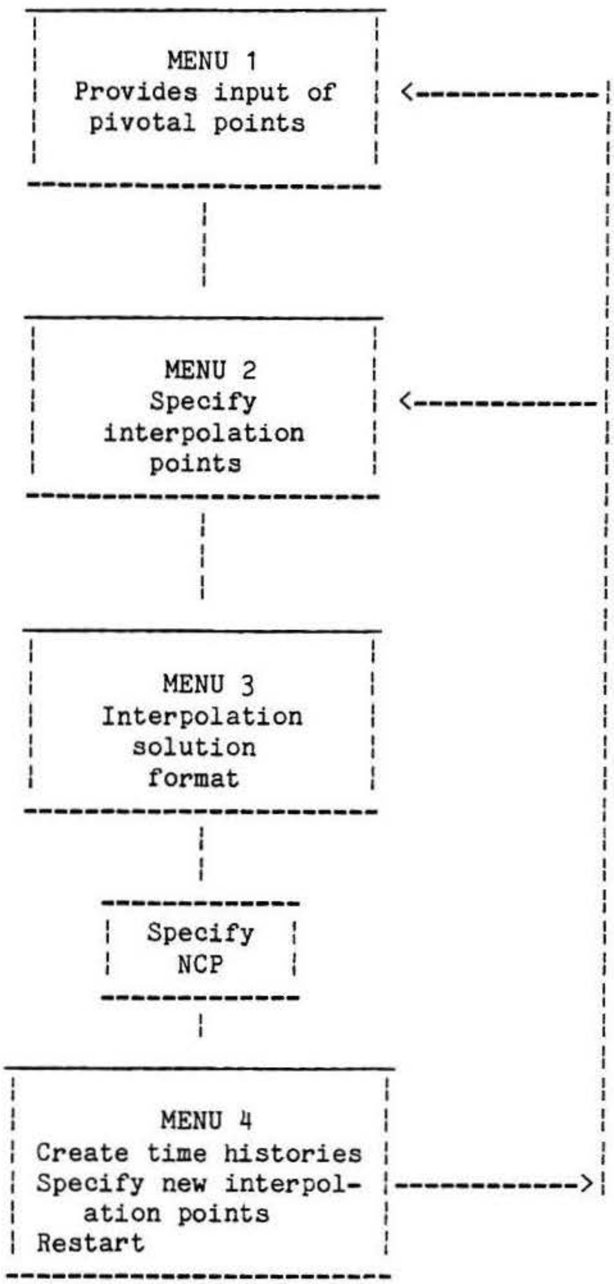
Introduction

EXPERT was developed under a contract from the Air Force Weapons Laboratory to display field test data. Its development and a demonstration of a typical practical application is given by Adams in a Thesis entitled "Development of a Computer-Aided Graphics Capability to Display Field Test Data" [Brigham Young University: August 1983]. Any set of "sparse" data that lies in a two dimensional x-y plane or that can be 'unwrapped' to lie in a plane can be input and displayed. Interpolation is performed over the entire domain or at any specific point in the domain using the input data as the "pivotal" values. In this manner a sufficient data base is created to enable the use of computer graphics software, specifically MOVIE.BYU and DISSPLA. The response function of a specific point over an interval of time can be graphically illustrated in the form of a time history, or a variable over the whole surface at a specific time can be graphically displayed with continuous tone color, contours, or warped surfaces. The program consists of a series of menus, or option lists, providing the user a wide range of choices. When a menu appears, the selection is read as an alphanumeric character. This allows for the specification of either an option number from the options listed or a special option. The special options are:

H	Help.	A series of HELP messages can be accessed describing each option for each menu.
R	Revert.	Reverts control to the previous menu, allowing the user to respecify data.
S	Start.	Causes the program to start from the first menu.
T	Terminate.	Terminates the program.

The program is "user friendly" as most prompts are self-explanatory.

Shown below is a simple flow chart of the program.



A complete explanation of each of the options is given below.

Input of Experimental Data Points
MENU 1

The first option list is as follows:

```
INPUT OF EXPERIMENTAL DATA POINTS:

  OPTIONS:
     1.  INPUT DATA AND CREATE FILE
     2.  READ DATA FILE
     3.  READ MULTIPLE DATA FILES
  SPECIAL OPTIONS:
     4.  READ E.U. TAPE AND CREATE DATA FILES
     5.  CREATE TIME HISTORIES FROM HONDO DATA FILE
     6.  PLOT TIME HISTORIES
     7.  HELP

SELECTION:
```

Option 1 allows the user to input the experimental data. The user is prompted for the number of points, the x and y coordinate, and z value of each point. The input data is then relisted, allowing verification of the data and corrections, if needed. The data can be assigned a file name and stored in a disk file if desired, to be read in at a later time using Option 2.

Option 2 allows the user to specify an existing disk file as the source of the input data. The data is listed, allowing verification and corrections if necessary. The file is read with the following FORTRAN steps:

```
READ(IDSK,*) NDP
READ(IDSK,*) (XD(N),YD(N),ZD(N),N=1,NDP)
```

where NDP is the number of data points, XD is an array containing the x coordinates, YD is an array containing the y coordinates, and ZD is an array containing the data point values. Any file thus formatted could be accessed.

Option 3 allows the user to specify a group of existing disk files as the source of the input data. The files would be read and interpolation performed one at a time as specified by subsequent menus. This is convenient when there is a series of files based on the same configuration but with the z values some time step apart. The file names must have the following format: one letter (not M or R) followed by a one- to five- digit number, with no leading zeros. The number should be the timestep corresponding to the file, in milliseconds. For example, the file corresponding to time t=0 would be given the name, say, C0.DAT. If the time step between sets of data is 2 milliseconds, succeeding files would be given the names C2, C4, C6, etc. The user then recieves the following series of prompts:

NUMBER OF FILES:	Input the total number of files to be read.
FILENAME OF FIRST FILE:	Input the name of the first file (e.g., C0). The CRAY version prompts separately for the letter and the number.
FILENAME INCREMENT:	Input the increment between file names (e.g., 2).

DO YOU WANT TIME HISTORIES?

This allows the user to create time histories for any set of points desired. The user would be prompted for the number of history files to be created, the coordinates of the points and the corresponding file names to be given the history files as follows:

```
NUMBER OF POINTS:
   INPUT COORDINATES (X,Y):
   ASSIGN TIME HISTORY FILENAME:
```

Time histories are created for the coordinates specified with a time range corresponding to the times defined by the multiple files.

To adequately identify the time history plots the following prompts are given:

```
TYPE:
   1. DISPLACEMENT
   2. VELOCITY
   3. ACCELERATION
   4. PRESSURE
   5. OTHER
SELECTION:

DIRECTION
   1. VERTICAL
   2. RADIAL
   3. OTHER
SELECTION:
```

These specify character strings which compose the legend on the plots. The type also is used to label the axis. The direction is indicated on the plot by a single character, Z for vertical and R for radial. Other character strings can be specified by the user if desired. The user will then be prompted to see if other interpolations are desired. If NO, the second and third menus will be skipped.

Option 4 allows the user to create properly formatted data files for use with Option 3 from A.F.W.L. Engineering Units tapes. This capability is accessed by using subroutines EUTAPE and RDTAPE as described below.

```
      SUBROUTINE EUTAPE
      DIMENSION XD(NDP),YD(NDP),HIST(NTS,NDP),ITIME(NTS)
      CALL RDTAPE(XD,YD,HIST,ITIME)
      DO 120 I=1,NTS
         ENCODE(10,100,FILNAM) ITIME(I)
100      FORMAT('C',I5)
         OPEN(UNIT=1,FILE=FILNAM,ACCESS='SEQUENTIAL',STATUS='NEW')
         WRITE(1,*) NDP
         DO 110 J=1,NDP
            WRITE(1,*) (XD(J),YD(J),HIST(I,J))
110      CONTINUE
         CLOSE(UNIT=1)
120   CONTINUE
      RETURN
      END
```

Subroutine RDTAPE is a user supplied subroutine that reads the A.F.W.L. Engineering Units tape and creates XD, an array of dimension NDP containing the x coordinates of the gauges or measuring devices, where NDP = Number of Data Points, or gauges; YD, an array of dimension NDP containing the y coordinates of the gauges; HIST(NTS,NDP), a doubly dimensioned array containing all of the time history data, where NTS = Number of time steps; and ITIME, an integer array of dimension NTS containing the time for each time step, in milliseconds. For example, XD(n) is the x coordinate and YD(n) is the y coordinate for some gauge n that recorded the time history data contained in HIST(1,n) to HIST(NTS,n) with time steps corresponding to times ITIME(1) to ITIME(NTS). RDTAPE should also allow the user a way to specify for which time steps the tape is to be read and files created. ITIME is used to assign names to the files created as indicated by the ENCODE statement and FORMAT number 100. Any letter but M and R can be used in place of C, and it is recommended that the subroutine be written to allow the user to define interactively the letter used. Due to the size of the array HIST required it may be more efficient to store the time history data in a temporary random access disk file rather than in HIST.

Option 5 and Option 6 are special options and will be explained later in conjunction with MENU 4.

Development of Interpolation Points
MENU 2

Using as pivotal values the field data input from MENU 1 interpolation is performed at points specified by the user according to the intended use of the data. The coordinates of the mesh that define the field to be interpolated should overlay the pivotal point data defined in MENU 1.

```
DEVELOPMENT OF INTERPOLATION POINTS:
    (POINTS AT WHICH INTERPOLATION WILL BE PERFORMED)

   OPTIONS:
      1.  READ QMESH DATA FILE
      2.  CREATE A REGULAR RECTANGULAR GRID
      3.  SPECIFY RANDOM POINTS TO BE INTERPOLATED
      4.  HELP
SELECTION:
```

Option 1 reads a file created by QMESH, an interactive mesh generation program, and assigns those points as interpolation points. This gives the user wide latitude as to the geometry of the interpolated surface and the coordinates of the points of interpolation. It also provides a convenient, consistant way of specifying the interpolation points.

Option 2 creates a rectangular grid specified by the user in the following way:

```
FORMATION OF GRID SYSTEM:

X-COORDINATE OF THE FIRST VERTICAL GRID LINE:
X-COORDINATE OF THE LAST VERTICAL GRID LINE:
TOTAL NUMBER OF VERTICAL GRID LINES:
Y-COORDINATE OF THE FIRST HORIZONTAL GRID LINE:
Y-COORDINATE OF THE LAST HORIZONTAL GRID LINE:
TOTAL NUMBER OF HORIZONTAL GRID LINES:
```

The intersection of the grid lines are used as the interpolation points.

