research report

WYLE LABORATORIES
TESTING DIVISION, HUNTSVILLE FACILITY

FACILITY FORM 602

N70-10328	
(ACCESSION NUMBER)	(THRU)
42	1
(PAGES)	(CODE)
	32
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

WYLE LABORATORIES - RESEARCH STAFF

REPORT WR 69-10

A MULTIVARIATE INTERPOLATION TECHNIQUE TO PRODUCE THREE-DIMENSIONAL SURFACE PLOTS OF VIBRATING STRUCTURES

By

D. M. Lister

Work Performed Under Contract No. NAS8-21260

May 1969

WYLE LABORATORIES
RESEARCH DIVISION, HUNTSVILLE FACILITY

COPY NO. 16

SUMMARY

A method of multivariate interpolation for the production of a smooth surface which passes exactly through a set of given points in three-dimensional space is discussed. Two methods of projecting this surface onto a plane are developed. A Fortran computer program which employs the above techniques to produce three-dimensional plots of the interpolated surface is described. The utilization of this program to produce mode plots of vibrating structures is indicated.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF SYMBOLS

$\vec{P}_{ij}$	A vector defining an input data point (x_{ij}, y_{ij}, z_{ij})
$P(u,v)$	The parametric form for the ordinates ofthe surface segments $\mathcal{S}_{ij}$
R_{pq}	Coefficients of u and v in $P(u,v)$. (Note that the interpolation technique revolves around the computation of the various sets of R_{pq}.)
$\vec{S}_{ij}$	The tangent to the surface at $\vec{P}_{ij}$ with respect to v
$\mathcal{S}_{ij}$	The surface segment fitted to the input data points $\vec{P}_{ij}$, $\vec{P}_{ij+1}$, $\vec{P}_{i+1,j}$ and $\vec{P}_{i+1,j+1}$
$\vec{T}_{ij}$	The tangent to the surface at $\vec{P}_{ij}$ with respect to u
u	Where $0 \leq u \leq 1$, parametric variable used to define the surface segment $\mathcal{S}_{ij}$. Note that u = 0 along $\vec{P}_{ij}\ \vec{P}_{ij+1}$ and u = 1 along $\vec{P}_{i+1j}\ \vec{P}_{i+1\ j+1}$
v	Where $0 \leq v \leq 1$, parametric variable used to define the surface segment $\mathcal{S}_{ij}$. Note that v = 0 along $\vec{P}_{ij}\ \vec{P}_{i+1j}$ and v = 1 along $\vec{P}_{ij+1}\ \vec{P}_{i+1\ j+1}$
X	The x coordinate of the projected point $\vec{P}_{ij}$
x_{ij} or x	The x coordinate of the input data point $\vec{P}_{ij}$
Y	The y coordinate of the projected point $\vec{P}_{ij}$
y_{ij} or y	The y coordinate of the input data point $\vec{P}_{ij}$
z_{ij} or z	The z coordinate of the input data point $\vec{P}_{ij}$

1.0 INTRODUCTION

The value of a three-dimensional representation of the distorted shape of a structure subjected to random or sinusoidal vibrations has been recognized for some time. Normal mode theory provides a convenient method of expressing the vibration of a dynamic system and the determination of the low order normal modes of a complex system is best performed by a surface interpolation of the measured response of the system. In order to obtain such a surface, the coordinates in x,y,z space of a set of points on the distorted structure must be obtained, such that the set form a reasonable grid (not necessarily rectangular) over the whole surface of the structure. An acceptable model of the distorted shape of the structure can be obtained by employing a multivariate interpolation technique. Transformation of this model to a two-dimensional set of axes facilitates the plotting of the surface on an X-Y plotter.

1.1 The Input Data

If the structure under consideration is a flat plate, then the set of points on the distorted structure may be fairly easily obtained. For convenience the original axes are chosen such that the stationary plate is contained entirely in the x-y plane. The displacements of the distorted plate are measured by transducers whose locations form the grid of points. Thus the x and y coordinates of the required set of points are merely the x and y coordinates of the transducers attached to the stationary plate, and the z coordinates are the displacements measured by the transducers at resonance. A detailed account of the application of this plotting technique will be published in a separate report. Similar techniques may be employed for cylinders and spheres except that the positions of the transducers are best expressed in cylindrical and spherical coordinates, respectively, and the displacements measured by the transducers must be added to the r coordinates in both cases provided that the axis of the cylinder coincides with the z-axis, and in case of the sphere, the origin of coordinates is at the centre of the sphere.

1.2 Multivariate Interpolation

The multivariate interpolation technique employed divides the surface into four-sided segments such that each corner of each segment is defined by a point on the input grid. Cubic surfaces are then fitted to each segment such that there is continuity of first and second derivatives between adjacent segments.

1.3 Projection of the Surface Onto a Plane

Two methods of projection are used. The first method projects the surface isometrically onto a predetermined plane. The second method projects the surface perspectively onto a predetermined plane.

1.4 The Computer Program

A Fortran computer program has been written which employs the techniques outlined above to produce three-dimensional plots of interpolated surfaces. The plots are in the form of a series of curves which outline the interpolated surface (see Figure 4). It is of interest to note that the projections of these curves onto the x-y plane of the three-dimensional set of axes form a series of straight lines.

2.0 MULTIVARIATE INTERPOLATION TECHNIQUE

Let $\vec{P}_{ij}$ be an array of m,n distinct points in space, $i = 0, 1, 2, \ldots (m-1)$ $j = 0, 1, 2, \ldots (n-1)$, arranged so that the structure obtained by connecting adjacent points by straight-line segments is topologically equivalent to an m x n planar rectangular grid (see Figure 1).

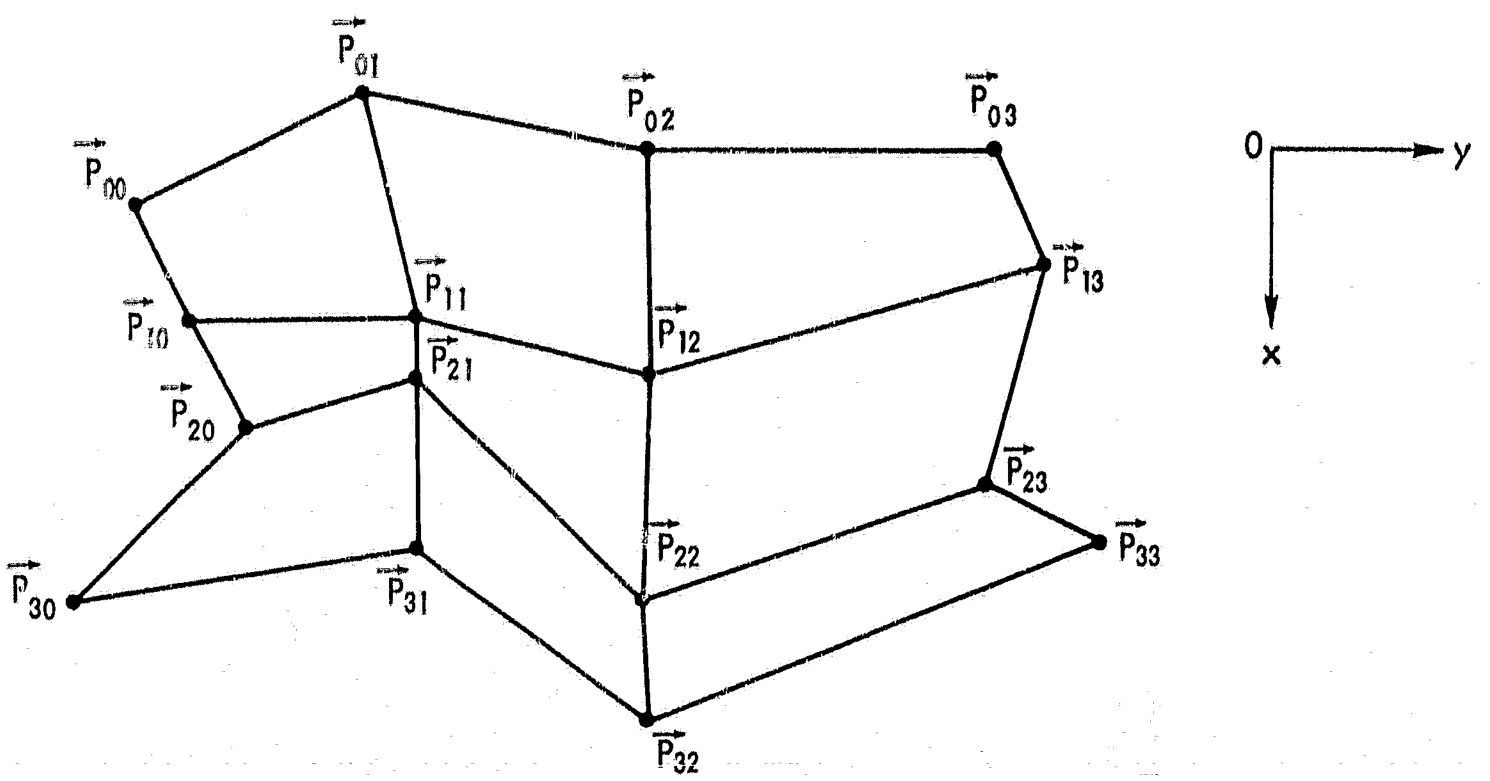

Figure 1 A Typical Grid in the x-y Plane, Formed by Connecting the x,y Coordinates of the Input Data

Then, given a knowledge of the tangents around the boundaries of the grid it is possible to construct a smooth surface which passes exactly through the $\vec{P}_{ij}$ (see Reference 1) points.

