Shape Function Interpolation of 2D and 3D Finite Element Results

Victor Genberg

Commercial & Government Systems
Eastman Kodak Company
Rochester, NY 14653-8116

Presented at
1993 MSC World's User Conference
May, 1993

Abstract

A finite element program such as MSC/NASTRAN provides displacements, rotations or temperatures at grid points only. There are a variety of applications which require results at other locations in the model:

1) Obtaining displacement boundary conditions on finer resolution breakout models
2) Optical ray tracing on deformed mirror surfaces
3) Placing temperatures from a coarse thermal model on to a finer structural model.

All of these applications involve interpolation of results over the finite element model.This paper describes a general purpose post-processing program to accurately interpolate over a model, using element shape functions. The user may specify a choice of linear interpolation for thermal models and solid elasticity models, or cubic interpolation for plate and shell models.

Introduction:

A finite element program such as MSC/NASTRAN [1] presents it's analysis results at grid points. There are many cases where it is necessary to interpolate results at intermediate geometric locations. A few example applications where interpolation is required are given below.

a) Temperatures determined from a coarse heat transfer model are to be applied to a detailed structural model for thermoelastic analysis, but the grid patterns for the 2 models do not coincide.

b) Structural displacements from a system level model are to be applied as boundary conditions on a detailed breakout model, but the breakout model has more grids on it's boundary than the same region in the system model.

c) A ray tracing program which bounces many random rays off deformed optical surfaces requires accurate displacement and slope information at ray-surface intersection points which are in general not at grid points.

d) Post processing programs such as optical evaluation codes require deformed surface data on a regularly spaced square pattern which typically do not line up with the finite element grid points.

In each case, interpolation within a 3D solid or over a 2D surface is required. This paper describes a post-processing program to provide this capability.

Linear Interpolation over a Solid:

When interpolating temperature results from a "coarse" heat transfer model to a "detailed" structural model the following procedure works effectively.

Heat Transfer: Create and solve the coarse heat transfer model using standard MSC/NASTRAN elements including: ROD, BAR, QUAD4, TRIA3, HEXA, PENTA, TETRA. This analysis can be conducted in MSC/NASTRAN or in a code like SINDA. For purposes of interpolation the thermal model must be represented as standard low-order finite elements which have geometry. (Note that this NASTRAN model could be created first and then the resulting MSC/NASTRAN conductivity, capacitance, radiation, loads, and boundary conditions matrices output and reformatted as SINDA input).

Interpolation: Create the detailed structural model as desired. Pass each structural grid through the search algorithm to find its interpolated temperature and output a corresponding TEMP card for each subcase.

Thermoelastic: Include the interpolated TEMP cards as thermal loads in the MSC/NASTRAN structural model.

A flow chart of the interpolation program is shown in Figure 1. Due to use of the program on a variety of computer types which may be different from the computer on which MSC/NASTRAN was run, all program inputs and outputs are in ASCII form. This feature makes use of binary OUTPUT2 files unnecessary. Instead, output features such as ECHO=PUNCH and THERMAL(PUNCH) are utilized to transfer data.

Since the interpolation is a geometric procedure, all grid locations must be converted to a common coordinate system (step 2). This common system may be the basic system or any other coordinate system in the model. Temperature results are scalar requiring no coordinate transformation, but when interpolating displacement results, the vectors must be converted to a common system also. At this point, it is convenient to include the option to allow linear combinations of input vectors, or scaling of inputs to other units.

In step 3 of the flow chart, all elements are converted to equivalent tetrahedrons or cylinders. For example, a two dimensional QUAD4 is first made into a solid HEXA by creating two surfaces parallel to the element but displaced a distance of 1/2 the thickness in the positive and the negative surface normal directions. This HEXA is subdivided into five TETRA elements as in the original NASTRAN HEXA1 element. In a similar manner a TRIA3 is converted to a PENTA, then subdivided into TETRAs. All one dimensional elements are arbitrarily converted into solid circular cylinders of equivalent cross-sectional area.