Option 3 allows the user to select any number of specific point to be interpolated. This gives the user greater control in selecting only the points of greatest interest, especially if there are only a few points. The user is prompted for the number of such points and the x and y coordinates of those points.

Solutions
MENU 3

```
SOLUTION:
   OPTIONS:
       1.  PRINT RESULTS ON TTY
       2.  CREATE DISK FILE OF RESULTS
       3.  CREATE MOVIE.BYU FILES
       4.  1 & 2
       5.  1 & 3
       6.  2 & 3
       7.  1,2, & 3
       8.  CONTINUE
       9.  HELP
SELECTION:
```

Option 1 lists all input parameters and resulting interpolation values on the screen.

Option 2 creates a file of results formatted the same as Option 1. If Option 3 (Multiple Data Files) of MENU 1 was specified the file would be given the name 'R' followed by the one- to five digit number corresponding to the input file (e.g., R0, R2, R4, etc.).

Option 3 creates a file of the results with proper format to use with MOVIE.BYU. Both a geometry and function file are created. If Option 3 (Multiple Data Files) of MENU 1 was specified the file would be given the name 'M' followed by the one- to five digit number corresponding to the input file (e.g., M0, M2, M4, etc.). MOVIE.BYU format files cannot be created when the interpolation points are defined by Option 3 (Random Points) of MENU 2.

Option 4 through Option 7 performs a combination of the above options as indicated. When single files are interpolated, the Solution options can be selected one at a time. When an option has been completed the menu reappears and another option can be selected. Option 8 causes the program to skip to the final MENU after all desired options from

MENU 3 have been performed. If Option 8 is selected upon the first appearance of MENU 3, control passes to the final option list and NO interpolations are performed. When multiple files are interpolated MENU 3 only appears once. After the option selected has been performed MENU 4, the final option list, appears. Since the menu only appears once the option that contains the desired combination of options should be selected.

NCP

After the data has been input and the interpolation points have been specified, the following series of prompts appear:

```
NUMBER OF CLOSEST DATA POINTS (NCP) TO BE USED
 FOR INTERPOLATION OF EACH POINT (OR TYPE HELP):
```

In calculating the coefficients of the polynomial the program uses the z values of some specified number of closest data points. There is no theory to dictate the optimum value to use, and it is somewhat dependent upon the distribution of the data points. The user may specify the value used or may input one of the special options:

```
SPECIAL OPTIONS:

<RETURN> CAUSES THE DEFAULT VALUE OF 12 TO BE USED (SAME
              AS D).
   D     DEFAULT.  THE VALUE OF 12 IS USED.
   M     MAXIMUM.  THE MAXIMUM ALLOWABLE VALUE IS USED.  THIS
              IS THE SMALLER OF 1 LESS THAN THE NUMBER OF DATA
              POINTS AND 25.
   H     HELP.  PRINTS THE HELP MESSAGE.
   T     TERMINATE.  TERMINATES THE PROGRAM.
```

The default value can be revised by altering the DATA statement containing NCPDFT at the beginning of the FORTRAN code. In general, M does not give good results. A value greater than 12 is seldom required, and if the data is well distributed through the domain a value of 4 or 5 may be sufficient. Smaller values of NCP use less computer time.

At this point in the program the interpolations are performed as specified.

Final Menu
MENU 4

```
OPTIONS:
      1.  CREATE TIME HISTORIES FROM HONDO DATA FILES
      2.  PLOT TIME HISTORIES
      3.  SPECIFY NEW INTERPOLATION POINTS
      4.  DEFINE NEW PROBLEM (START)
      5.  TERMINATE
      6.  HELP
SELECTION:
```

Option 1 reads a HONDO.MOV file, a file created by a finite element code called HONDO, and creates time histories. The following prompts appear:

HONDO FILENAME:	Input the name of the HONDO.MOV file.
MAXIMUM TIME:	Input the maximum time (in seconds). If an arbitrarily large number is entered, histories will be created only to the maximum time contained in the HONDO.MOV file.

DO YOU WANT THE NODE NUMBERS AND COORDINATES LISTED ON TTY?

"YES" will cause a list of all the node numbers and coordinates in the finite element mesh to be listed on the screen.

NUMBER OF TIME HISTORIES TO BE CREATED: List the total number of time histories to be created.

```
SPECIFY NODE NUMBER AND PLOT OPTION:
   PLOT OPTIONS:
      1.  R DISPLACEMENT
      2.  Z DISPLACEMENT
      3.  R VELOCITY
      4.  Z VELOCITY
      5.  R ACCELERATION
      6.  Z ACCELERATION
      7.  PRESSURE
      8.  *RELIST PLOT OPTIONS*
```

Specify the node number of the node at which a time history is to be created and a plot option from the above list as prompted:

```
1  NODE NUMBER:
   PLOT OPTION:
```

Specify each of the node numbers and plot options in turn, taking care to enter them in the proper order (i.e., node number first, then plot option). Plot Option 8 will cause the Plot Option List to be relisted.

The history files are then created as specified and the file names are listed based on the function. All file names start with H (for HONDO). The second letter corresponds to the direction (R or Z). The third letter corresponds to the type (D for Displacement, V for Velocity, A for Acceleration). Pressure histories are given the letters HPP. Next is a one- to five digit number corresponding to the node number. For example, HZD245 is the name of the history file for the vertical displacement at node 245.

Option 2 creates the time history plots from the time history files. Any properly formatted files could be specified, such as those created by Option 1 or by Option 4 of MENU 1. The file is read with the following FORTRAN steps:

```
      READ(IDSK,100) IFILE
100   FORMAT(1X,2A4)
      READ(IDSK,110) ITYPE
110   FORMAT(1X,2A6)
      READ(IDSK,120) IDRCTN
120   FORMAT(1X,A1)
      READ(IDSK,130) NPT
130   FORMAT(I)
      READ(IDSK,140) (CURVE(I),I=1,NPT)
      READ(IDSK,140) (TIME(I),I=1,NPT)
140   FORMAT(*)
```

IFILE is the file name or the word that is to be used in the legend. ITYPE is the function such as displacement, velocity, etc. IDRCTN is the direction (e.g., Z or R). NPT is the number points. CURVE is an array containing the function values and TIME is an array containing the corresponding time values in seconds.

If Option 2 is selected the following prompt appears:

```
OPTIONS:
  1.  ONE CURVE
  2.  TWO CURVES--SAME TYPE
  3.  TWO CURVES--DIFFERENT TYPES
  4.  HELP
SELECTION:
```

The first option plots a single curve. The second option plots two curves of the same type. Only one y-axis is drawn. The third option plots two curves of differing types. Separate y-axes are drawn. The user is prompted for the filenames. A system-dependent prompt appears allowing the user to specify the output device. The plots are drawn using DISSPLA, a collection of plotting subroutines. DISSPLA subroutines are not an integral part of EXPERT but are called by EXPERT and so must by linked with the program.

Option 3 of MENU 4 causes MENU 2, Development of Interpolation Points, to appear. This is useful when the data points remain the same but the user desires to interpolate at different configurations of interpolation points.

Option 4 causes the program to start over again. All variables and flags are reinitialized. This does the same thing as entering special option S.

Option 5 terminates the program, same as special option T.

Note that special option R can be used to revert back to the previous option list, MENU 3--Solutions.

APPENDIX C

Sample of a Typical EXPERT Interactive Session

```
****************************************************************
*                                                              *
*  EEEEEEE  X       X  PPPPPP   EEEEEEE  RRRRRR  TTTTTTTTT     *
*  E         X     X   P     P  E        R     R     T         *
*  E          X   X    P     P  E        R     R     T         *
*  EEEEE       XX      PPPPPP   EEEEE    RRRRRR      T         *
*  E          X   X    P        E        R   R       T         *
*  E         X     X   P        E        R     R     T         *
*  EEEEEEE  X       X  P        EEEEEEE  R      R    T         *
*                                                              *
*        EXPERIMENTAL DATA INTERPOLATION PROGRAM               *
*                                                              *
****************************************************************


INPUT OF EXPERIMENTAL DATA POINTS:

  OPTIONS:
     1.  INPUT DATA AND CREATE FILE
     2.  READ DATA FILE
     3.  READ MULTIPLE DATA FILES
  SPECIAL OPTIONS:
     4.  READ E.U. TAPE AND CREATE DATA FILES
     5.  CREATE TIME HISTORIES FROM HONDO DATA FILE
     6.  PLOT TIME HISTORIES
     7.  HELP

SELECTION: 2
ENTER THE FILENAME.EXT C21
DPN        X                Y                Z

 1    1.00000E+00    -2.30000E-01    2.20000E-02
 2    1.00000E+00    -7.60000E-01    4.40000E-02
 3    1.00000E+00    -1.50000E+00    9.10000E-02
 4    1.00000E+00    -2.30000E+00    4.10000E-01
 5    1.00000E+00    -4.30000E+00    8.53000E-01
 6    1.00000E+00    -7.50000E+00    9.69000E-01
 7    3.00000E+00    -2.30000E-01    1.70000E-02
 8    3.00000E+00    -7.60000E-01    2.60000E-02
 9    3.00000E+00    -1.50000E+00    2.10000E-02
10    3.00000E+00    -2.30000E+00    2.91000E-01
11    3.00000E+00    -4.30000E+00    7.37000E-01
12    6.50000E+00    -2.30000E-01    0.00000E+00
13    6.50000E+00    -7.60000E-01    1.00000E-02
14    6.50000E+00    -1.50000E+00    1.10000E-02
15    6.50000E+00    -2.30000E+00    7.00000E-02
16    6.50000E+00    -4.30000E+00    4.12000E-01
17    1.08500E+01    -2.30000E-01    0.00000E+00
18    1.08500E+01    -7.60000E-01    0.00000E+00
19    1.08500E+01    -1.50000E+00    0.00000E+00
20    1.08500E+01    -2.30000E+00    0.00000E+00
21    1.08500E+01    -4.30000E+00    4.00000E-02
22    1.83500E+01    -2.30000E-01    0.00000E+00
23    3.58000E+01    -3.80000E-01    0.00000E+00
CORRECTIONS? N
STORE THE DATA? N
```

```
DEVELOPMENT OF INTERPOLATION POINTS:
     (POINTS AT WHICH INTERPOLATION WILL BE PERFORMED)

   OPTIONS:
       1.  READ QMESH DATA FILE
       2.  CREATE A REGULAR RECTANGULAR GRID
       3.  SPECIFY RANDOM POINTS TO BE INTERPOLATED
       4.  HELP
SELECTION: 2
FORMATION OF GRID SYSTEM:

X-COORDINATE OF THE FIRST VERTICAL GRID LINE: 0.
X-COORDINATE OF THE LAST VERTICAL GRID LINE: 11.
TOTAL NUMBER OF VERTICAL GRID LINES: 30
Y-COORDINATE OF THE FIRST HORIZONTAL GRID LINE: 0.
Y-COORDINATE OF THE LAST HORIZONTAL GRID LINE: -5.
TOTAL NUMBER OF HORIZONTAL GRID LINES: 24


SOLUTION:
   OPTIONS:
       1.  PRINT RESULTS ON TTY
       2.  CREATE DISK FILE OF RESULTS
       3.  CREATE MOVIE.BYU FILES
       4.  1 & 2
       5.  1 & 3
       6.  2 & 3
       7.  1,2, & 3
       8.  CONTINUE
       9.  HELP
SELECTION: 7

NUMBER OF CLOSEST DATA POINTS (NCP) TO BE USED
 FOR INTERPOLATION OF EACH POINT (OR TYPE HELP): 10


THE VALUE OF NCP IS 10  <------

ASSIGN THE FILENAME.EXT: RES.DAT

INTERPOLATED DATA CORRESPONDING TO DATA FILE C21

NCP= 10
NUMBER OF DATA POINTS= 23
NUMBER OF INTERPOLATION POINTS= 720

INPUT DATA POINTS:
```