The method fits a series of surface segments $\mathcal{S}_{ij}$ of the parametric form

$$P(u,v) = \sum_{p=0}^{3} \sum_{q=0}^{3} u^p v^q R_{pq}; \quad 0 \leq u, v \leq 1 \qquad (1)$$

such that there is continuity of first and second derivatives between adjacent surface segments (see Figure 2)

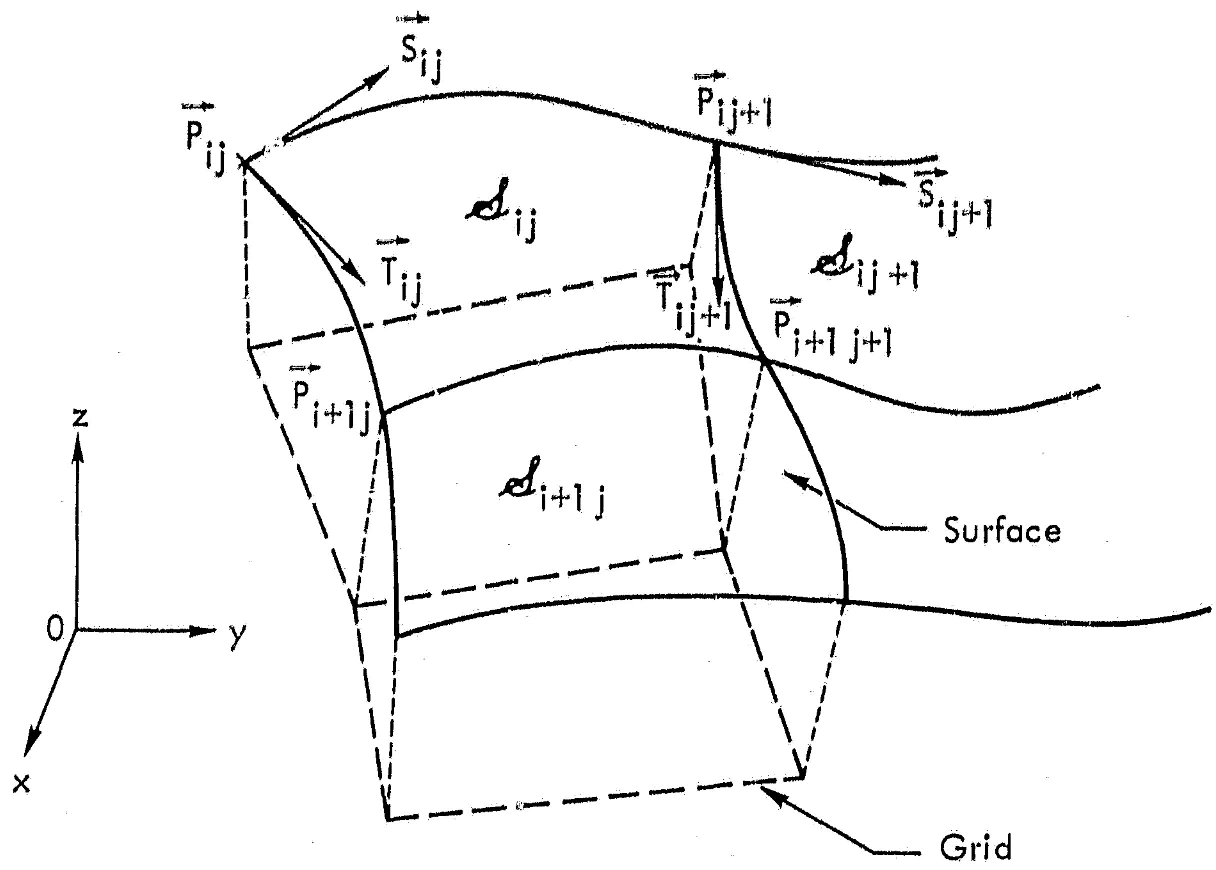

Figure 2. Diagram Showing Typical Surface Segments, Input Data Points P_{ij}, etc.) and Surface Tangents (S_{ij}, T_{ij})

The $\vec{R}_{pq}$ coefficients for any surface segment $\mathcal{S}_{ij}$ are given as functions of the four $\vec{P}_{ij}$'s at the corners of the segment and the eight tangents at these points:

(Let the vector arrows in the following equations be assumed.)

$$R_{00} = P_{ij}$$

$$R_{01} = S_{ij}$$

$$R_{02} = 3\,(P_{ij+1} - P_{ij}) - (2\,S_{ij} + S_{i+1j})$$

$$R_{03} = 2\,(P_{ij} - P_{ij+1}) + (S_{ij} + S_{i+1j})$$

$$R_{10} = T_{ij}$$

$$R_{11} = 0$$

$$R_{12} = 3\,(T_{ij+1} - T_{ij})$$

$$R_{13} = 2\,(T_{ij} = T_{ij+1})$$

$$R_{20} = 3\,(P_{i+1j} - P_{ij}) - (2\,T_{ij} + T_{ij+1})$$

$$R_{21} = 3\,(S_{i+1j} - S_{ij})$$

$$R_{22} = 3\,[3(P_{i+1\,j+1} - P_{ij+1} + P_{ij} - P_{i+1j}) + 2\,(T_{ij} - T_{ij+1}) + (T_{i+1j} - T_{i+1\,j+1}) + 2\,(S_{ij} - S_{i+1j}) + (S_{ij+1} - S_{i+1\,j+1})]$$

$$R_{23} = 2\,[3(P_{i+1j} - P_{ij} + P_{ij+1} - P_{i+1\,j+1}) + 2\,(T_{ij+1} - T_{ij}) + (T_{i+1\,j+1} - T_{i+ij})] + 3(S_{i+1j} + S_{i+1\,j+1} - S_{ij} - S_{ij+1})$$

$$R_{30} = 2\,(P_{ij} - P_{i+1j}) + T_{ij} + T_{ij+1}$$

$$R_{31} = 2\,(S_{ij} - S_{i+1j})$$

$$R_{32} = 3\,[2\,(P_{ij+1} - P_{i+1\,j+1} + P_{i+1j} - P_{ij}) + (T_{ij+1} + T_{i+1\,j+1}) - (T_{ij} + T_{i+1j})] + 4\,(S_{i+1j} - S_{ij}) + 2\,(S_{i+1\,j+1} - S_{ij+1})$$

$$R_{33} = 2\,[2(P_{ij} - P_{i+1j} + P_{i+1\,j+1} - P_{ij+1}) + (T_{ij} + T_{i+1j}) - (T_{ij+1} + T_{i+1\,j+1}) + (S_{ij} - S_{i+1j} + S_{ij+1} - S_{i+1\,j+1})] \; . \qquad (2)$$

The recursive relationships between the tangents at the points $\vec{P}_{ij}$ are:

$$\vec{S}_{ij} + 4\,\vec{S}_{ij+1} + \vec{S}_{ij+2} = 3\,(\vec{P}_{ij+2} - \vec{P}_{ij}) \qquad j = 0, 1, \ldots (m-3)$$

$$\vec{T}_{ij} + 4\,\vec{T}_{i+1j} + \vec{T}_{i+2j} = 3\,(\vec{P}_{i+2j} - \vec{P}_{ij}) \qquad i = 0, 1, \ldots (m-3) \; . \qquad (3)$$

Thus for the grid line implied by P_{2j} (say) it is required that S_{20} and S_{2n-1} be known so that the set of simultaneous linear equations implied by Equation (3) above may be solved.

Note that for the surface segment $\mathcal{S}_{ij}$, $\vec{P}_{ij} \equiv P(0,0)$, $\vec{P}_{i+1j} \equiv P(1,0)$, $\vec{P}_{ij+1} \equiv P(0,1)$, $\vec{P}_{i+1\,j+1} \equiv P(1,1)$.

If some point P on the surface has coordinates (x,y,z) with respect to some predefined set of axes, then:

$$x = X(u,v)$$

$$y = Y(u,v)$$

$$z = P(u,v)$$

where

$$X(u,v) = (1 - v)\,[(1 - u)\,x_{ij} + u\,x_{i+1\,j}] + v\,[(1 - u)\,x_{ij+1} + u\,x_{i+1\,j+1}]$$

$$Y(u,v) = (1 - v)\,[(1 - u)\,y_{ij} + u\,y_{i+1\,j}] + v\,[(1 - u)\,y_{ij+1} + u\,y_{i+1\,j+1}] \quad . \tag{4}$$

A proof of Equations (4) is offered in Appendix A.

3.0 SURFACE PROJECTION

Section 2.0 above shows how to obtain a multivariate interpolated surface with respect to a rectangular set of axes 0,x,y,z. There remains the problem of transforming to a two-dimensional set of axes XOY for the purposes of plotting.

3.1 Isometric Projection

It is assumed that the origins of XOY, 0,x,y,z are coincident and that the point in space p(x,y,z) is transformed to P(X,Y) on the projection plane. It can be shown (see Appendix B) that:

$$X = -x \sin\theta_x \sin(\theta_{zx} + \phi) + y \sin\theta_y \sin(\theta_{zy} - \phi) - z \sin\theta_z \sin\phi$$

$$Y = -x \sin\theta_x \cos(\theta_{zx} + \phi) - y \sin\theta_y \cos(\theta_{zy} - \phi) + z \sin\theta_z \cos\phi$$

where $[\cos\theta_x, \cos\theta_y, \cos\theta_z] \equiv [\ell, m, n]$ are the direction cosines of the normal to the projection plane at the origin.