Cylinders and tetrahedrons are used because their Jacobian transformation matrix [J] is a constant throughout the element and not a function of spatial location as in a general hexahedron. The transformations can be calculated once and then stored for the search routine. In step 4, the element transformation matrices from spatial coordinates $\{x\}$ to parametric coordinates $\{\xi\}$ are calculated for the resulting cylinders and tetrahedrons. For element *m*,

$$[J_m] = [dx/d\xi \; ....]$$

The full matrices for the tetrahedron and cylinder are listed in the appendix. The origin of element of element *m* in parametric coordinates $\{\xi_0\}$ can be found from the spatial center $\{x_0\}$

$$\{\xi_0\} = [J_m]^{-1} \{x_0\}$$

For every grid point $\{x_p\}$in the new (detailed) model (step 5), a search over all elements in the old (coarse) model is conducted in steps 6-8. First, convert the grid location to element *m*'s parametric coordinates $\{\xi_p\}$

$$\{\xi_p\} = [J_m]^{-1} \{x_p\} + \{\xi_0\}$$

then test to see if contained within element *m*.

$$0-\varepsilon \leq \xi_p \leq +1+\varepsilon$$

A user controlled tolerance ε is placed on the search to find points on curved surfaces which may fall slightly outside of straight edged elements representing the surface. The search is conducted first with $\varepsilon = 0$ to find points totally within elements. Points failing this first pass are searched again with a user specified tolerance. All TETRAs are searched before the cylinders, because the tetrahedron representation is more accurate than the cylinders.

When the proper element is found containing the grid point , the interpolation takes place in Step 9 using the element's shape functions (N) and the element's nodal results, such as temperatures (T).

$$T(x_p) = \Sigma \; N_j(\xi_p) * T_j$$

Example of Linear Interpolation over a Solid:

Two examples of linear interpolation are given for finding temperature inputs for a thermoelastic analysis. In Figure 2a, a simple QUAD4 mdel with skewed elements was used to analyze a linear gradient (Figure 2b) where the sides are insulated, the bottom fixed, and a flux input to the top. The temperatures were interpolated on the detailed 3D HEXA structural model (Figure 2c). The resulting temperature contours are identical with the thermal model.

In the second example, a coarse 3D heat transfer model of a mirror and its support ring (Figure 3a) were analyzed for a surface flux with temperature contours as shown in Figure 3b. Temperature results are interpolated on to a detailed 3D structural model of the mirror and its support ring (Figure 3c). The interpolated temperatures are shown in Figure 3d showing the resulting contours are essentially identical. This technique is much more accurate than the common alternative in which the temperatures for the 3D structural model were obtained by a separate thermal analysis where the closest grids were fixed to the thermal model results and intermediate grids were found by an additional heat transfer analysis on the structural model. In an optical element, small displacements are important to optical performance , so accurate interpolation is required.

Cubic Interpolation over a General Surface:

In a ray trace program evaluating a deformed surface, the plate bending behavior is required for accurate slope information at intermediate ray-surface intersection points. The surface interpolation scheme is similar to Figure 1 with the changes a noted below.

In step 1, a specific surface within the model representing the optical surface is chosen with its corresponding coordinate system. This subset of grids can be specified by giving a range of grid numbers or by giving a coordinate system number for the coordinate position (CP) or the displacement coordinates (CD). The set of elements (QUAD4/TRIA3) comprising the surface can be specified by a range of element numbers or by specifying a property number for the PSHELL. These are all convenient techniques for defining a subset of the model.

As in the previous section, all grid points, displacements, and rotations must be converted to a common coordinate system (step 2). This common system could be the basic system or any other system in the model. If desired, this system may be present for no other reason than for interpolation. Again, this is a convenient point to allow linear combinations or scaling of input vectors.

The only element types supported in this interpolation are the low order plate and shell elements (QUAD4 and TRIA3). Non-rectangular quadrilaterals are converted to two triangles in step 3.