X	Y	Z
1.00000E+00	-2.30000E-01	2.20000E-02
1.00000E+00	-7.60000E-01	4.40000E-02
1.00000E+00	-1.50000E+00	9.10000E-02
1.00000E+00	-2.30000E+00	4.10000E-01
1.00000E+00	-4.30000E+00	8.53000E-01
1.00000E+00	-7.50000E+00	9.69000E-01
3.00000E+00	-2.30000E-01	1.70000E-02
3.00000E+00	-7.60000E-01	2.60000E-02
3.00000E+00	-1.50000E+00	2.10000E-02
3.00000E+00	-2.30000E+00	2.91000E-01
3.00000E+00	-4.30000E+00	7.37000E-01
6.50000E+00	-2.30000E-01	0.00000E+00
6.50000E+00	-7.60000E-01	1.00000E-02
6.50000E+00	-1.50000E+00	1.10000E-02
6.50000E+00	-2.30000E+00	7.00000E-02
6.50000E+00	-4.30000E+00	4.12000E-01
1.08500E+01	-2.30000E-01	0.00000E+00

1.08500E+01	-7.60000E-01	0.00000E+00
1.08500E+01	-1.50000E+00	0.00000E+00
1.08500E+01	-2.30000E+00	0.00000E+00
1.08500E+01	-4.30000E+00	4.00000E-02
1.83500E+01	-2.30000E-01	0.00000E+00
3.58000E+01	-3.80000E-01	0.00000E+00

INTERPOLATED VALUES AT GRID POINTS:

X	Y	Z
0.00000E+00	0.00000E+00	9.35043E-03
0.00000E+00	-2.17391E-01	4.55480E-02
0.00000E+00	-4.34783E-01	5.62744E-02
0.00000E+00	-6.52174E-01	4.10893E-02
0.00000E+00	-8.69565E-01	7.26565E-02
0.00000E+00	-1.08696E+00	6.79666E-02
0.00000E+00	-1.30435E+00	5.04566E-02
0.00000E+00	-1.52174E+00	7.83430E-02
0.00000E+00	-1.73913E+00	1.65949E-01
0.00000E+00	-1.95652E+00	3.15863E-01
0.00000E+00	-2.17391E+00	4.37236E-01
0.00000E+00	-2.39130E+00	4.87483E-01
0.00000E+00	-2.60870E+00	5.31158E-01
0.00000E+00	-2.82609E+00	5.80758E-01
0.00000E+00	-3.04348E+00	6.36156E-01
0.00000E+00	-3.26087E+00	6.95364E-01
0.00000E+00	-3.47826E+00	7.55388E-01
0.00000E+00	-3.69565E+00	8.13080E-01
0.00000E+00	-3.91304E+00	8.65986E-01
0.00000E+00	-4.13043E+00	9.13201E-01
0.00000E+00	-4.34783E+00	9.56169E-01
0.00000E+00	-4.56522E+00	9.94868E-01
0.00000E+00	-4.78261E+00	1.02457E+00
0.00000E+00	-5.00000E+00	1.04241E+00
3.79310E-01	0.00000E+00	2.81694E-04
3.79310E-01	-2.17391E-01	3.56831E-02
3.79310E-01	-4.34783E-01	4.82609E-02
3.79310E-01	-6.52174E-01	3.66364E-02
3.79310E-01	-8.69565E-01	6.89199E-02
3.79310E-01	-1.08696E+00	6.89166E-02
3.79310E-01	-1.30435E+00	5.70575E-02
3.79310E-01	-1.52174E+00	8.64835E-02
3.79310E-01	-1.73913E+00	1.66811E-01
3.79310E-01	-1.95652E+00	3.04517E-01
3.79310E-01	-2.17391E+00	4.15682E-01
3.79310E-01	-2.39130E+00	4.63371E-01
3.79310E-01	-2.60870E+00	5.06032E-01
3.79310E-01	-2.82609E+00	5.54232E-01
3.79310E-01	-3.04348E+00	6.07963E-01
3.79310E-01	-3.26087E+00	6.65357E-01
3.79310E-01	-3.47826E+00	7.23539E-01
3.79310E-01	-3.69565E+00	7.79478E-01
3.79310E-01	-3.91304E+00	8.30842E-01
3.79310E-01	-4.13043E+00	8.76843E-01
3.79310E-01	-4.34783E+00	9.19056E-01
3.79310E-01	-4.56522E+00	9.57800E-01
3.79310E-01	-4.78261E+00	9.88373E-01
3.79310E-01	-5.00000E+00	1.00779E+00
7.58621E-01	0.00000E+00	-8.57730E-03
7.58621E-01	-2.17391E-01	2.60280E-02
7.58621E-01	-4.34783E-01	4.02532E-02
7.58621E-01	-6.52174E-01	3.18458E-02
7.58621E-01	-8.69565E-01	6.46799E-02
7.58621E-01	-1.08696E+00	6.87666E-02
7.58621E-01	-1.30435E+00	6.18599E-02

7.58621E-01	-1.52174E+00	9.25413E-02
7.58621E-01	-1.73913E+00	1.66272E-01
7.58621E-01	-1.95652E+00	2.93027E-01
7.58621E-01	-2.17391E+00	3.95037E-01
7.58621E-01	-2.39130E+00	4.40386E-01
7.58621E-01	-2.60870E+00	4.82064E-01
7.58621E-01	-2.82609E+00	5.28924E-01
7.58621E-01	-3.04348E+00	5.81066E-01
7.58621E-01	-3.26087E+00	6.36733E-01
7.58621E-01	-3.47826E+00	6.93161E-01
7.58621E-01	-3.69565E+00	7.47430E-01
7.58621E-01	-3.91304E+00	7.97318E-01
7.58621E-01	-4.13043E+00	8.42149E-01
7.58621E-01	-4.34783E+00	8.83616E-01
7.58621E-01	-4.56522E+00	9.22361E-01
7.58621E-01	-4.78261E+00	9.53711E-01
7.58621E-01	-5.00000E+00	9.74566E-01
1.13793E+00	0.00000E+00	-1.69475E-02
1.13793E+00	-2.17391E-01	1.66951E-02
1.13793E+00	-4.34783E-01	3.32696E-02
1.13793E+00	-6.52174E-01	2.69848E-02
1.13793E+00	-8.69565E-01	5.97772E-02
1.13793E+00	-1.08696E+00	6.65469E-02
1.13793E+00	-1.30435E+00	6.37738E-02
1.13793E+00	-1.52174E+00	9.64733E-02
1.13793E+00	-1.73913E+00	1.65341E-01
1.13793E+00	-1.95652E+00	2.83519E-01
1.13793E+00	-2.17391E+00	3.76453E-01
1.13793E+00	-2.39130E+00	4.18532E-01
1.13793E+00	-2.60870E+00	4.59445E-01
1.13793E+00	-2.82609E+00	5.05238E-01
1.13793E+00	-3.04348E+00	5.56076E-01
1.13793E+00	-3.26087E+00	6.10263E-01
1.13793E+00	-3.47826E+00	6.65100E-01
1.13793E+00	-3.69565E+00	7.17729E-01
1.13793E+00	-3.91304E+00	7.65991E-01
1.13793E+00	-4.13043E+00	8.09273E-01
1.13793E+00	-4.34783E+00	8.49889E-01
1.13793E+00	-4.56522E+00	8.88563E-01
1.13793E+00	-4.78261E+00	9.20444E-01
1.13793E+00	-5.00000E+00	9.42358E-01
1.51724E+00	0.00000E+00	-1.95016E-02
1.51724E+00	-2.17391E-01	1.14710E-02
1.51724E+00	-4.34783E-01	3.11209E-02
1.51724E+00	-6.52174E-01	2.46059E-02
1.51724E+00	-8.69565E-01	5.34761E-02
1.51724E+00	-1.08696E+00	5.70070E-02
1.51724E+00	-1.30435E+00	6.51031E-02
1.51724E+00	-1.52174E+00	9.59672E-02
1.51724E+00	-1.73913E+00	1.69271E-01
1.51724E+00	-1.95652E+00	2.82973E-01
1.51724E+00	-2.17391E+00	3.50460E-01
1.51724E+00	-2.39130E+00	3.96416E-01
1.51724E+00	-2.60870E+00	4.38398E-01
1.51724E+00	-2.82609E+00	4.85149E-01
1.51724E+00	-3.04348E+00	5.36416E-01
1.51724E+00	-3.26087E+00	5.90088E-01
1.51724E+00	-3.47826E+00	6.43047E-01
1.51724E+00	-3.69565E+00	6.92020E-01
1.51724E+00	-3.91304E+00	7.35742E-01
1.51724E+00	-4.13043E+00	7.78115E-01
1.51724E+00	-4.34783E+00	8.19167E-01
1.51724E+00	-4.56522E+00	8.57536E-01
1.51724E+00	-4.78261E+00	8.88635E-01
1.51724E+00	-5.00000E+00	9.09608E-01
1.89655E+00	0.00000E+00	-1.37407E-02