ϕ is the angle measured anticlockwise from OY to the projection of OZ on the projection plane.

$$\theta_{zy} = \pi - \cos^{-1}\left[\frac{mn}{\sqrt{(\ell^2 + m^2)(\ell^2 + n^2)}}\right]$$

$$\theta_{zx} = \pi - \cos^{-1}\left[\frac{\ell n}{\sqrt{(\ell^2 + m^2)(m^2 + n^2)}}\right] .$$

Thus in order to define the plot, ϕ and two of $[\ell, m, n]$ must be defined. (Note that $\ell^2 + m^2 + n^2 = 1$, because they are direction cosines.)

3.2 Perspective Projection

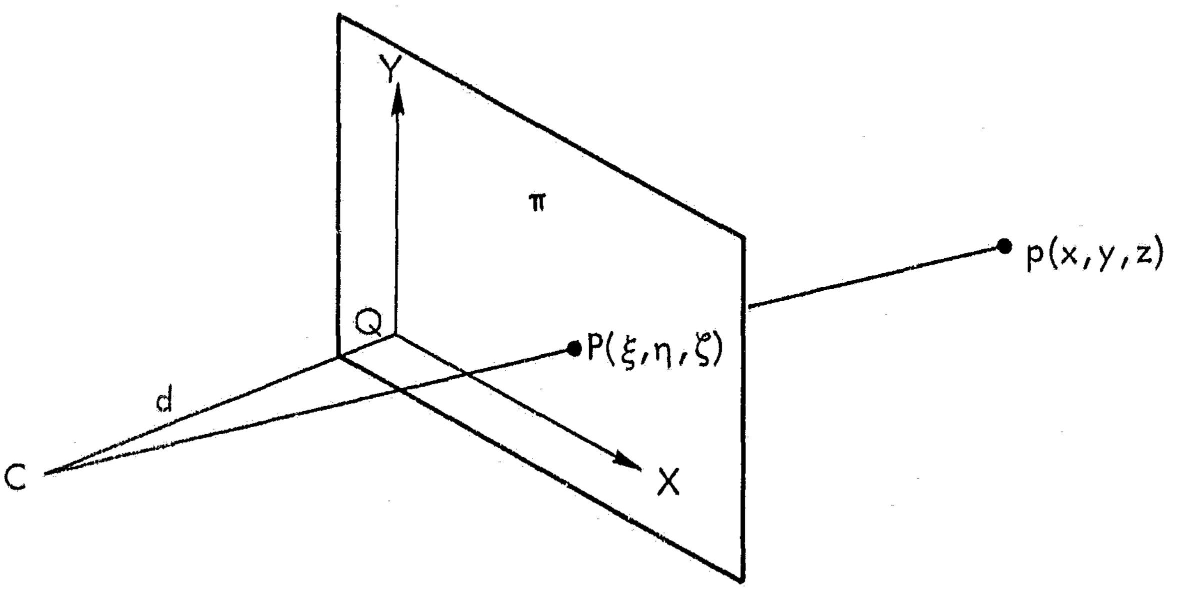

Figure 3. Illustration of the Perspective Projection of the Point p(x,y,z) onto the Plane of Projection (Π)

Given the center of projection (c_x, c_y, c_z), the direction of sight, $[\cos\alpha, \cos\beta, \cos\gamma]$, and the distance d from the center of projection to the origin on the plane of projection (Π) then it can be shown (see Reference 2) that the transformed coordinates P(X,Y) of p(x,y,z) are given by

$$X = [(\xi - q_x)\cos\beta - (\eta - q_y)\cos\alpha]/\sin\gamma$$

$$Y = (\zeta - q_z)/\sin\gamma$$

where

$$\xi = c_x + K(x - c_x)$$

$$\eta = c_y + K(y - c_y)$$

$$\zeta = c_z + K(z - c_z)$$

$$K = d/[(x - c_x)\cos\alpha + (y - c_y)\cos\beta + (z - c_z)\cos\gamma]$$

$$q_x = c_x + d\cos\alpha$$

$$q_y = c_y + d \cos \beta$$

$$q_z = c_z + d \cos \gamma .$$

Thus in order to define the plot, c_x, c_y, c_z, d and two of α, β, γ must be defined. (Note $\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1$ because they are direction cosines.)

It will be noticed that the above transformation depends on $\sin \gamma \neq 0$. It can be shown (Reference 2) that if $\sin \gamma = 0$, the x-axis can be defined as the intersection of of the plane of projection and the vertical plane $\eta - q_y = 0$. If, furthermore, up corresponds to an increasing y-component, positive X is to the right and positive Y is up, then the perspective transformation is:

$$X = [- (\xi - q_x) \cos \gamma + (\zeta - q_z) \cos \alpha] / \sin \beta$$

$$Y = (\eta - q_y) / \sin \beta \; .$$

4.0 THE COMPUTER PROGRAM

A computer program has been prepared in the Fortran language to utilize the methods of Sections 2.0 and 3.0, which yields three-dimensional plots of interpolated surfaces obtained from a given set of input conditions. This program is an extension of an existing program. The new program has four distinct advantages over the old one --

(i) It has a much larger general application.

(ii) Two methods of projection are offered to the user.

(iii) The user has a choice of three coordinate systems for the input data; namely, Cartesian, cylindrical polars, and spherical polars.

(iv) All the two-dimensional arrays involved in the computation are held as one-dimensional arrays so that best use is made of the available amount of data storage space in the computer.

4.1 The Main Program (REVIEW)

The main program performs the following functions:

(i) Inputs and decodes the control parameters. (A more detailed account of the function of these parameters will be given later in this section.)

(ii) Computes those parameters required for the transformation of coordinates and places them in convenient locations in COMMON for use in subroutine TRANS.

(iii) Inputs the data points.

(iv) Uses subroutine ACCEL (supplied by the user) to input further data (e.g., transducer data) and adds the input vector ADDIT to:

(a) Adds to z coordinate for Cartesians

(b) Adds to u coordinate for cylindrical polars

(c) Adds to r coordinate for spherical polars.

(v) Converts the data to Cartesian coordinates.

(vi) Normalizes the data so that the modulii of the maxima of the x, y and z coordinates are each equal to unity. It is worth noting here that the user must be careful in the selection of units for (iii) and (iv) above, or the data may be ruined by the normalization process.

(vii) Sets up the initial boundary conditions, according to the values of parameters ITT, ITS input at stage (i) above. This may involve subroutine CUBIC.

(viii) Fits the interpolated surface. This involves the use of subroutines NEWJAC and MATMULT.

(ix) Transforms the interpolated surface for the purposes of plotting. This utilizes subroutine TRANS.

(x) Plots the three-dimensional surface.

4.1.1 The Control Parameters

ITAX — the value of ITAX determines the coordinate system of the input data:

if ITAX = 1	then Cartesian coordinates
= 2	then cylindrical polars (u,ϕ,z) where $x = u \cos \phi$, $y = u \sin \phi$, $z = z$
= 3	then spherical polars (r,θ,ϕ) where $x = r \sin \theta \cos \phi$, $y = r \sin \theta \sin \phi$, $z = r \cos \theta$
$\neq 1,2,3$	then end of run.

MIP — the number of rows on the input grid

MJP — the number of columns on the input grid (Note: there are (MIP-1) (MJP-1) surface segments).

ITS, ITT — these parameters define the initial boundary conditions for the S and T tangents, respectively.

= 1	then the boundary tangents are input from cards. Note that 2x MIP S's and 2x MJP T's are required.

= 3 then a cubic is fitted through four data points (see example below) and the required tangent estimated from the cubic. For example, for S_{3n-1} the points P_{3n-1}, P_{3n-2}, P_{3n-3} and P_{3n-4} are used.

$\neq$ 1,2,3 then end of run.

IPER this parameter determines which form of projection is to be used

if IPER ≤ 0 then the normal projection as described in Section 3.1 is used.

> 0 then the perspective transformation as described in Section 3.2 is used.

IUSTOP this parameter determines how many contour lines per surface segment are to be plotted. That is, u will vary between zero and unity in steps of USTEP where USTEP = 1/(IUSTOP-1).

IVSTOP this parameter determines how many points per contour line are to be plotted. That is, v will vary between zero and unity in steps of VSTEP where VSTEP = 1/(IVSTOP - 1).

4.2 Subroutine ACCEL (X,M)

This subroutine must be supplied by the user. On exit the array X must be filled with supplementary data (e.g., transducer data). M is the number of input data points.

The main program will add the array X to the input data according to the conditions described in Section 4.1(iv).

4.3 Subroutine MATMULT (A,P,S,M,MJP)

This is a matrix multiplication routine. The matrix $\underline{A}$ is multiplied by the vector $\underline{P}$ to give the tangent vector $\underline{S}$. M is the number of tangents to be computed and MJP the number of columns of input data.

Entry MULTMAT deals with the case where the T tangents are required.

4.4 Subroutine CUBIC (Y,X,T,M,N)

This subroutine fits the model

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3$$

by the method of least squares, where y and x are assumed to be arrays of four elements each. If N = 1, then the tangent at x = X(1) is computed and placed in T(1), i.e.,

$$T(1) = b_1 + b_2 X(1) + b_3 [X(1)]^2$$

If N = 2, then the tangent at x = X(4) is computed. Entry RECUBE deals with the T tangents and CUBIC with the S tangents. M is the number of columns of input data.

Note N = 1 for the computation of tangents along the "top" and "left hand side" of the grid and N = 2 for the opposite boundaries.

4.5 Subroutine NEWJAC (NX,N,A)

This subroutine inverts the matrix A. NX is the number of columns nominated in the deceleration statement for A in the calling routine and N is the actual number of rows (and columns) in A.