Rectangles and triangles are used because their Jacobian transformation matrix [J] is a constant throughout the element and not a function of spacial location as in a general quadrilateral.. The transformations can be calculated once and then stored for the search routine. In step 4, the element transformation matrices from spatial coordinates {x} to parametric coordinates { ξ } are calculated for the resulting rectangles and triangles. For element *m*,

$$[J_m] = [dx/d\xi \;....]$$

The origin of element of element *m* in parametric coordinates $\{\xi_0\}$ can be found from the geometric center $\{x_0\}$

$$\{\xi_0\} = [J_m]^{-1} \{x_0\}$$

For every grid point $\{x_p\}$ in the new (detailed) model or ray point $\{x_p\}$, a search over all elements is conducted in steps 6-8. The point is converted to element *m*'s parametric coordinates $\{\xi_p\}$

$$\{\xi_p\} = [\, J_m]^{-1} \{x_p\} + \{\xi_0\}$$

then tested to see if contained within element *m*.

$$-1-\varepsilon \le \xi_p \le +1+\varepsilon$$

A user controlled tolerance ε is placed on the search to find points on curved surfaces which may fall slightly outside of straight edged elements representing the surface. The search is conducted first with ε =0 to find points totally within elements. Points failing this first pass are searched again with a user specified tolerance.

When the proper element is found containing the point, the interpolation takes place in step 9 using the element's shape functions (N) and the element's nodal results, such as displacements (U). In-plane motion is found from membrane behavior

$$u(x_p) = \Sigma\, N_j(\xi_p) * U_j$$

The out-of-plane displacement $w(x_p)$ and slopes w_x and w_y are found from plate bending behavior as given in Appendix B. The equations have the general form of

$$w(x_p) = \Sigma \; [f_j W_j + g_j W_{xj} + h_j W_{yj}]$$

where W_j, W_{xj}, and W_{yj} are the nodal displacements and rotations.

Small displacement theory plate bending behavior provides slope data due to surface normal displacements and rotations. For curved surfaces, a ray intersection point may have a change of slope due to in-plane displacements. This rigid body motion correction for a cylinder would be:

$$R'_z = R_z - D_\theta / r$$

where:

D_θ = displacement in θ direction on cylinder from $u(x_p)$
R_z = rotation about z axis as interpolated in step 9.
r = radius of the cylinder

This feature is provided as a user selected option.

Examples of Cubic Interpolation over a Surface:

A simple test case with triangles, rectangles and multiple coordinate systems is shown in Figure 4. This was given unit displacements and unit slopes at the central nodes which result in the theoretical curves shown. The plate interpolation equations in the appendix provided excellent correlation with theory.

The two mirror system shown in Figure 6 was used to test the ray-trace algorithm which used interpolation over the shell surface. The incoming rays are collected at the focal plane by very low angle of incidence rays grazing off mirrors which are nearly cylindrical. Actually the surfaces are slightly parabolic and hyperbolic. The interpolation was verified by forcing a known functional displacement over both mirror surfaces. The interpolated rays were then compared to the rays from the perfect functional surface, again with excellent agreement.

In an experimental test configuration these mirrors are mounted at 12 points about a belt line. In a horizontal test configuration in gravity, there are local bending effects about the mount points which must be considered in the ray trace algorithm. These highly local distortions require the local interpolation over a single element for accuracy. Global functions must be of very high order to describe these effects. The interpolated ray trace agreed well with the experimental test.

Useful Program Features:

There are a variety of features which makes this program a good all purpose tool. Some are listed below.

1) Coordinate system conversions for grid points:
Grids located in any coordinate system can be transformed to any other coordinate system. The converted grids can be "punched" in new coordinate system with the updated CP field. The displacement coordinate system field CD may be changed also to match item 2 below.

2) Coordinate system conversions for displacements:
Displacements may be transformed to any other system and "punched" for futher post-processing or plotting. Disjoint superelement output is handled in the program.