1.89655E+00	-2.17391E-01	1.32523E-02
1.89655E+00	-4.34783E-01	1.72617E-02
1.89655E+00	-6.52174E-01	2.36328E-02
1.89655E+00	-8.69565E-01	4.76969E-02
1.89655E+00	-1.08696E+00	5.29749E-02
1.89655E+00	-1.30435E+00	6.74401E-02
1.89655E+00	-1.52174E+00	8.75534E-02
1.89655E+00	-1.73913E+00	1.65598E-01
1.89655E+00	-1.95652E+00	2.50861E-01
1.89655E+00	-2.17391E+00	3.22743E-01
1.89655E+00	-2.39130E+00	3.72427E-01
1.89655E+00	-2.60870E+00	4.16605E-01
1.89655E+00	-2.82609E+00	4.65094E-01
1.89655E+00	-3.04348E+00	5.16625E-01
1.89655E+00	-3.26087E+00	5.68070E-01
1.89655E+00	-3.47826E+00	6.15804E-01
1.89655E+00	-3.69565E+00	6.62424E-01
1.89655E+00	-3.91304E+00	7.07978E-01
1.89655E+00	-4.13043E+00	7.51588E-01
1.89655E+00	-4.34783E+00	7.93436E-01
1.89655E+00	-4.56522E+00	8.31668E-01
1.89655E+00	-4.78261E+00	8.61919E-01
1.89655E+00	-5.00000E+00	8.82149E-01
2.27586E+00	0.00000E+00	-4.79637E-03
2.27586E+00	-2.17391E-01	1.78811E-02
2.27586E+00	-4.34783E-01	1.48660E-02
2.27586E+00	-6.52174E-01	2.10529E-02
2.27586E+00	-8.69565E-01	4.40411E-02
2.27586E+00	-1.08696E+00	6.28435E-02
2.27586E+00	-1.30435E+00	5.01311E-02
2.27586E+00	-1.52174E+00	7.10965E-02
2.27586E+00	-1.73913E+00	1.41089E-01
2.27586E+00	-1.95652E+00	2.12813E-01
2.27586E+00	-2.17391E+00	2.98407E-01
2.27586E+00	-2.39130E+00	3.48410E-01
2.27586E+00	-2.60870E+00	3.94545E-01
2.27586E+00	-2.82609E+00	4.43511E-01
2.27586E+00	-3.04348E+00	4.92449E-01
2.27586E+00	-3.26087E+00	5.41317E-01
2.27586E+00	-3.47826E+00	5.90831E-01
2.27586E+00	-3.69565E+00	6.39491E-01
2.27586E+00	-3.91304E+00	6.86191E-01
2.27586E+00	-4.13043E+00	7.30579E-01
2.27586E+00	-4.34783E+00	7.73320E-01
2.27586E+00	-4.56522E+00	8.12105E-01
2.27586E+00	-4.78261E+00	8.43267E-01
2.27586E+00	-5.00000E+00	8.66076E-01
2.65517E+00	0.00000E+00	3.10542E-04
2.65517E+00	-2.17391E-01	1.93100E-02
2.65517E+00	-4.34783E-01	2.03428E-02
2.65517E+00	-6.52174E-01	1.39336E-02
2.65517E+00	-8.69565E-01	4.36905E-02
2.65517E+00	-1.08696E+00	4.62012E-02
2.65517E+00	-1.30435E+00	2.34389E-02
2.65517E+00	-1.52174E+00	4.86187E-02
2.65517E+00	-1.73913E+00	1.03345E-01
2.65517E+00	-1.95652E+00	1.94796E-01
2.65517E+00	-2.17391E+00	2.78143E-01
2.65517E+00	-2.39130E+00	3.26967E-01
2.65517E+00	-2.60870E+00	3.73453E-01
2.65517E+00	-2.82609E+00	4.21200E-01
2.65517E+00	-3.04348E+00	4.71031E-01
2.65517E+00	-3.26087E+00	5.22024E-01
2.65517E+00	-3.47826E+00	5.72890E-01
2.65517E+00	-3.69565E+00	6.22379E-01
2.65517E+00	-3.91304E+00	6.69631E-01

2.65517E+00	-4.13043E+00	7.14546E-01
2.65517E+00	-4.34783E+00	7.58043E-01
2.65517E+00	-4.56522E+00	7.98597E-01
2.65517E+00	-4.78261E+00	8.34254E-01
2.65517E+00	-5.00000E+00	8.65241E-01
3.03448E+00	0.00000E+00	-1.46152E-03
3.03448E+00	-2.17391E-01	1.54631E-02
3.03448E+00	-4.34783E-01	2.01233E-02
3.03448E+00	-6.52174E-01	1.27570E-02
3.03448E+00	-8.69565E-01	3.93145E-02
3.03448E+00	-1.08696E+00	2.54853E-02
3.03448E+00	-1.30435E+00	-1.93400E-03
3.03448E+00	-1.52174E+00	2.27734E-02
3.03448E+00	-1.73913E+00	8.29519E-02
3.03448E+00	-1.95652E+00	1.78437E-01
3.03448E+00	-2.17391E+00	2.60173E-01
3.03448E+00	-2.39130E+00	3.08250E-01
3.03448E+00	-2.60870E+00	3.54634E-01
3.03448E+00	-2.82609E+00	4.03420E-01
3.03448E+00	-3.04348E+00	4.54300E-01
3.03448E+00	-3.26087E+00	5.06283E-01
3.03448E+00	-3.47826E+00	5.58052E-01
3.03448E+00	-3.69565E+00	6.08330E-01
3.03448E+00	-3.91304E+00	6.56233E-01
3.03448E+00	-4.13043E+00	7.01631E-01
3.03448E+00	-4.34783E+00	7.45515E-01
3.03448E+00	-4.56522E+00	7.89032E-01
3.03448E+00	-4.78261E+00	8.28951E-01
3.03448E+00	-5.00000E+00	8.59576E-01
3.41379E+00	0.00000E+00	-5.39788E-03
3.41379E+00	-2.17391E-01	1.06828E-02
3.41379E+00	-4.34783E-01	1.45874E-02
3.41379E+00	-6.52174E-01	9.21816E-03
3.41379E+00	-8.69565E-01	3.37499E-02
3.41379E+00	-1.08696E+00	6.94223E-03
3.41379E+00	-1.30435E+00	-2.92600E-02
3.41379E+00	-1.52174E+00	-1.99941E-03
3.41379E+00	-1.73913E+00	6.24152E-02
3.41379E+00	-1.95652E+00	1.64215E-01
3.41379E+00	-2.17391E+00	2.44649E-01
3.41379E+00	-2.39130E+00	2.88870E-01
3.41379E+00	-2.60870E+00	3.35054E-01
3.41379E+00	-2.82609E+00	3.84338E-01
3.41379E+00	-3.04348E+00	4.36184E-01
3.41379E+00	-3.26087E+00	4.89372E-01
3.41379E+00	-3.47826E+00	5.42358E-01
3.41379E+00	-3.69565E+00	5.93636E-01
3.41379E+00	-3.91304E+00	6.42093E-01
3.41379E+00	-4.13043E+00	6.87356E-01
3.41379E+00	-4.34783E+00	7.30683E-01
3.41379E+00	-4.56522E+00	7.74259E-01
3.41379E+00	-4.78261E+00	8.11207E-01
3.41379E+00	-5.00000E+00	8.33533E-01
3.79310E+00	0.00000E+00	-8.75509E-03
3.79310E+00	-2.17391E-01	6.93415E-03
3.79310E+00	-4.34783E-01	8.87644E-03
3.79310E+00	-6.52174E-01	9.22278E-03
3.79310E+00	-8.69565E-01	2.73959E-02
3.79310E+00	-1.08696E+00	-1.51684E-02
3.79310E+00	-1.30435E+00	-4.88269E-02
3.79310E+00	-1.52174E+00	-1.83714E-02
3.79310E+00	-1.73913E+00	5.25976E-02
3.79310E+00	-1.95652E+00	1.59390E-01
3.79310E+00	-2.17391E+00	2.24647E-01
3.79310E+00	-2.39130E+00	2.66219E-01
3.79310E+00	-2.60870E+00	3.11834E-01

3.79310E+00	-2.82609E+00	3.61396E-01
3.79310E+00	-3.04348E+00	4.13990E-01
3.79310E+00	-3.26087E+00	4.68017E-01
3.79310E+00	-3.47826E+00	5.21558E-01
3.79310E+00	-3.69565E+00	5.72728E-01
3.79310E+00	-3.91304E+00	6.20066E-01
3.79310E+00	-4.13043E+00	6.63618E-01
3.79310E+00	-4.34783E+00	7.05706E-01
3.79310E+00	-4.56522E+00	7.46286E-01
3.79310E+00	-4.78261E+00	7.75738E-01
3.79310E+00	-5.00000E+00	7.86358E-01
4.17241E+00	0.00000E+00	-1.11025E-02
4.17241E+00	-2.17391E-01	4.51648E-03
4.17241E+00	-4.34783E-01	4.70287E-03
4.17241E+00	-6.52174E-01	1.07081E-02
4.17241E+00	-8.69565E-01	2.12365E-02
4.17241E+00	-1.08696E+00	-3.26196E-02
4.17241E+00	-1.30435E+00	-4.55610E-02
4.17241E+00	-1.52174E+00	-2.31242E-02
4.17241E+00	-1.73913E+00	5.37002E-02
4.17241E+00	-1.95652E+00	1.55774E-01
4.17241E+00	-2.17391E+00	2.00528E-01
4.17241E+00	-2.39130E+00	2.39863E-01
4.17241E+00	-2.60870E+00	2.84459E-01
4.17241E+00	-2.82609E+00	3.33741E-01
4.17241E+00	-3.04348E+00	3.86267E-01
4.17241E+00	-3.26087E+00	4.39914E-01
4.17241E+00	-3.47826E+00	4.92235E-01
4.17241E+00	-3.69565E+00	5.40935E-01
4.17241E+00	-3.91304E+00	5.85951E-01
4.17241E+00	-4.13043E+00	6.27616E-01
4.17241E+00	-4.34783E+00	6.68105E-01
4.17241E+00	-4.56522E+00	7.03895E-01
4.17241E+00	-4.78261E+00	7.24184E-01
4.17241E+00	-5.00000E+00	7.24154E-01
4.55172E+00	0.00000E+00	-1.24615E-02
4.55172E+00	-2.17391E-01	3.27710E-03
4.55172E+00	-4.34783E-01	3.32721E-03
4.55172E+00	-6.52174E-01	1.11937E-02
4.55172E+00	-8.69565E-01	1.60696E-02
4.55172E+00	-1.08696E+00	-3.73702E-02
4.55172E+00	-1.30435E+00	-3.39485E-02
4.55172E+00	-1.52174E+00	-1.75112E-02
4.55172E+00	-1.73913E+00	6.14514E-02
4.55172E+00	-1.95652E+00	1.40902E-01
4.55172E+00	-2.17391E+00	1.73766E-01
4.55172E+00	-2.39130E+00	2.10751E-01
4.55172E+00	-2.60870E+00	2.53794E-01
4.55172E+00	-2.82609E+00	3.01897E-01
4.55172E+00	-3.04348E+00	3.52945E-01
4.55172E+00	-3.26087E+00	4.04137E-01
4.55172E+00	-3.47826E+00	4.52569E-01
4.55172E+00	-3.69565E+00	4.98727E-01
4.55172E+00	-3.91304E+00	5.41844E-01
4.55172E+00	-4.13043E+00	5.81690E-01
4.55172E+00	-4.34783E+00	6.20351E-01
4.55172E+00	-4.56522E+00	6.50813E-01
4.55172E+00	-4.78261E+00	6.63134E-01
4.55172E+00	-5.00000E+00	6.57971E-01
4.93103E+00	0.00000E+00	-1.31343E-02
4.93103E+00	-2.17391E-01	2.78233E-03
4.93103E+00	-4.34783E-01	5.72900E-03
4.93103E+00	-6.52174E-01	1.07813E-02
4.93103E+00	-8.69565E-01	1.25920E-02
4.93103E+00	-1.08696E+00	-2.14807E-02
4.93103E+00	-1.30435E+00	-1.89104E-02