4.6 Subroutine TRANS (XP,YP,X,Y,Z,IPER)

This subroutine transforms the three-dimensional coordinates (X,Y,Z) to the two-dimensional set (XP,YP). If IPER > 0 then the transformation is as described in Section 3.2 (i.e., perspective). If IPER ≤ 0 then the transformation is as described in Section 3.1 (i.e., normal).

4.7 Input to the Program

This is best illustrated in tabular form on the following page.

4.8 Output From the Program

The output from the program consists merely of a three-dimensional plot.

Card Type	Symbol	Nmemonic	Columns	Format	Description
1	ITAX	ITAX	1-5	I5	This parameter defines the coordinate system of the input data. See Section 4.1.1.
1	M	MIP	6-10	I5	The number of rows of input data.
1	N	MJP	11-15	I5	The number of columns of input data.
1	ITS	ITS	16-20	I5	These parameters define the tangential boundary conditions for S and T, respectively. See Section 4.1.1.
1	ITT	ITT	21-25	I5	
1	IPER	IPER	26-30	I5	This parameter defines the type of plot transformation to be used. See Section 4.1.1.
1	IUSTOP	IUSTOP	31-35	I5	The number of contour lines to be plotted per surface segment.
1	IVSTOP	IVSTOP	36-40	I5	The number of points to be plotted per contour line.
*2	ϕ	PHI	1-10	F10.0	The angle ϕ in degrees. See Section 3.1.
2	θ_x	THX	11-20	F10.0	The angle θ_x in degrees, where $\cos\theta_x = \ell$. See Section 3.1.
2	θ_y	THY	21-30	F10.0	The angle θ_y in degrees, where $\cos\theta_y = m$. See Section 3.1.
2	θ_z	THZ	31-40	F10.0	The angle θ_z in degrees, where $\cos\theta_z = n$. See Section 3.1.
2	TI	TI	41-50	F10.0	TI = 1,2 or 3 and indicates which of θ_x, θ_y, θ_z above is redundant, as only two of them need to be defined. See Section 3.1.
**2	d	D	1-10	F10.0	The distance from the center of projection to the plane of projection. (Units as per input data.) See Section 3.2.

Card Type	Symbol	Nmemonic	Columns	Format	Description
2 2 2	c_x c_y c_z	CX CY CZ	11-20 21-30 31-40	F10.0 F10.0 F10.0	The coordinates of the center of projection. See Section 3.2. Units as for D above
2 2 2	α β γ	THX THY THZ	41-50 51-60 61-70	F10.0 F10.0 F10.0	The angles in degrees, of the direction of sight. See Section 3.2.
2	TI	TI	71-80	F10.0	TI = 1,2 or 3 and indicates which of α, β, γ above is redundant as only two of them need to be defined. See Section 3.2.
***3+	x y z	X(L) Y(L) Z(L)	1-10 11-20 21-30	F10.0 F10.0 F10.0	The coordinates of one input data point. See Section 2.0.
+4+	S_{ij}	S(I)	1-80	8F10.0	The boundary values of the S tangents. Here the tangents S_{11} through S_{m1} are read up to eight per card.
+4+	S_{ij}	S(I)	1-80	8F10.0	The boundary values of the S tangents. Here the tangents S_{1n} through S_{mn} are read up to eight per card.
++4+	T_{ij}	T(I)	1-80	8F10.0	The boundary values of the T tangents. Here the tangents T_{11} through T_{1n} are read up to eight per card.
++4+	T_{ij}	T(I)	1-80	8F10.0	The boundary values of the T tangents. Here the tangents T_{m1} through T_{mn} are read up to eight per card.

* This card type 2 required if IPER $\leq$ 0.
** This card type 2 required if IPER > 0.
*** There will be MIP x MJP type 3 cards. Note also that following the type 3 cards extra data cards may be required for subroutine ACCEL (See Section 4.2).

+ There will be MIP S tangents, if required at all.
++ There will be MJP T tangents, if required at all.

(Note that the values of these tangents must be adjusted to allow for the normalization of the input data described in Section 4.1(vi).)

5.0 APPLICATION OF THE PROGRAM

Outlined in the following subsections are some important aspects of using the program.

5.1 Presentation of Input Data

Due consideration should be given to the best way to present the input data so that the required information is best illustrated in the resulting plot. For instance, the structure involved in Figure 4 is basically a cylinder. Cartesian coordinates with the x-axis along the axis of the cylinder were obviously used to express the input data. Had cylindrical polar coordinates with the z-axis along the axis of the cylinder been used, then the surface curves in the plot would have been approximately at right angles to those shown in Figure 4.

5.2 Choice of the Plane of Projection

The choice of the plane of projection determines the view that the user will obtain of the interpolated surface. In practice it may be desirable to repeat the plot several times, each with a different plane of projection, so that an optimum overall view of a structural vibration mode is obtained. It is of importance, however, to note that a "poor" choice of the plane of projection may result in the plot being distorted beyond recognition.

5.3 Tangential Boundary Conditions

It is important to note that the tangents S and T are $\partial P(u,v)/\partial v$ and $\partial P(u,v)/\partial u$, respectively, (or $\partial z/\partial v$, $\partial z/\partial u$, respectively). Only when a rectangular grid is used with the grid lines parallel to the x and y axes, are these tangents equivalent to $\partial z/\partial y$ and $\partial z/\partial x$, respectively. Subroutine CUBIC should only be used to estimate tangential boundary conditions when the user has no knowledge of these values. It must also be noted that subroutine CUBIC provides better estimates of the tangential boundary conditions when a rectangular grid with the grid lines parallel to the x and y axes is employed.

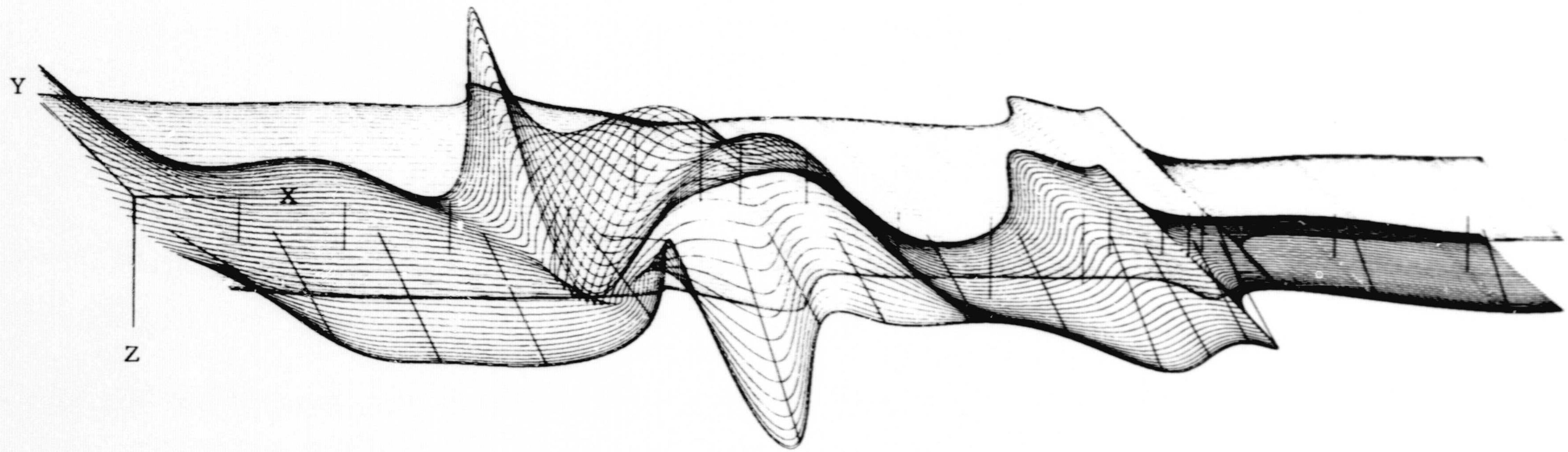

Figure 4. Surface Plot of Cylindrical Structure at its 16 Hz Resonant Mode

6.0 CONCLUSIONS

The techniques and computer program described in this report are very powerful engineering tools, particularly when used to provide plots of mode shapes of structures vibrating at resonance. The options offered the user in methods of presentation of data, and choices of methods of, and planes of projection, provide versatile means of ensuring that the required information is best illustrated in the plots produced. A subsequent report will provide illustrations of the application of the program to provide modal plots of vibrating structures subjected to sinusoidal and random excitations.

6.0 CONCLUSIONS

The techniques and computer program described in this report are very powerful engineering tools, particularly when used to provide plots of mode shapes of structures vibrating at resonance. The options offered the user in methods of presentation of data, and choices of methods of, and planes of projection, provide versatile means of ensuring that the required information is best illustrated in the plots produced. A subsequent report will provide illustrations of the application of the program to provide modal plots of vibrating structures subjected to sinusoidal and random excitations.

REFERENCES

1. Ferguson, J., "Multivariate Curve Interpolation," Journal of the Association of Computing Machinery, Vol. 11, No. 2 (April 1964), pp. 221-228.

2. Kubert, B., Szabo, J. and Giulieri, S., "The Perspective Representation of Functions of Two Variables," The Journal of the Association of Computing Machinery, Vol. 15, No. 2 (April 1968), pp. 193-204.

3. Lowry, H. V. and Hayden, H. A., "Advanced Mathematics for Technical Students," Part II, Longmans, Green and Co., pp. 178.