3) Linear combinations of results:
Any linear combination of subcases may be interpolated or transformed and output for further post-processing. This can be useful for scaling or changing units

4) Output formats:
A variety of output formats are available for interpolated, combined or transformed results making them useful in other programs for analysis or display.

* TEMP cards for thermoelastic analysis
* SPC cards for breakout models in thermal analysis
* SPC cards for breakout models in structural analysis
* DISP punch format to simulate MSC/NASTRAN output
* EXT format for MSC/XL processing
* NOD format for PATRAN processing
* ARRAY format for optical processing

5) Summary tables:
All vectors for position and results are summarized for model checking:

* Number of grids and elements processed and used
* Number of coordinate systems and transformation matrix
* Max, Min, and Average values for grid location components
* Max, Min, and Average values for displacement components

6) Submodel specification:
Selection of surface elements and grids through simple criteria

* GRIDs by grid id range
* GRIDs by position coordinate system id (CP)
* GRIDs by output coordinate system id (CD)
* GRIDs by nonzero load from a pressure load (OLOAD)
* ELEMENTs by element id range
* ELEMENTs by property id (PID)

Suggested Improvements:

HEXAs and non-rectangular QUADs are converted to TETRAs and TRIAs for the search algorithm and then the interpolation. The efficiency of a constant [J] requires this conversion for the search algorithm. There is no loss of accuracy in the geometry because geometrically the QUAD is exactly equal to 2 TRIAs and the HEXA is exactly equal to 5 TETRAs. Once the proper element has been found, the interpolation algorithm however could use the shape functions for the original generally-shaped QUADs and HEXAs without a great loss of efficiency. Also, the shear deflection term could be included in the analysis.

Conclusions:

A post-processing program for 2D and 3D interpolation is a valuable tool for many applications. This tool is an accurate method for thermoelastic analysis and optical ray tracing. Many other applications, as noted in the introduction, can benefit from this capability.

Acknowledgements:

The programming for interpolation of temperature results over a 3D solid was written by Geoff Philbrick.

References:

[1] *MSC/NASTRAN User's Manual, Version 67*, MacNeal-Schwendler Corp., 1991

[2] Yang, T. Y., *Finite Element Structural Analysis*, Prentice-Hall, 1986

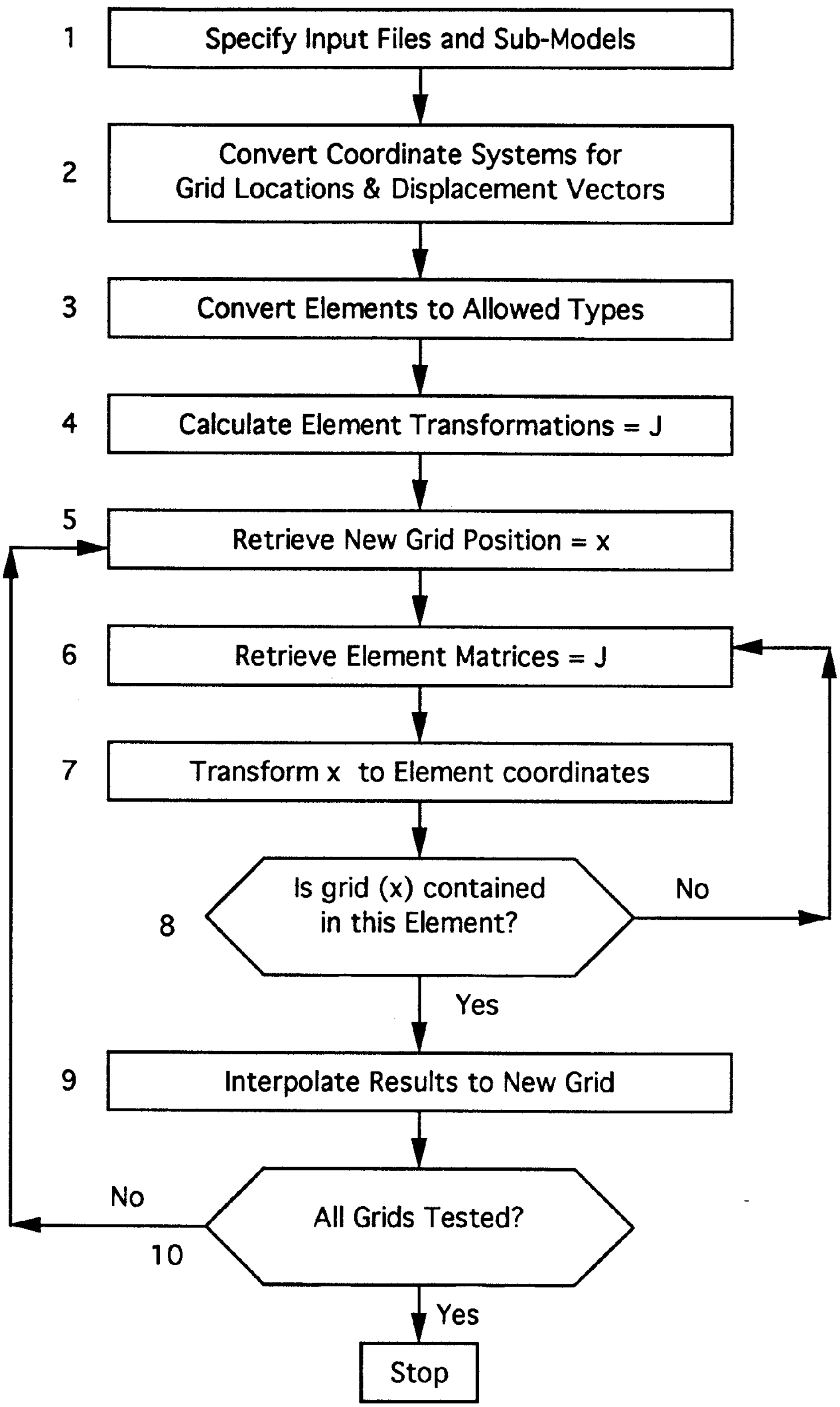
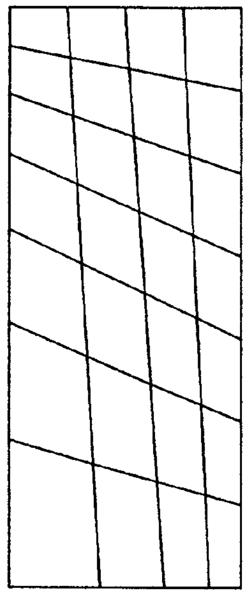
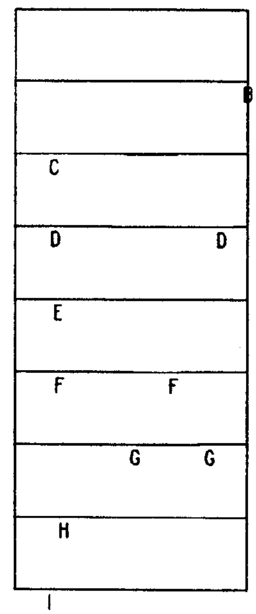