4.93103E+00	-1.52174E+00	-5.63558E-03
4.93103E+00	-1.73913E+00	6.87310E-02
4.93103E+00	-1.95652E+00	1.17390E-01
4.93103E+00	-2.17391E+00	1.46094E-01
4.93103E+00	-2.39130E+00	1.80654E-01
4.93103E+00	-2.60870E+00	2.21529E-01
4.93103E+00	-2.82609E+00	2.67216E-01
4.93103E+00	-3.04348E+00	3.14776E-01
4.93103E+00	-3.26087E+00	3.60890E-01
4.93103E+00	-3.47826E+00	4.06842E-01
4.93103E+00	-3.69565E+00	4.51541E-01
4.93103E+00	-3.91304E+00	4.93070E-01
4.93103E+00	-4.13043E+00	5.31215E-01
4.93103E+00	-4.34783E+00	5.67951E-01
4.93103E+00	-4.56522E+00	5.93805E-01
4.93103E+00	-4.78261E+00	6.02212E-01
4.93103E+00	-5.00000E+00	6.01895E-01
5.31034E+00	0.00000E+00	-1.35332E-02
5.31034E+00	-2.17391E-01	2.48863E-03
5.31034E+00	-4.34783E-01	1.15487E-02
5.31034E+00	-6.52174E-01	9.55578E-03
5.31034E+00	-8.69565E-01	1.14836E-02
5.31034E+00	-1.08696E+00	3.60547E-03
5.31034E+00	-1.30435E+00	-4.88929E-03
5.31034E+00	-1.52174E+00	7.17398E-03
5.31034E+00	-1.73913E+00	6.71924E-02
5.31034E+00	-1.95652E+00	9.40578E-02
5.31034E+00	-2.17391E+00	1.19508E-01
5.31034E+00	-2.39130E+00	1.51606E-01
5.31034E+00	-2.60870E+00	1.89614E-01
5.31034E+00	-2.82609E+00	2.31308E-01
5.31034E+00	-3.04348E+00	2.73130E-01
5.31034E+00	-3.26087E+00	3.17007E-01
5.31034E+00	-3.47826E+00	3.62147E-01
5.31034E+00	-3.69565E+00	4.05817E-01
5.31034E+00	-3.91304E+00	4.46079E-01
5.31034E+00	-4.13043E+00	4.82693E-01
5.31034E+00	-4.34783E+00	5.17537E-01
5.31034E+00	-4.56522E+00	5.40763E-01
5.31034E+00	-4.78261E+00	5.52171E-01
5.31034E+00	-5.00000E+00	5.71116E-01
5.68966E+00	0.00000E+00	-1.40093E-02
5.68966E+00	-2.17391E-01	1.91363E-03
5.68966E+00	-4.34783E-01	1.24974E-02
5.68966E+00	-6.52174E-01	7.67021E-03
5.68966E+00	-8.69565E-01	1.34926E-02
5.68966E+00	-1.08696E+00	1.63363E-02
5.68966E+00	-1.30435E+00	4.06989E-03
5.68966E+00	-1.52174E+00	1.59859E-02
5.68966E+00	-1.73913E+00	5.32019E-02
5.68966E+00	-1.95652E+00	7.30518E-02
5.68966E+00	-2.17391E+00	9.57044E-02
5.68966E+00	-2.39130E+00	1.25341E-01
5.68966E+00	-2.60870E+00	1.59700E-01
5.68966E+00	-2.82609E+00	1.95831E-01
5.68966E+00	-3.04348E+00	2.35742E-01
5.68966E+00	-3.26087E+00	2.79418E-01
5.68966E+00	-3.47826E+00	3.24174E-01
5.68966E+00	-3.69565E+00	3.67214E-01
5.68966E+00	-3.91304E+00	4.06538E-01
5.68966E+00	-4.13043E+00	4.41843E-01
5.68966E+00	-4.34783E+00	4.74961E-01
5.68966E+00	-4.56522E+00	4.98796E-01
5.68966E+00	-4.78261E+00	5.22950E-01
5.68966E+00	-5.00000E+00	5.57423E-01
6.06897E+00	0.00000E+00	-1.46811E-02

6.06897E+00	-2.17391E-01	8.07223E-04
6.06897E+00	-4.34783E-01	9.54694E-03
6.06897E+00	-6.52174E-01	5.43028E-03
6.06897E+00	-8.69565E-01	1.89595E-02
6.06897E+00	-1.08696E+00	1.80873E-02
6.06897E+00	-1.30435E+00	5.94235E-03
6.06897E+00	-1.52174E+00	1.78890E-02
6.06897E+00	-1.73913E+00	4.00673E-02
6.06897E+00	-1.95652E+00	5.56577E-02
6.06897E+00	-2.17391E+00	7.55199E-02
6.06897E+00	-2.39130E+00	1.02732E-01
6.06897E+00	-2.60870E+00	1.32841E-01
6.06897E+00	-2.82609E+00	1.67376E-01
6.06897E+00	-3.04348E+00	2.07417E-01
6.06897E+00	-3.26087E+00	2.51194E-01
6.06897E+00	-3.47826E+00	2.95919E-01
6.06897E+00	-3.69565E+00	3.38696E-01
6.06897E+00	-3.91304E+00	3.77420E-01
6.06897E+00	-4.13043E+00	4.11690E-01
6.06897E+00	-4.34783E+00	4.43378E-01
6.06897E+00	-4.56522E+00	4.72295E-01
6.06897E+00	-4.78261E+00	5.07059E-01
6.06897E+00	-5.00000E+00	5.39060E-01
6.44828E+00	0.00000E+00	-1.52643E-02
6.44828E+00	-2.17391E-01	-6.77360E-04
6.44828E+00	-4.34783E-01	6.20566E-03
6.44828E+00	-6.52174E-01	3.37908E-03
6.44828E+00	-8.69565E-01	1.79046E-02
6.44828E+00	-1.08696E+00	1.39497E-02
6.44828E+00	-1.30435E+00	2.34699E-03
6.44828E+00	-1.52174E+00	1.36318E-02
6.44828E+00	-1.73913E+00	3.00514E-02
6.44828E+00	-1.95652E+00	4.17413E-02
6.44828E+00	-2.17391E+00	5.83708E-02
6.44828E+00	-2.39130E+00	8.33105E-02
6.44828E+00	-2.60870E+00	1.12039E-01
6.44828E+00	-2.82609E+00	1.46513E-01
6.44828E+00	-3.04348E+00	1.86738E-01
6.44828E+00	-3.26087E+00	2.30802E-01
6.44828E+00	-3.47826E+00	2.75776E-01
6.44828E+00	-3.69565E+00	3.18623E-01
6.44828E+00	-3.91304E+00	3.57098E-01
6.44828E+00	-4.13043E+00	3.90657E-01
6.44828E+00	-4.34783E+00	4.21338E-01
6.44828E+00	-4.56522E+00	4.54214E-01
6.44828E+00	-4.78261E+00	4.88259E-01
6.44828E+00	-5.00000E+00	5.10276E-01
6.82759E+00	0.00000E+00	-1.53307E-02
6.82759E+00	-2.17391E-01	-1.91982E-03
6.82759E+00	-4.34783E-01	4.27927E-03
6.82759E+00	-6.52174E-01	1.68677E-03
6.82759E+00	-8.69565E-01	1.53746E-02
6.82759E+00	-1.08696E+00	9.74137E-03
6.82759E+00	-1.30435E+00	-2.75618E-03
6.82759E+00	-1.52174E+00	8.49327E-03
6.82759E+00	-1.73913E+00	2.17066E-02
6.82759E+00	-1.95652E+00	2.88468E-02
6.82759E+00	-2.17391E+00	4.20940E-02
6.82759E+00	-2.39130E+00	6.56053E-02
6.82759E+00	-2.60870E+00	9.37885E-02
6.82759E+00	-2.82609E+00	1.28453E-01
6.82759E+00	-3.04348E+00	1.69309E-01
6.82759E+00	-3.26087E+00	2.14145E-01
6.82759E+00	-3.47826E+00	2.59737E-01
6.82759E+00	-3.69565E+00	3.02748E-01
6.82759E+00	-3.91304E+00	3.40639E-01