APPENDIX A

COMPUTATION OF THE x AND y COORDINATES OF A POINT P(u, v) ON THE INTERPOLATED SURFACE

APPENDIX A

COMPUTATION OF THE x AND y COORDINATES OF A POINT P(u,v) ON THE INTERPOLATED SURFACE

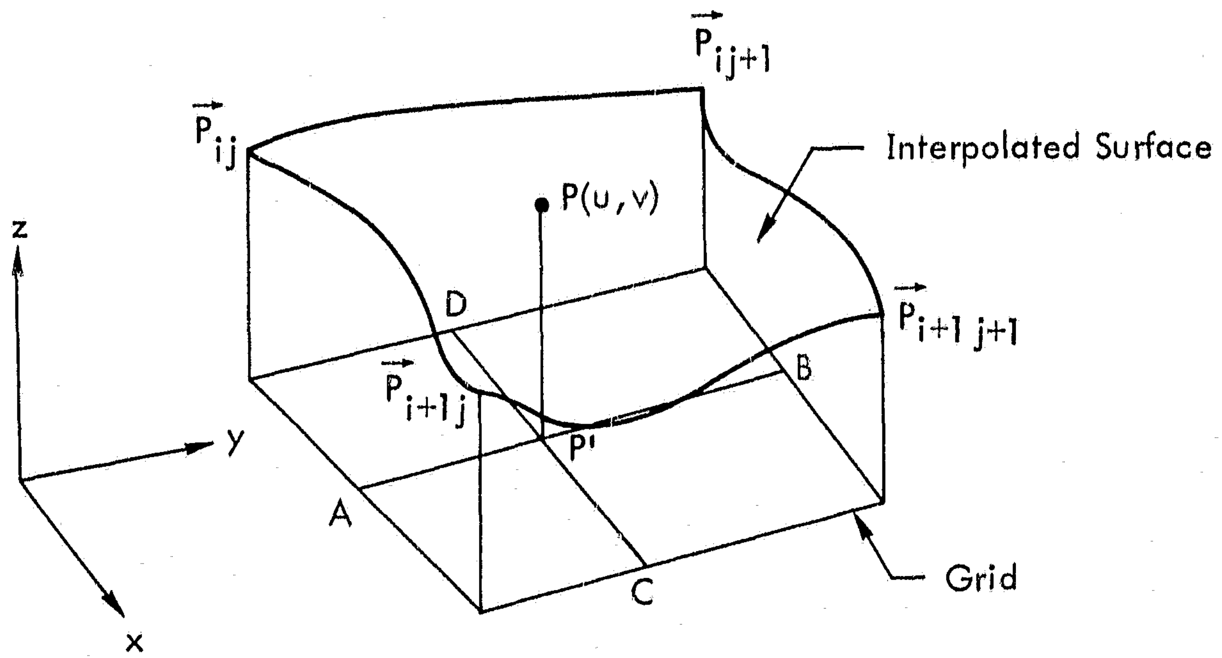

Figure A-1. Illustration of Interpolated Surface Showing the Projection of a Point P Onto the Plane of the Grid

Let the coordinates of the points $\vec{P}_{ij}$, $\vec{P}_{i+1j}$ ... $\vec{P}_{i+1\,j+1}$ be (x_{ij}, y_{ij}, z_{ij}), etc.

Now $\vec{P}_{ij} \equiv P(0,0)$, $\vec{P}_{ij+1} \equiv P(0,1)$, $\vec{P}_{i+1j} \equiv P(1,0)$, $\vec{P}_{i+1\,j+1} \doteq P(1,1)$. Consider some point $P(u,v) \equiv (x,y,z)$ on the interpolated surface. Its x,y coordinates are given by the position of P' on the x-y plane (see Figure A-1).

Let the coordinates of A, B be $(x_a, y_a, 0)$, $(x_b, y_b, 0)$, then

$$\frac{x_a - x_{ij}}{u} = \frac{x_{i+1j} - x_{ij}}{1}$$

$$\therefore \quad x_a = x_{ij}(1-u) + u\,x_{i+1\,j} \quad .$$

Similarly,

$$x_b = x_{ij+1}(1-u) + u\,x_{i+1\,j+1}$$

but

$$\frac{x - x_a}{v} = \frac{x_b - x_a}{1}$$

$$\therefore \quad x = x_a(1-v) + v\,x_b$$

$$\underline{x = (1-v)\,[x_{ij}(1-u) + u\,x_{i+1\,j}] + v\,[x_{ij+1}(1-u) + u\,x_{i+1\,j+1}]} \quad .$$

The argument to find y is similar to the above, hence

$$\underline{y = (1-v)\,[y_{ij}(1-u) + u\,y_{i+1\,j}] + v\,[y_{ij+1}(1-u) + u\,y_{i+1\,j+1}]} \quad .$$

APPENDIX B

ISOMETRIC SURFACE PROJECTION

APPENDIX B

ISOMETRIC SURFACE PROJECTION

Given the surface $z = f(x,y)$ the problem is to project this surface onto a plane, which shall be called the "plot" plane or plane of projection, such that the projected surface may be plotted by a computer on an X-Y plotter. Thus it is required that a new set of axes XOY be defined in the plane of projection to facilitate the plot. For convenience let it be assumed that the plane of projection passes through the origin of the 0,x,y,z coordinate system; the normal to this plane has direction cosines $[\ell, m, n]$, where $\ell = \cos\theta_x$, $m = \cos\theta_y$, $n = \cos\theta_z$; the Y axis of the new coordinate system is at an angle ϕ measured anticlockwise from 0Y to the projection of 0z on the plane of projection; the angle between the projections of 0z and 0y on the plane of projection is θ_{zy} and the angle between the projections of 0z and 0x on the plane of projection is θ_{zx} (see Figures B-1 and B-2). Thus to define the plot, ϕ, and two of $[\ell, m, n]$ must be defined. (Note that $\ell^2 + m^2 + n^2 = 1$ - a property of direction cosines.)

Let the projections of 0x, 0y, 0z onto the plane of projection be 0x', 0y', 0z'. Then,

$$x' = x \sin\theta_x \tag{B-1}$$

$$y' = y \sin\theta_y \tag{B-2}$$

$$z' = z \sin\theta_z \quad . \tag{B-3}$$

It is required to find the coordinates of the points x', y', z' with respect to XOY, and thus θ_{zx}, θ_{zy} must be found.

Let the equation of the plane containing the normal $[\ell, m, n]$ and the z-axis be given by:

$$A_1 x + B_1 y + C_1 x + D_1 = 0 \quad .$$

As the plane passes through the origin, $D_1 = 0$. As the plane contains the z axis, $C_1 = 0$. As the plane contains $[\ell, m, n]$, then

$$A_1 \ell + B_1 m = 0$$

Hence,

$$A_1 = - B_1 m/\ell$$

$$\therefore B_1 \left[- \frac{mx}{\ell} + y \right] = 0 \quad .$$

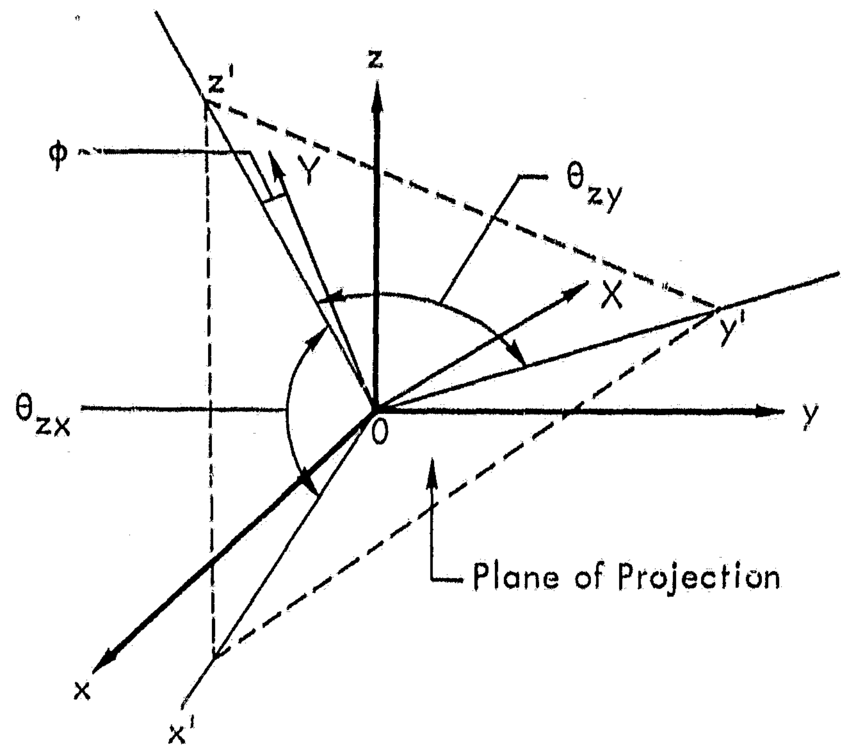

Figure B-1. Diagram Showing the Relationship Between the Original Axes and the Plotting Axes for the Normal Projection