6.82759E+00	-4.13043E+00	3.72565E-01
6.82759E+00	-4.34783E+00	4.01471E-01
6.82759E+00	-4.56522E+00	4.33644E-01
6.82759E+00	-4.78261E+00	4.61232E-01
6.82759E+00	-5.00000E+00	4.69173E-01
7.20690E+00	0.00000E+00	-1.48597E-02
7.20690E+00	-2.17391E-01	-2.43650E-03
7.20690E+00	-4.34783E-01	2.53028E-03
7.20690E+00	-6.52174E-01	8.24039E-04
7.20690E+00	-8.69565E-01	1.32140E-02
7.20690E+00	-1.08696E+00	4.66082E-03
7.20690E+00	-1.30435E+00	-7.59177E-03
7.20690E+00	-1.52174E+00	4.80018E-03
7.20690E+00	-1.73913E+00	1.42211E-02
7.20690E+00	-1.95652E+00	1.65388E-02
7.20690E+00	-2.17391E+00	2.85965E-02
7.20690E+00	-2.39130E+00	5.00290E-02
7.20690E+00	-2.60870E+00	7.80988E-02
7.20690E+00	-2.82609E+00	1.13722E-01
7.20690E+00	-3.04348E+00	1.55962E-01
7.20690E+00	-3.26087E+00	2.01961E-01
7.20690E+00	-3.47826E+00	2.47847E-01
7.20690E+00	-3.69565E+00	2.89637E-01
7.20690E+00	-3.91304E+00	3.24144E-01
7.20690E+00	-4.13043E+00	3.51148E-01
7.20690E+00	-4.34783E+00	3.77743E-01
7.20690E+00	-4.56522E+00	4.06383E-01
7.20690E+00	-4.78261E+00	4.23955E-01
7.20690E+00	-5.00000E+00	4.17145E-01
7.58621E+00	0.00000E+00	-1.36951E-02
7.58621E+00	-2.17391E-01	-2.09346E-03
7.58621E+00	-4.34783E-01	1.43488E-03
7.58621E+00	-6.52174E-01	2.05467E-03
7.58621E+00	-8.69565E-01	1.15675E-02
7.58621E+00	-1.08696E+00	-1.27635E-05
7.58621E+00	-1.30435E+00	-8.50095E-03
7.58621E+00	-1.52174E+00	2.96871E-03
7.58621E+00	-1.73913E+00	8.49733E-03
7.58621E+00	-1.95652E+00	7.09639E-03
7.58621E+00	-2.17391E+00	1.93411E-02
7.58621E+00	-2.39130E+00	3.78685E-02
7.58621E+00	-2.60870E+00	6.60053E-02
7.58621E+00	-2.82609E+00	1.02772E-01
7.58621E+00	-3.04348E+00	1.46233E-01
7.58621E+00	-3.26087E+00	1.92529E-01
7.58621E+00	-3.47826E+00	2.36791E-01
7.58621E+00	-3.69565E+00	2.74036E-01
7.58621E+00	-3.91304E+00	3.00992E-01
7.58621E+00	-4.13043E+00	3.23430E-01
7.58621E+00	-4.34783E+00	3.47747E-01
7.58621E+00	-4.56522E+00	3.70963E-01
7.58621E+00	-4.78261E+00	3.77139E-01
7.58621E+00	-5.00000E+00	3.58319E-01
7.96552E+00	0.00000E+00	-1.19116E-02
7.96552E+00	-2.17391E-01	-9.95747E-04
7.96552E+00	-4.34783E-01	1.27041E-03
7.96552E+00	-6.52174E-01	3.50246E-03
7.96552E+00	-8.69565E-01	1.04509E-02
7.96552E+00	-1.08696E+00	-3.09488E-03
7.96552E+00	-1.30435E+00	-5.02041E-03
7.96552E+00	-1.52174E+00	2.91651E-03
7.96552E+00	-1.73913E+00	4.95878E-03
7.96552E+00	-1.95652E+00	2.32754E-03
7.96552E+00	-2.17391E+00	1.32833E-02
7.96552E+00	-2.39130E+00	2.94503E-02
7.96552E+00	-2.60870E+00	5.75705E-02

7.96552E+00	-2.82609E+00	9.50665E-02
7.96552E+00	-3.04348E+00	1.38649E-01
7.96552E+00	-3.26087E+00	1.83110E-01
7.96552E+00	-3.47826E+00	2.22226E-01
7.96552E+00	-3.69565E+00	2.50083E-01
7.96552E+00	-3.91304E+00	2.70190E-01
7.96552E+00	-4.13043E+00	2.89159E-01
7.96552E+00	-4.34783E+00	3.11256E-01
7.96552E+00	-4.56522E+00	3.28095E-01
7.96552E+00	-4.78261E+00	3.23673E-01
7.96552E+00	-5.00000E+00	2.99004E-01
8.34483E+00	0.00000E+00	-9.75365E-03
8.34483E+00	-2.17391E-01	5.82105E-04
8.34483E+00	-4.34783E-01	2.14473E-03
8.34483E+00	-6.52174E-01	4.69102E-03
8.34483E+00	-8.69565E-01	9.76734E-03
8.34483E+00	-1.08696E+00	-3.51400E-03
8.34483E+00	-1.30435E+00	-1.47423E-03
8.34483E+00	-1.52174E+00	4.19620E-03
8.34483E+00	-1.73913E+00	3.66400E-03
8.34483E+00	-1.95652E+00	3.67493E-03
8.34483E+00	-2.17391E+00	1.01126E-02
8.34483E+00	-2.39130E+00	2.43674E-02
8.34483E+00	-2.60870E+00	5.21237E-02
8.34483E+00	-2.82609E+00	8.93355E-02
8.34483E+00	-3.04348E+00	1.31009E-01
8.34483E+00	-3.26087E+00	1.70232E-01
8.34483E+00	-3.47826E+00	1.99142E-01
8.34483E+00	-3.69565E+00	2.17930E-01
8.34483E+00	-3.91304E+00	2.33370E-01
8.34483E+00	-4.13043E+00	2.49701E-01
8.34483E+00	-4.34783E+00	2.69660E-01
8.34483E+00	-4.56522E+00	2.80108E-01
8.34483E+00	-4.78261E+00	2.68065E-01
8.34483E+00	-5.00000E+00	2.47123E-01
8.72414E+00	0.00000E+00	-7.56513E-03
8.72414E+00	-2.17391E-01	2.26593E-03
8.72414E+00	-4.34783E-01	4.06606E-03
8.72414E+00	-6.52174E-01	5.44354E-03
8.72414E+00	-8.69565E-01	9.35820E-03
8.72414E+00	-1.08696E+00	-2.60382E-04
8.72414E+00	-1.30435E+00	1.62822E-03
8.72414E+00	-1.52174E+00	6.13212E-03
8.72414E+00	-1.73913E+00	4.44332E-03
8.72414E+00	-1.95652E+00	8.34318E-03
8.72414E+00	-2.17391E+00	9.01468E-03
8.72414E+00	-2.39130E+00	2.17091E-02
8.72414E+00	-2.60870E+00	4.84904E-02
8.72414E+00	-2.82609E+00	8.38062E-02
8.72414E+00	-3.04348E+00	1.20606E-01
8.72414E+00	-3.26087E+00	1.49918E-01
8.72414E+00	-3.47826E+00	1.67794E-01
8.72414E+00	-3.69565E+00	1.80920E-01
8.72414E+00	-3.91304E+00	1.93126E-01
8.72414E+00	-4.13043E+00	2.07481E-01
8.72414E+00	-4.34783E+00	2.25411E-01
8.72414E+00	-4.56522E+00	2.30394E-01
8.72414E+00	-4.78261E+00	2.15884E-01
8.72414E+00	-5.00000E+00	2.11590E-01
9.10345E+00	0.00000E+00	-5.71997E-03
9.10345E+00	-2.17391E-01	3.65164E-03
9.10345E+00	-4.34783E-01	7.01270E-03
9.10345E+00	-6.52174E-01	5.57252E-03
9.10345E+00	-8.69565E-01	9.05403E-03
9.10345E+00	-1.08696E+00	6.59875E-03
9.10345E+00	-1.30435E+00	3.79571E-03

9.10345E+00	-1.52174E+00	7.95729E-03
9.10345E+00	-1.73913E+00	7.03566E-03
9.10345E+00	-1.95652E+00	1.14100E-02
9.10345E+00	-2.17391E+00	8.90171E-03
9.10345E+00	-2.39130E+00	2.02913E-02
9.10345E+00	-2.60870E+00	4.52228E-02
9.10345E+00	-2.82609E+00	7.64315E-02
9.10345E+00	-3.04348E+00	1.04457E-01
9.10345E+00	-3.26087E+00	1.21604E-01
9.10345E+00	-3.47826E+00	1.33109E-01
9.10345E+00	-3.69565E+00	1.42480E-01
9.10345E+00	-3.91304E+00	1.52554E-01
9.10345E+00	-4.13043E+00	1.65431E-01
9.10345E+00	-4.34783E+00	1.81465E-01
9.10345E+00	-4.56522E+00	1.82847E-01
9.10345E+00	-4.78261E+00	1.73202E-01
9.10345E+00	-5.00000E+00	1.85918E-01
9.48276E+00	0.00000E+00	-4.55214E-03
9.48276E+00	-2.17391E-01	4.37506E-03
9.48276E+00	-4.34783E-01	9.45133E-03
9.48276E+00	-6.52174E-01	4.93085E-03
9.48276E+00	-8.69565E-01	8.72579E-03
9.48276E+00	-1.08696E+00	1.04470E-02
9.48276E+00	-1.30435E+00	4.58252E-03
9.48276E+00	-1.52174E+00	8.95021E-03
9.48276E+00	-1.73913E+00	1.12255E-02
9.48276E+00	-1.95652E+00	1.23987E-02
9.48276E+00	-2.17391E+00	8.64239E-03
9.48276E+00	-2.39130E+00	1.88862E-02
9.48276E+00	-2.60870E+00	4.08291E-02
9.48276E+00	-2.82609E+00	6.51212E-02
9.48276E+00	-3.04348E+00	8.13289E-02
9.48276E+00	-3.26087E+00	9.15472E-02
9.48276E+00	-3.47826E+00	9.89619E-02
9.48276E+00	-3.69565E+00	1.05987E-01
9.48276E+00	-3.91304E+00	1.14696E-01
9.48276E+00	-4.13043E+00	1.26425E-01
9.48276E+00	-4.34783E+00	1.40723E-01
9.48276E+00	-4.56522E+00	1.41306E-01
9.48276E+00	-4.78261E+00	1.45006E-01
9.48276E+00	-5.00000E+00	1.63057E-01
9.86207E+00	0.00000E+00	-4.28600E-03
9.86207E+00	-2.17391E-01	4.18163E-03
9.86207E+00	-4.34783E-01	8.97896E-03
9.86207E+00	-6.52174E-01	3.46294E-03
9.86207E+00	-8.69565E-01	8.33598E-03
9.86207E+00	-1.08696E+00	1.05254E-02
9.86207E+00	-1.30435E+00	3.72525E-03
9.86207E+00	-1.52174E+00	8.57172E-03
9.86207E+00	-1.73913E+00	1.44160E-02
9.86207E+00	-1.95652E+00	1.10344E-02
9.86207E+00	-2.17391E+00	7.29176E-03
9.86207E+00	-2.39130E+00	1.64524E-02
9.86207E+00	-2.60870E+00	3.40044E-02
9.86207E+00	-2.82609E+00	4.85492E-02
9.86207E+00	-3.04348E+00	5.76137E-02
9.86207E+00	-3.26087E+00	6.37022E-02
9.86207E+00	-3.47826E+00	6.86139E-02
9.86207E+00	-3.69565E+00	7.42026E-02
9.86207E+00	-3.91304E+00	8.19809E-02
9.86207E+00	-4.13043E+00	9.27254E-02
9.86207E+00	-4.34783E+00	1.05473E-01
9.86207E+00	-4.56522E+00	1.09000E-01
9.86207E+00	-4.78261E+00	1.20758E-01
9.86207E+00	-5.00000E+00	1.46076E-01
1.02414E+01	0.00000E+00	-4.96649E-03

1.02414E+01	-2.17391E-01	2.99623E-03
1.02414E+01	-4.34783E-01	6.50703E-03
1.02414E+01	-6.52174E-01	1.25581E-03
1.02414E+01	-8.69565E-01	7.96838E-03
1.02414E+01	-1.08696E+00	7.80627E-03
1.02414E+01	-1.30435E+00	1.27974E-03
1.02414E+01	-1.52174E+00	6.60187E-03
1.02414E+01	-1.73913E+00	1.26974E-02
1.02414E+01	-1.95652E+00	7.45866E-03
1.02414E+01	-2.17391E+00	4.32127E-03
1.02414E+01	-2.39130E+00	1.23651E-02
1.02414E+01	-2.60870E+00	2.39231E-02
1.02414E+01	-2.82609E+00	3.19277E-02
1.02414E+01	-3.04348E+00	3.72313E-02
1.02414E+01	-3.26087E+00	4.08270E-02
1.02414E+01	-3.47826E+00	4.41574E-02
1.02414E+01	-3.69565E+00	4.87193E-02
1.02414E+01	-3.91304E+00	5.56687E-02
1.02414E+01	-4.13043E+00	6.54256E-02
1.02414E+01	-4.34783E+00	7.68348E-02
1.02414E+01	-4.56522E+00	8.59873E-02
1.02414E+01	-4.78261E+00	9.81016E-02
1.02414E+01	-5.00000E+00	1.37774E-01
1.06207E+01	0.00000E+00	-6.38935E-03
1.06207E+01	-2.17391E-01	9.92911E-04
1.06207E+01	-4.34783E-01	3.21262E-03
1.06207E+01	-6.52174E-01	-1.40977E-03
1.06207E+01	-8.69565E-01	5.64327E-03
1.06207E+01	-1.08696E+00	3.71844E-03
1.06207E+01	-1.30435E+00	-2.24215E-03
1.06207E+01	-1.52174E+00	3.27677E-03
1.06207E+01	-1.73913E+00	8.06561E-03
1.06207E+01	-1.95652E+00	2.45939E-03
1.06207E+01	-2.17391E+00	-1.51188E-04
1.06207E+01	-2.39130E+00	6.64575E-03
1.06207E+01	-2.60870E+00	1.38113E-02
1.06207E+01	-2.82609E+00	1.89439E-02
1.06207E+01	-3.04348E+00	2.20439E-02
1.06207E+01	-3.26087E+00	2.39514E-02
1.06207E+01	-3.47826E+00	2.59558E-02
1.06207E+01	-3.69565E+00	2.94011E-02
1.06207E+01	-3.91304E+00	3.52901E-02
1.06207E+01	-4.13043E+00	4.38898E-02
1.06207E+01	-4.34783E+00	5.41979E-02
1.06207E+01	-4.56522E+00	6.33392E-02
1.06207E+01	-4.78261E+00	7.98230E-02
1.06207E+01	-5.00000E+00	1.40311E-01
1.10000E+01	0.00000E+00	-8.06924E-03
1.10000E+01	-2.17391E-01	-1.34529E-03
1.10000E+01	-4.34783E-01	3.75290E-04
1.10000E+01	-6.52174E-01	-4.06636E-03
1.10000E+01	-8.69565E-01	2.40442E-03
1.10000E+01	-1.08696E+00	3.50435E-05
1.10000E+01	-1.30435E+00	-5.92910E-03
1.10000E+01	-1.52174E+00	-6.20923E-04
1.10000E+01	-1.73913E+00	3.32442E-03
1.10000E+01	-1.95652E+00	-2.63070E-03
1.10000E+01	-2.17391E+00	-5.34738E-03
1.10000E+01	-2.39130E+00	6.99804E-04
1.10000E+01	-2.60870E+00	6.37505E-03
1.10000E+01	-2.82609E+00	9.88373E-03
1.10000E+01	-3.04348E+00	1.12746E-02
1.10000E+01	-3.26087E+00	1.14413E-02
1.10000E+01	-3.47826E+00	1.17270E-02
1.10000E+01	-3.69565E+00	1.35291E-02
1.10000E+01	-3.91304E+00	1.79043E-02

```
   1.10000E+01   -4.13043E+00    2.51729E-02
   1.10000E+01   -4.34783E+00    3.41761E-02
   1.10000E+01   -4.56522E+00    4.12986E-02
   1.10000E+01   -4.78261E+00    6.87431E-02
   1.10000E+01   -5.00000E+00    1.54832E-01
ASSIGN THE GEOMETRY FILENAME.EXT GEOM.PLT
ASSIGN THE FUNCTION FILENAME.EXT FUNCT.PLT

COMMAND COMPLETED


SOLUTION:
   OPTIONS:
       1.  PRINT RESULTS ON TTY
       2.  CREATE DISK FILE OF RESULTS
       3.  CREATE MOVIE.BYU FILES
       4.  1 & 2
       5.  1 & 3
       6.  2 & 3
       7.  1,2, & 3
       8.  CONTINUE
       9.  HELP
SELECTION: 8


 OPTIONS:
       1.  CREATE TIME HISTORIES FROM HONDO DATA FILES
       2.  PLOT TIME HISTORIES
       3.  SPECIFY NEW INTERPOLATION POINTS
       4.  DEFINE NEW PROBLEM (START)
       5.  TERMINATE
       6.  HELP
SELECTION: T
FORTRAN STOP
```