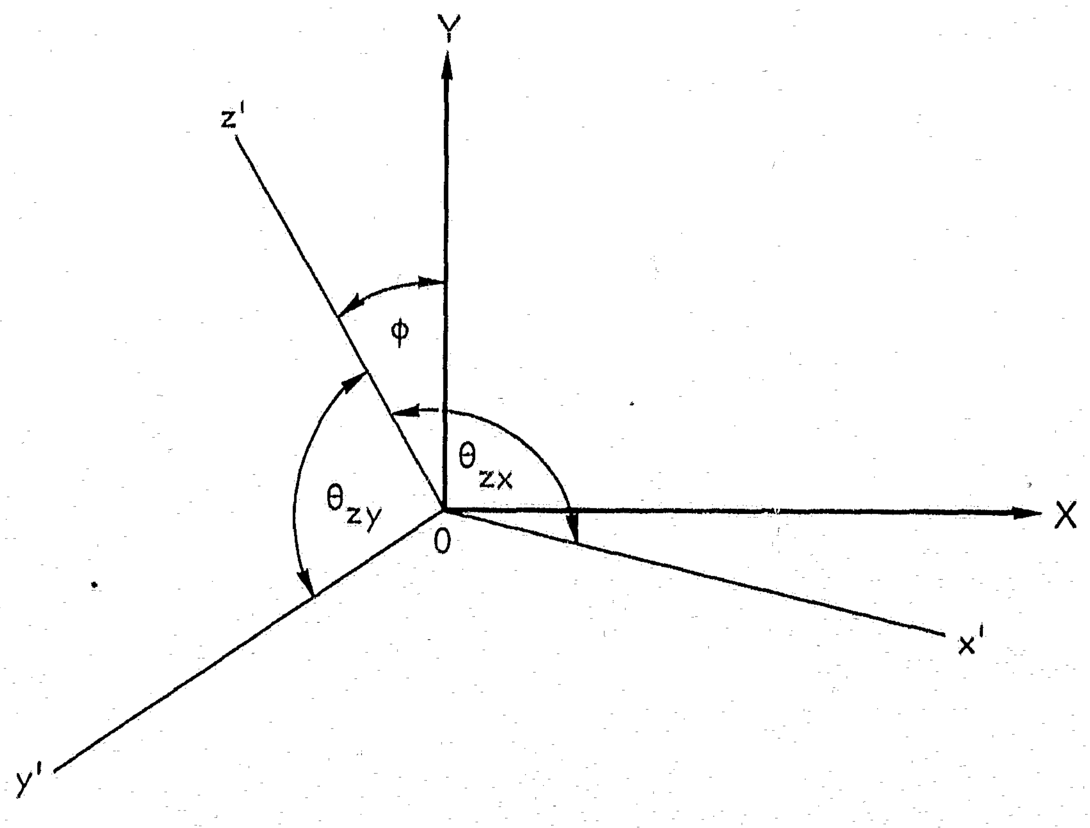

Figure B-2. Diagram Showing the Angular Relationship Between the Projections of the Original Axes on the Normal Plane of Projection and the Plotting Axes

Thus, the equation of the plane is given by

$$- m x + \ell y = 0 \tag{B-4}$$

Similarly, it can be shown that the plane containing the normal $[\ell, m, n]$ and the x-axis is given by

$$ny - mz = 0 \tag{B-5}$$

Obviously, the two planes mentioned above are normal to the plane of projection, contain Oz' and Oz', respectively, and the angle between them is θ_{zx}.

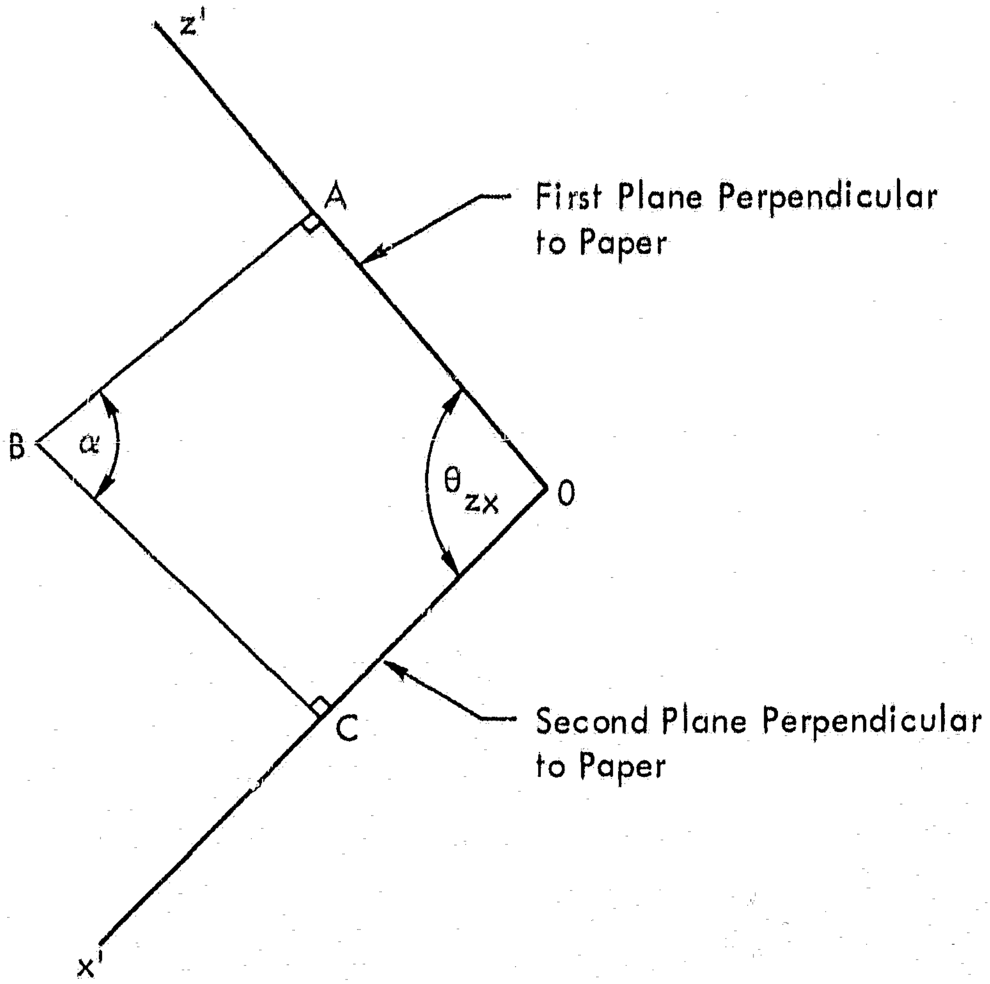

Figure B-3. Diagram Showing the Angular Relationships Between the Projections of Oz and Ox on the Plot Plane and the Normals to These Projections

If AB, CB are normals to the two planes and intersect at an angle α (see Figure B3) then

$$\theta_{zx} = \pi - \alpha \tag{B-6}$$

because 0ABC is a cyclic quadrilateral in which opposite angles sum to π.

Now the direction cosines of the normal to the plane

$$Ax + By + Cz + D = 0$$

are given by

$$[\ell, m, n] = \left[\frac{A, B, C}{\sqrt{A^2 + B^2 + C^2}}\right] .$$

Thus the direction cosines of the normals to the planes given by (B-4) and (B-5) are

$$[\ell_1, m_1, n_1] = \left[\frac{-m, \ell, 0}{\sqrt{m^2 + \ell^2}}\right] \tag{B-7}$$

and

$$[\ell_2, m_2, n_2] = \left[\frac{0, n, -m}{\sqrt{n^2 + m^2}}\right]$$

Now the angle θ between two rays $[\ell, m, n]$, $[\ell', m', n']$ is given by (Reference 3)

$$\cos\theta = \ell\,\ell' + m\,m' + n\,n'$$

and thus

$$\cos\alpha = \frac{\ell n}{\sqrt{(\ell^2 + m^2)(n^2 + m^2)}} . \tag{B-8}$$

Thus by solving (B-8) and substituting in (B-6), θ_{zx} can be found

$$\theta_{zx} = \pi - \cos^{-1}\left[\frac{\ell n}{\sqrt{(\ell^2 + m^2)(n^2 + m^2)}}\right] . \tag{B-9}$$

By a similar argument it can be shown that

$$\theta_{zy} = \pi - \cos^{-1}\left[\frac{mn}{\sqrt{(\ell^2 + m^2)(\ell^2 + n^2)}}\right] \quad . \qquad \text{(B-10)}$$

Thus the (X, Y) coordinates of a projected point p(x,y,z) are given by

$$X = -x \sin\theta_x \cos\left(\theta_{zx} + \phi - \frac{\pi}{2}\right) - x \sin\theta_z \cos\left(\frac{\pi}{2} - \phi\right) + y^2 \sin\theta_y \cos\left(\theta_{zy} - \phi - \frac{\pi}{2}\right)$$

$$= \underline{-x \sin\theta_x \sin(\theta_{zx} + \phi) + y \sin\theta_y \sin(\theta_{zy} - \phi) - z \sin\theta_z \sin\phi}$$

$$Y = -x \sin\theta_x \cos(\pi - \theta_{zx} - \phi) - y \sin\theta_y \cos(\pi - \theta_{zy} + \phi) + z \sin\theta_z \cos\phi$$

$$= \underline{-x \sin\theta_x \cos(\theta_{zx} + \phi) - y \sin\theta_y \cos(\theta_{zy} - \phi) + z \sin\theta_z \cos\phi} \quad .$$