APPENDIX D

COORD Program Listing

```
      PROGRAM COORD
C                                                                        C
C************************************************************************C
C                                                                        C
C  PROGRAM COORD-ALLOWS A GRAPH TO BE PLACED ON A DIGITIZER AND          C
C                TRACED, PRODUCING A STRING OF COORDINATES THAT HAVE     C
C                BEEN TRANSFORMED FROM THE GLOBAL COORDINATES OF THE     C
C                DIGITIZER TO THE LOCAL COORDINATES OF THE GRAPH         C
C                                                                        C
C************************************************************************C
C                                                                        C
C       THIS PROGRAM WAS WRITTEN FOR THE SUPERSET COMPUTER FOR USE       C
C             WITH THE SUMMAGRAPHICS BIT PAD ONE DIGITIZER               C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SUBROUTINES CALLED:                                                   C
C      PTPAD   =PRODUCES THE COORDINATES OF A SINGLE POINT IN THE        C
C               GLOBAL COORDINATE SYSTEM                                 C
C      STRPAD  =PRODUCES STREAM OF COORDINATES FROM DIGITIZER, TRANS-    C
C               FORMS THEM TO THE LOCAL COORDINATE SYSTEM, AND CREATES C
C               FILES OF THE RESULTS                                     C
C                                                                        C
C************************************************************************C
C                                                                        C
C  SYSTEM ROUTINES CALLED:  INTERACT                                     C
C                                                                        C
C************************************************************************C
C                                                                        C
C  IMPORTANT INTERNAL VARIABLES:                                         C
C     THETA  =ANGLE BETWEEN LOCAL AND GLOBAL COORDINATE AXES (ANGLE OF C
C             TILT OF GRAPH PLACED ON THE DIGITIZER)                     C
C     XSCL   =SCALE FACTOR TO TRANSFORM X-COORDINATES FROM GLOBAL TO     C
C             LOCAL                                                      C
C     YSCL   =SCALE FACTOR TO TRANSFORM Y-COORDINATES FROM GLOBAL TO     C
C             LOCAL                                                      C
C                                                                        C
C************************************************************************C
C                                                                        C
      DIMENSION NAME(2),NAMET(2)
      INTEGER ANS
      CALL INITBITPAD('POINT',2)
 10   WRITE(6,20)
 20   FORMAT('PLOT NUMBER:')
      READ(5,30) NPN
 30   FORMAT(I6)
      ENCODE(12,40,NAME) NPN
 40   FORMAT('C',I6)
      ENCODE(12,50,NAMET) NPN
 50   FORMAT('Z',I6)
 60   WRITE(6,70)
 70   FORMAT('LOCATE THE ORIGIN')
      CALL PTPAD(IXORIG,IYORIG)
      WRITE(6,80)
 80   FORMAT('REGISTER?(Y/N)[Y]')
      READ(5,90) ANS
 90   FORMAT(A1)
      IF(ANS.EQ.1HN) GO TO 60
      XORIG=FLOAT(IXORIG)
      YORIG=FLOAT(IYORIG)
100   WRITE(6,110)
110   FORMAT('LOCATE A POINT ON THE X-AXIS:')
      CALL PTPAD(IX1,IY1)
      WRITE(6,80)
```

```
      READ(5,90) ANS
      IF(ANS.EQ.1HN) GO TO 100
      X1=FLOAT(IX1)
      Y1=FLOAT(IY1)
      WRITE(6,120)
120   FORMAT('X-COORD IN LOCAL SYSTEM:')
      READ(5,130) X1LOC
130   FORMAT(U)
140   WRITE(6,150)
150   FORMAT('LOCATE A POINT ON THE Y-AXIS:')
      CALL PTPAD(IX2,IY2)
      WRITE(6,80)
      READ(5,90) ANS
      IF(ANS.EQ.1HN) GO TO 140
      Y2=FLOAT(IY2)
      WRITE(6,160)
160   FORMAT('Y-COORD IN LOCAL SYSTEM:')
      READ(5,130) Y2LOC
      THETA=ATAN((Y1-YORIG)/(X1-XORIG))
      XSCL=X1LOC/(X1-XORIG)*COS(THETA)
      YSCL=Y2LOC/(Y2-YORIG)*COS(THETA)
      WRITE(6,170)
170   FORMAT('READY FOR POINTS'//'  X        Y')
      CALL STRPAD(XSCL,YSCL,THETA,XORIG,YORIG,NAME,NAMET,NPN)
      WRITE(6,180)
180   FORMAT('DO YOU WANT TO DO ANOTHER GRAPH?')
      READ(5,90) ANS
      IF(ANS.EQ.1HY) GO TO 10
      STOP
      END

      SUBROUTINE PTPAD(IX,IY)
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE PTPAD-PRODUCES A SINGLE SET OF COORDINATES IN THE GLOBAL C
C                   SYSTEM FROM THE DIGITIZER                          C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINES CALLED: MZZSEROUT (PART OF INTERACT, PRODUCES "BEEP")   C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  INPUT ARGUMENTS: NONE                                               C
C                                                                      C
C  OUTPUT ARGUMENTS: IX, IY                                            C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  IMPORTANT INTERNAL VARIABLES:                                       C
C       IX =X-COORDINATE OF A POINT                                    C
C       IY =Y-COORDINATE OF A POINT                                    C
C                                                                      C
C**********************************************************************C
C                                                                      C
 10   J=INPUTBITPAD(IX,IY)
      IF(J.LE.0) GO TO 10
      CALL MZZSEROUT(1,7,6,0,0)
      WRITE(6,20) IX,IY
 20   FORMAT(I5,'   ',I5)
      RETURN
      END
```

```
      SUBROUTINE STRPAD(XSCALE,YSCALE,ANGLE,XORIG,YORIG,NAME,NAMET,NPN)
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINE STRPAD-PRODUCES A STRING OF COORDINATES FROM THE DIG-     C
C                    ITIZER, TRANSFORMS TO LOCAL COORDINATES, AND       C
C                    CREATES FILES                                      C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  SUBROUTINES CALLED: INITBITPAD, MZZSEROUT (PART OF INTERACT)         C
C                      REDUCE,OPEN,CLOSE                                C
C                                                                       C
C***********************************************************************C
C                                                                       C
C  INPUT ARGUMENTS: ALL                                                 C
C                                                                       C
C  OUTPUT ARGUMENTS: NONE                                               C
C                                                                       C
C***********************************************************************C
C                                                                       C
      DIMENSION X(4000),Y(4000)
      DIMENSION NAME(2),NAMET(2),BLANK(2)
      INTEGER ANS,BLANK
      DATA BLANK(1)/'      '/,BLANK(2)/'      '/
      CALL INITBITPAD('STREAM',40)
      KOUNT=0
 10   J=INPUTBITPAD(IXX,IYY)
      IF(J) 40,10,20
 20   KOUNT=KOUNT+1
C     CALL MZZSEROUT(1,7,6,0,0)
      XX=FLOAT(IXX)
      YY=FLOAT(IYY)
      DELX=XX-XORIG
      DELY=YY-YORIG
      ALPHA=ATAN(DELY/DELX)-ANGLE
      VLEN=SQRT(DELY**2+DELX**2)
      X(KOUNT)=XSCALE*VLEN*COS(ALPHA)
      Y(KOUNT)=YSCALE*VLEN*SIN(ALPHA)
      WRITE(6,30) X(KOUNT),Y(KOUNT)
 30   FORMAT(F12.7,7X,F12.7)
      GO TO 10
 40   WRITE(6,50)
 50   FORMAT('DO YOU WANT A FULL FILE CREATED?')
      READ(5,60) ANS
 60   FORMAT(A1)
      IF(ANS.NE.1HY) GO TO 120
      CALL OPEN(NAME,IUNIT)
      WRITE(IUNIT,70) NAME
 70   FORMAT(' FILE: ',2A6)
      WRITE(IUNIT,80) KOUNT
 80   FORMAT(I5)
      WRITE(IUNIT,90)
 90   FORMAT(6X,'X',16X,'Y')
      DO 110 I=1,KOUNT
         WRITE(IUNIT,100) X(I),Y(I)
100      FORMAT(F12.7,5X,F12.7)
110   CONTINUE
      CALL CLOSE
120   WRITE(6,130)
130   FORMAT('DO YOU WANT A REDUCED FILE CREATED?')
      READ(5,60) ANS
      IF(ANS.NE.1HY) GO TO 140
      CALL REDUCE(X,Y,NPN,KOUNT)
140   WRITE(6,150)
150   FORMAT('DO YOU WANT A THTEST FILE CREATED?')
```

```
      READ(5,60) ANS
      IF(ANS.EQ.1HY) THEN
         CALL OPEN(NAMET,IUNIT)
         WRITE(IUNIT,160) NAMET
         WRITE(IUNIT,160) BLANK
         WRITE(IUNIT,160) BLANK
         WRITE(IUNIT,170) KOUNT
         WRITE(IUNIT,180) (Y(I),I=1,KOUNT)
         WRITE(IUNIT,180) (X(I),I=1,KOUNT)
160      FORMAT(1X,2A6)
170      FORMAT(I6)
180      FORMAT(5F12.7)
         CALL CLOSE
      ENDIF
      RETURN
      END


      SUBROUTINE REDUCE(X,Y,NUMBER,KOUNT)
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE REDUCE-DETERMINES WHICH SETS OF COORDINATES ARE NEAREST  C
C                    THE COORDINATES OF INTEREST AND PLACES THEM IN A  C
C                    FILE                                              C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE CALLED BY STRPAD                                         C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINES CALLED: OPEN,CLOSE                                      C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  INPUT ARGUMENTS: X,Y,NUMBER,KOUNT                                   C
C                                                                      C
C  OUTPUT ARGUMENTS: NONE                                              C
C                                                                      C
C**********************************************************************C
C                                                                      C
C  IMPORTANT INTERNAL VARIABLES:                                       C
C       NEWKNT =NUMBER OF POINTS SAVED IN FILE                         C
C       XR     =REDUCED X ARRAY                                        C
C       YR     =REDUCED Y ARRAY                                        C
C                                                                      C
C**********************************************************************C
C                                                                      C
      DIMENSION X(KOUNT),Y(KOUNT),XR(1000),YR(1000)
      DIMENSION NAMER(2)
      WRITE(6,10)
 10   FORMAT('TIMESTEP INCREMENT: ')
      READ(6,20) TINCR
 20   FORMAT(U)
      NEWKNT=0
      DO 40 I=1,KOUNT
          XDIV=X(I)/TINCR
          IXDIV=IFIX(XDIV)
          FIXDIV=FLOAT(IXDIV)
          DELTA=XDIV-FIXDIV
          IF(DELTA.GT..05) GO TO 30
          IF(DELTA.LT..05) THEN
              NEWKNT=NEWKNT+1
              XR(NEWKNT)=X(I)
              YR(NEWKNT)=Y(I)
```

```
                GO TO 40
            ENDIF
 30         JXDIV=IFIX(XDIV+1.)
            FJXDIV=FLOAT(JXDIV)
            DELTAJ=ABS(XDIV-FJXDIV)
            IF(DELTAJ.LT..05) THEN
                NEWKNT=NEWKNT+1
                XR(NEWKNT)=X(I)
                YR(NEWKNT)=Y(I)
            ENDIF
 40     CONTINUE
        ENCODE(12,50,NAMER) NUMBER
 50     FORMAT('RE',I6)
        CALL OPEN(NAMER,IUNIT)
        WRITE(IUNIT,60) NAMER
 60     FORMAT(2A6)
        WRITE(IUNIT,70) NEWKNT
 70     FORMAT(I6)
        WRITE(IUNIT,80)
 80     FORMAT(8X,'X',16X,'Y')
        DO 100 I=1,NEWKNT
            WRITE(IUNIT,90) XR(I),YR(I)
 90         FORMAT(F12.5,5X,F12.5)
100     CONTINUE
        CALL CLOSE
        RETURN
        END


        SUBROUTINE OPEN(FNAME,IUNIT)
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE OPEN-OPENS A DISK FILE ON THE SUPERSET COMPUTER          C
C                                                                      C
C**********************************************************************C
C                                                                      C
        COMMON/UNIT/ILUN,LUN
        INTEGER LSTRING(10)
        INTEGER FNAME(2)
        IUNIT='!BYU'
        ENCODE(60,101,LSTRING) FNAME
101     FORMAT('#ACTIVATE U=!BYU,E=',2A)
        CALL OPSYSTEM(LSTRING,60)
        RETURN
        END

        SUBROUTINE CLOSE
C                                                                      C
C**********************************************************************C
C                                                                      C
C  SUBROUTINE CLOSE-CLOSES A DISK FILE ON THE SUPERSET COMPUTER        C
C                                                                      C
C**********************************************************************C
C                                                                      C
        CALL OPSYSTEM('#DEACTIVATE U=!BYU',18)
        RETURN
        END

C                                                                      C
C**********************************************************************C
C                                                                      C
C  THE FOLLOWING IS THE LINK FILE FOR THE SUPERSET.  NODE !WORKALLEN   C
```

```
C  WOULD BE REPLACED BY THE ACTUAL NODE NAME.  THIS IS NOT PART OF       C
C  THE PROGRAM.                                                          C
C                                                                        C
C************************************************************************C
C                                                                        C
      MAIN COORD
      IMAGE COORD
      SCAN N=!WORKALLEN
      SCAN N=INTERACT
```

APPENDIX E

HONDO Input for CHEAT

```
  CHEAT TEST, SI UNITS 1 GAUSS POINTS
  5  951  873    0     0    1        0.0     0     2    0     1     0   20          0
      0.0       0.15    0.0015       0.0        0.15   0.0015         1.0
  1
  1
  1    7    1.61E+3
LAYER 1     SE7
 3.978E7    8.756E7    2.71E9    7.860E4        0.0
     0.0        0.0    -.0005    4.551E4
 -0.0633    4.661E5    -0.095    3.909E6
     0.0
     0.0
     0.0
  2    7    1.75E+3
LAYER 2     SE7
 2.048E8    5.323E8  7.511E10    3.634E5        0.0
     0.0        0.0  -.000322    2.296E5
 -.00641    8.274E5  -0.00995    1.896E6
      0.
      0.
      0.
  3    7    1.84E+3
LAYER 3     SE7
 3.647E8    9.480E8  1.019E11    2.461E5        0.0
     0.0        0.0 -.0001768    2.296E5
-.003584    8.274E5     -0.01    6.082E6
      0.
      0.
      0.
  4   10  1.640E+3
TNT HIGH EXPLOSIVE
 1.043E3
 1.043E3
    3.16
      5.
      0.
      0.
  5    6  1.840E+3
PSUEDO MATERIAL AROUND HIGH EXPLOSIVE
 1.034E9
     0.3
 3.447E7
 6.895E5
     0.0
     0.0
     100         0.       150.   5.309E8   5.309E8          .9          .9          .9
     200         3.
  1
951
       0        0.0       0.0
END OF DATA
```

DEVELOPMENT OF A COMPUTER-AIDED GRAPHICS CAPABILITY

TO DISPLAY FIELD TEST DATA

L. Allen Adams

Department of Civil Engineering

M.S. Degree, August 1983

ABSTRACT

This thesis develops an interactive computer-aided graphics package for the display of experimental and field test data. EXPERT was developed to be used in conjunction with MOVIE.BYU and DISSPLA, utilizing a bivariate interpolation scheme to create a full field of data using the "sparse" field data as pivotal points.

The usefulness and practicality of the program is demonstrated by graphically displaying the data from the Air Force Weapons Laboratory's CHEAT test.

COMMITTEE APPROVAL:

Steven E. Benzley
Committee Chairman

Michael B. Stephenson
Committee Member

Henry N. Christiansen
Department Chairman