APPENDIX C

LISTING OF COMPUTER PROGRAM REVIEW

```
      PROGRAM REVIEW
      COMMON S1,S2,S3,C1,C2,C3,D,CX,CY,CZ,DCL,DCM,DCN,QX,QY,QZ,SL,SM,SN
      COMMON Z(250),X(250),Y(250),T(250),R(250),A(25,50),P(25)
      COMMON/DATA/PI,CON
      DATA(PI=3.1415927)
      DATA(CON=180.0)
      DIMENSION ADDIT(250),C(4,4),YM(4),XM(4)
1     READ( ,2)ITAX,MIP,MJP,ITS,ITT,IPER,IUSTOP,IVSTOP
2     FORMAT (16I5)
      MIJP=MIP*MJP
      U=FLOAT(IUSTOP-1)
      USTEP=1.0/U
      V=FLOAT(IVSTOP-1)
      VSTEP=1.0/V
      IF(IPER) 100,100,10
100   READ( ,7)PHI,THX,THY,THZ,TI
      IT=IFIX(TI)
      PHI=PHI*PI/CON
      THX=THX*PI/CON
      THY=THY*PI/CON
      THZ=THZ*PI/CON
      SL=SIN(THX)
      SM=SIN(THY)
      SN=SIN(THZ)
      DCL=COSF(THX)
      DCM=COSF(THY)
      DCN=COSF(THZ)
      GO TO (101,102,103),IT
101   DCL=SQRT(1.0-DCM*DCM-DCN*DCN)
      SL=SQRT(1.0-DCL*DCL)
      GO TO 104
102   DCM=SQRT(1.0-DCN*DCN-DCL*DCL)
      SM=SQRT(1.0-DCM*DCM)
      GO TO 104
103   DCN=SQRT(1.0-DCL*DCL-DCM*DCM)
      SN=SQRT(1.0-DCN*DCN)
104   ARG1=DCL*DCN/SQRT((DCL*DCL+DCM*DCM)*(DCM*DCM+DCN*DCN))
      ARG1=SQRT(1.0-ARG1*ARG1)/ARG1
      ALF=ATAN(ARG1)
      IF(ALF)105,106,106
105   ALF=ALF+PI
106   THZX=PI-ALF
      ARG1=DCM*DCN/SQRT((DCL*DCL+DCM*DCM)*(DCL*DCL+DCN*DCN))
      ARG1=SQRT(1.0-ARG1*ARG1)/ARG1
      ALF=ATAN(ARG1)
      IF(ALF)107,108,108
107   ALF=ALF+PI
```

```
108   THZY=PI-ALF
      ARG1=THZX+PHI
      S1=SIN(ARG1)
      C1=COS(ARG1)
      ARG1=THZY-PHI
      S2=SIN(ARG1)
      C2=COS(ARG1)
      S3=SI (PHI)
      C3=CO (PHI)
      GO TO 115
109   READ( 0,7) D,CX,CY,CZ,THX,THY,THZ,TI
      IT=IFIX(TI)
      THX=THX*PI/CO
      THY=THY*PI/CO
      THZ=THZ*PI/CO
      DCL=COS(THX)
      DCM=COS(THY)
      DCN=COS(THZ)
      GO TO (110,111,112), IT
110   DCL=SQRT(1.0-DCM*DCM-DCN*DCN)
      GO TO 114
111   DCM=SQRT(1.0-DCN*DCN-DCL*DCL)
      GO TO 114
112   DCN=SQRT(1.0-DCL*DCL-DCM*DCM)
114   OX=CX+D*DCL
      OY=CY+D*DCM
      OZ=CZ+D*DCN
      S3=SQRT(1.0-DCN*DCN)
      S2=SQRT(1.0-DCM*DCM)
115   CONTINUE
      IF(IT Y)3,3,5
3     WRITE(61,4)
4     FORMAT (9X,11HEND OF JOB.)
      TOP
5     IF(IT Y-4)6,3,3
6     DO 8 I=1,MIP
      K=(I- )*MJP
      DO   J=1,MJP
      L=K+J
      READ (60,7) X(L),Y(L),Z(L)
7     FORMAT (8F10.0)
      (L)=0.0
8     T(L)=0.0
      CALL ACCEL (ADDIT,MIJP)
      DO 13 I=1,MIP
      K=(I-1)*MJP
      DO 13 J=1,MJP
      L=K+J
      GO TO (12,9,9), ITAX
9     X(L)=X(L)+ADDIT(L)
      GO TO (13,10,11) ITAX
10    Y(L)=X(L)*SINF(Y(L))
```

```
      X(L)=X(L)*COSF(Y(L))
      GO TO 12
11    X1=X(L)*SIN(Y(L))
      Y1=X(L)*COS(Y(L))
      X(L)=X1*COS(Z(L))
      Y(L)=X1*SIN(Z(L))
      Z(L)=Y1
      GO TO 13
12    Z(L)=Z(L)+ADDIT(L)
13    CONTINUE
      XMAX=-1.0E200
      ZMAX=XMAX
      YMAX=ZMAX
      DO 605 I=1,MIJP
      IF(ABS(X(I))-XMAX)601,601,600
600   XMAX=ABS(X(I))
601   IF(ABS(Y(I))-YMAX)603,603,602
602   YMAX=ABS(Y(I))
603   IF(ABS(Z(I))-ZMAX)605,605,604
604   ZMAX=ABS(Z(I))
605   CONTINUE
      DO 606 I=1,MIJP
      X(I)=X(I)/XMAX
      Y(I)=Y(I)/YMAX
      Z(I)=Z(I)/ZMAX
606   CONTINUE
      IF (ITS) 14,14,16
14    WRITE(61,15)
15    FORMAT (10X,27HERROR IN TANGENT PARAMETER.)
      GO TO 3
16    IF(ITS-4)17,14,14
17    GO TO (25,18,19), ITS
18    IF=MJP*(MIP-1)+1
      READ(60,7)(S(I),I=1,IE,MJP)
      IE=MIP*MJP
      READ(60,7)(S(I),I=MJP,IE,MJP)
      GO TO 25
19    IF(MJP-6)20,22,22
20    WRITE(61,21)
21    FORMAT (10X,48HTANGENT BOUNDARY CONDITIONS CANNOT BE SATISFIED.)
      GO TO 3
22    IE=MJP*(MIP-1)+1
      DO 23 I=1,IE,MJP
      CALL CUBIC(Z(I),Y(I),S(I),MJP,1)
23    CONTINUE
      IE=MIP*MJP
      DO 24 I=MJP,IE,MJP
      K=I-4
      CALL CUBIC (Z(K),Y(K),S(I),MJP,2)
24    CONTINUE
25    IF(ITT)14,14,26
26    IF(ITT-4)27,14,14
```

```
27    GO TO (33,28,29), ITT
28    READ( ,7)(T(I),I=1,MJP)
      IS=(MIP-1)*MJP
      IE=IS+MJP
      IS=IS+1
      READ( ,7)(T(I),I=IS,IE)
      GO TO 33
29    IF(MIP-6)20,30,30
30    DO 31 I=1,MJP
      CALL RECUBE(Z(I),X(I),T(I),MJP,1)
31    CONTINUE
      IS=(MIP-1)*MJP
      IE=IS+MJP
      IS=IS+1
      DO 32 I=IS,IE
      K=I-4*MJP
      CALL RECUBE (Z(K),X(K),T(I),MJP,2)
32    CONTINUE
33    MJJ=MJP-2
      DO 34 I=1,MJP
      DO 34 J=1,MJP
      A(I,J)=0.0
34    CONTINUE
      DO 36 I=1,MJJ
      A(I,I)=4.0
      IF(I-MJJ)35,36,36
35    A(I+1,I)=A(I,I+1)=1.0
36    CONTINUE
      CALL NEWJAC(25,MJJ,A)
      DO 38 I=1,MIP
      K=(I-1)*MJP
      DO 37 J=1,MJJ
      L1=K+J
      L2=L1+2
      P(J)=(Z(L2)-Z(L1))*3.0
37    CONTINUE
      K1=K+1
      K2=K1+1
      P(1)=P(1)-S(K1)
      K1=K+MJP
      P(MJJ)=P(MJJ)-S(K1)
      CALL MATMULT(AA,P,S(K2),MJJ,MJP)
38    CONTINUE
      MII=MIP-2
      DO 39 I=1,MII
      DO 39 J=1,MII
39    A(I,J)=0.0
      DO 41 I=1,MII
      A(I,I)=4.0
      IF(I-MII)40,41,41
40    A(I+1,I)=A(I,I+1)=1.0
41    CONTINUE
```

```
      CALL NEWJAC(25,MII,A)
      DO 43 I=1,MJP
      DO 42 J=1,MII
      L1=(J-1)*MJP+1
      L2=L1+2*MJP
      P(J)=(Z(L2)-Z(L1))*3.0
42    CONTINUE
      K2=(MIP-1)*MJP+I
      P(1)=P(1)-T(I)
      K1=I+MJP
      P(MII)=P(MII)-T(K2)
      CALL MULTMAT(AA,P,T(K1),MII,MJP)
43    CONTINUE
      XX=X( )
      YY=Y( )
      ZZ=Z( )
      CALL TRANS (XP,YP,XX,YY,ZZ,IPER)
      XM(1)=XP
      YM(1)=YP
      XX=X(MJP)
      YY=Y( JP)
      ZZ=Z(MJP)
      CALL TRANS (XP,YP,XX,YY,ZZ,IPER)
      XM(2)=XP
      YM(2)=YP
      K=(MIP-1)*MJP
      K1=K+1
      XX=X(K1)
      YY=Y(K1)
      ZZ=Z(K1)
      CALL TRANS (XP,YP,XX,YY,ZZ,IPER)
      XM(3)=XP
      YM(3)=YP
      K=K+MJP
      XX=X(K)
      YY=Y(K)
      ZZ=Z(K)
      CALL TRANS (XP,YP,XX,YY,ZZ,IPER)
      XM(4)=XP
      YM(4)=YP
      XMAX=-1.0E+200
      YMAX=XMAX
      XMIN=-YMAX
      YMIN=XMIN
      DO 207 I=1,4
      IF (XM(I)-XMAX) 201,203,200
200   XMAX=XM(I)
      GO TO 203
201   IF(XM(I)-XMIN) 202,203,203
202   XMIN=XM(I)
203   IF(YM(I)-YMAX) 205,207,204
204   YMAX=YM(I)
```

```
      GO TO 207
205   IF(YM(I)-YMIN) 206,207,207
206   YMIN=YM(I)
207   CONTINUE
      RANGEX=XMAX-XMIN
      RANGEY=YMAX-YMIN
      XSCALE=6.0/RANGEX
      YSCALE=6.0/RANGEY
      CALL PLOT(-3.0,0.0,-3)
      MJ1=MJP-1
      MI1=MIP-1
      DO 557 IP=1,MI1
      K=(IP-1)*MJP
      DO 557 JP=1,MJ1
      K1=K+1
      K2=K1+1
      K3=K1+MJP
      K4=K2+MJP
      W1=Z(K1)-Z(K3)
      W2=Z(K4)-Z(K2)
      W3=W1+W2
      W4=Z(K1)-Z(K2)
      W6=S(K1)-S(K3)
      W7=S(K2)-S(K4)
      W8=W6+W7
      W9=S(K1)+S(K3)
      W10=W9+S(K1)
      W11=T(K1)+T(K2)
      W12=T(K1)-T(K2)
      W13=T(K3)-T(K4)
      W14=W11+T(K1)
      R(1,1)=Z(K1)
      R(1,2)=S(K1)
      R(1,3)=-W4*3.0-W10
      R(1,4)=W4*2.0+W9
      R(2,1)=T(K1)
      R(2,3)=-W12*3.0
      R(2,4)=W12*2.0
      R(3,1)=-W1*3.0-W14
      R(3,2)=-W6*3.0
      R(3,3)=3.0*(3.0*W3+2.0*(W12+W6)+W13+W7)
      R(3,4)=-2.0*(3.0*W3+2.0*W12+W13)-3.0*W8
      R(4,1)=2.0*W1+W11
      R(4,2)=2.0*W6
      R(4,3)=-3.0*(2.0*W3+W12+W13)-4.0*W6-2.0*W7
      R(4,4)=2.0*(2.0*W3+W12+W13+W8)
      U=0.0
      DO 556 IU=1,IUSTOP
      V=0.0
      DO 555 IV=1,IVSTOP
      ZZ=0.0
```

```
      DO 550 I=1,4
      DO 550 J=1,4
      UM=1.0
      VM=1.0
      IF (I-1) 548,548,547
  547 UM=U**(I-1)
  548 IF (J-1) 550,550,549
  549 VM=V**(J-1)
  550 ZZ=ZZ+R(I,J)*UM*VM
      XX=(1.0-V)*((1.0-U)*X(K1)+X(K3)*U)+V*(X(K2)*(1.0-U)+U*X(K4))
      YY=(1.0-V)*((1.0-U)*Y(K1)+Y(K3)*U)+V*(Y(K2)*(1.0-U)+U*Y(K4))
      CALL TRANS (XP,YP,XX,YY,ZZ,IPER)
      XP=(XP-XMIN)*XSCALE
      YP=(YP-YMIN)*YSCALE
                  GO TO 553
  551 STOP
  552 CALL PLOT(XP,YP,3)
      GO TO 555
  553 IF(IV-1)551,552,554
  554 CALL PLOT(XP,YP,2)
  555 V=V+VSTEP
  556 U=U+USTEP
  557 CONTINUE
      CALL CONCR
      GO TO 1
      END

         3200 FORTRAN DIAGNOSTIC RESULTS - FOR   REVIEW

NO ERRORS

                                    3200   FORTRAN    (2.2)          /  /

      SUBROUTINE ACCEL(X,M)
      DIMENSION X(1)
C ***
C
C        NOTE THAT THIS SUBROUTINE MUST BE SUPPLIED BY THE USER.
C        ON EXIT THE ARRAY X SHOULD CONTAIN ACCELEROMETER DATA
C
C ***
      DO 1 I=1,M
    1 X(I)=0.0
      RETUR
      END

         3200 FORTRAN DIAGNOSTIC RESULTS - FOR   ACCEL

NO ERRORS
```

```
      SUBROUTINE MATMULT (A,P,S,M,MJP)
      DIMENSION A(1),P(1),S(1),PP(25)
      MK=1
1     DO 2 I=1,M
      PP(I)=0.0
      K=(I-1)*M
      DO 2 J=1,M
      L=K+J
2     PP(I)=PP(I)+A(L)*P(J)
      GO TO (3,6),MK
3     DO 4 I=1,M
4     S(I)=PP(I)
5     RETURN
6     K=1
      DO 7 I=1,M
      S(K)=PP(I)
7     K=K+MJP
      GO TO 5
      ENTRY MULTMAT
      MK=2
      GO TO 1
      END
```

3200 FORTRAN DIAGNOSTIC RESULTS - FOR MATMULT

NO ERRORS


```
      SUBROUTINE CUBIC(Y,X,T,M,N)
      DIMENSION Y(1),X(1),T(1),XX(4),YY(4),S(3,4),SV(3),B(3)
      DO 1 I=1,4
      XX(I)=X(I)
1     YY(I)=Y(I)
2     SX1=SX2=SX3=SX4=SX5=SX6=SYN1=SYN2=SYN3=SY=0.0
      Z=XX(1)
      IF(N-1)103,103,102
102   Z=XX(4)
103   CONTINUE
      DO 3 I=1,4
      U=XX(I)
      SX6=SX6+U*U*U*U*U*U
      SX5=SX5+U*U*U*U*U
      SX4=SX4+U*U*U*U
      SX3=SX3+U*U*U
      SX2=SX2+U*U
      SX1=SX1+U
      V=YY(I)
```

```
      SYN3=SYN3+V*U*U*U
      SYN2=SYN2+V*U*U
      SYN1=SYN1+V*U
3     SY=SY+V
      S(1,1)=SX2-SX1*SX1/4.0
      S(1,2)=S(2,1)=SX3-SX2*SX1/4.0
      S(2,2)=SX4-SX2*SX2/4.0
      S(1,3)=S(3,1)=SX4-SX3*SX1/4.0
      S(3,2)=S(2,3)=SX5-SX3*SX2/4.0
      S(3,3)=SX6-SX3*SX3/4.
      CALL NEWJAC(3,3,S)
      SV(1)=SYN1-SY*SX1/4.0
      SV(2)=SYN2-SY*SX2/4.0
      SV(3)=SYN3-SY*SX3/4.0
      CALL MATMULT(S,SV,B,3,4)
      F(1)=(B(3)*Z+B(2))*Z+B(1)
      RETURN
      ENTRY RECUBE
      DO 4 I=1,4
      J=(I-1)*M+1
      XX(I)=X(J)
4     YY(I)=Y(J)
      GO TO 2
      END
```

3200 FORTRAN DIAGNOSTIC RESULTS - FOR CUBIC

0 ERRORS

3200 FORTRAN (2.2) / /

```
      SUBROUTINE NEWJAC (NX,N,A)
      DIMENSION A(1)
      N2=N+N
      NN=N+1
      DO 2 J=NN,N2
      LA=(J-1)*NX
      DO 1 I=1,N
      KA=LA+I
    1 A(KA)=0.0
      KA=LA+J-N
    2 A(KA)=1.0
      DO 8 I=1,N
      K1=I+N
      LA=(I-1)*NX+I
      B=A(LA)
      DO 3 J=I,K1
      LA=(J-1)*NX+I
    3 A(LA)=A(LA)/B
      DO 7 K=1,N
      IF (K-I) 4,7,4
```

```
    4 LA=(I-1)*NX+K
      IF (A(LA)) 5,7,5
    5 BB=A(LA)
      DO 6 J=I,K1
      LA=(J-1)*NX
      KA=LA+K
      LA=LA+I
    6 A(KA)=A(KA)-A(LA)*BB
    7 CONTINUE
    8 CONTINUE
      IF (NN.EQ.N) 11,9
    9 LA=NX*(NX-N)
      DO 10 J=1,N
      N2=N2-1
      NI=N
      K=N2*NX
      DO 10 I=1,N
      KK=K+NI
      KA=KK-LA
      A(KA)=A(KK)
   10 NI=NI-1
   11 RETURN
      END
```

3200 FORTRAN DIAGNOSTIC RESULTS - FOR NEWJAC

NO ERRORS

3200 FORTRAN (2.2) / /

```
      SUBROUTINE TRANS (XP,YP,X,Y,Z,IPER)
      COMMON S1,S2,S3,C1,C2,C3,D,CX,CY,CZ,DCL,DCM,DCN,OX,OY,OZ,SL,SM,SN
      IF (IPER) 1,1,3
1     XP=-X*SL*S1+Y*SM*S2-Z*SN*S3
      YP=-X*SL*C1-Y*SM*C2+Z*SN*C3
2     RETURN
3     AK=D/((X-CX)*DCL+(Y-CY)*DCM+(Z-CZ)*DCN)
      A1=CX+AK*(X-CX)
      A2=CY+AK*(Y-CY)
      A3=CZ+AK*(Z-CZ)
      IF(ABS(DCN-1.0)-0.000001)5,5,4
4     XP=((A1-OX)*DCM-(A2-OY)*DCL)/S3
      YP=(A3-OZ)/S3
      GO TO 2
5     XP=(-DCN*(A1-OX)+(A3-OZ)*DCL)/S2
      YP=(A2-OY)/S3
      GO TO 2
      END
```

3200 FORTRAN DIAGNOSTIC RESULTS - FOR TRANS

NO ERRORS