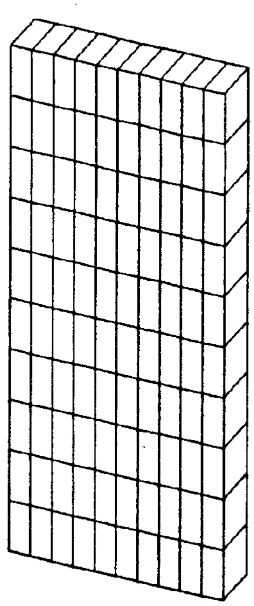
Figure 1: Interpolation Flow Chart



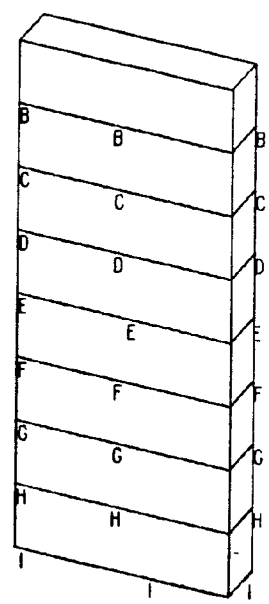
a) Coarse 2D Thermal Model



b) Heat Transfer Temperature Results

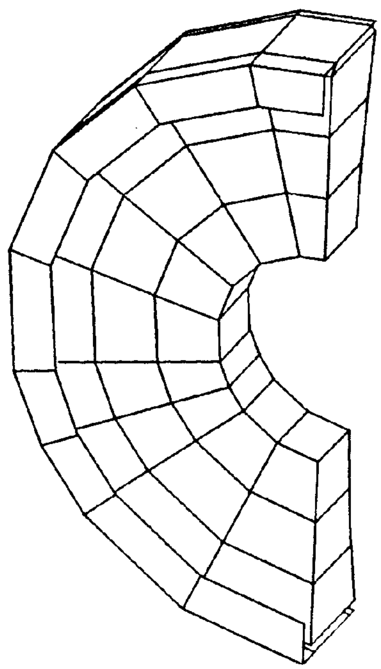


c) Detailed 3D Structural Model

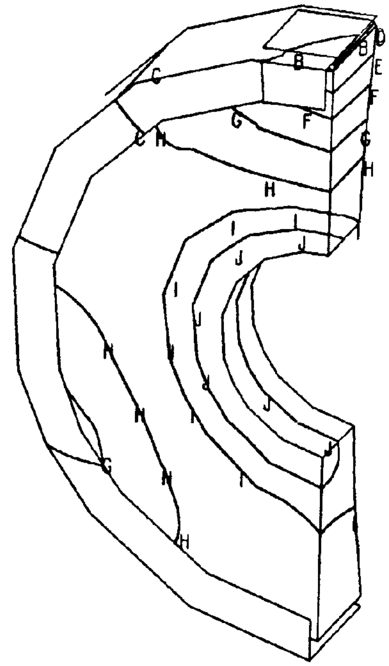


d) Interpolated Temperatures

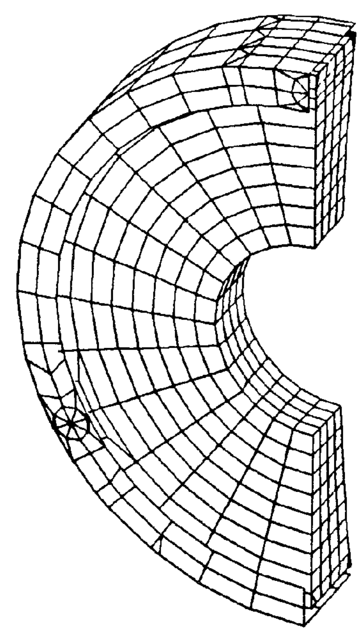
Figure 2: Simple Interpolation Test Case



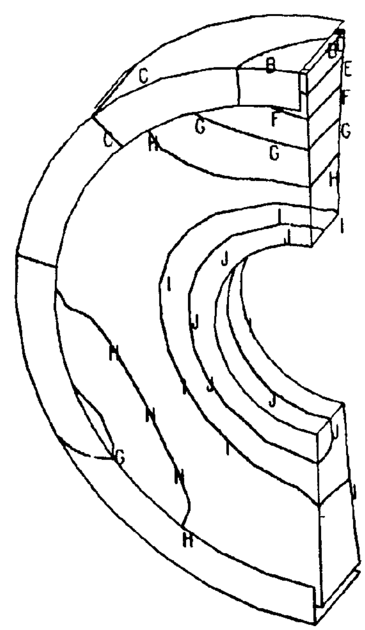
a) Coarse 3D Thermal Model



b) Heat Transfer Temperature Results



c) Detailed 3D Structural Model



d) Interpolated Temperatures

Figure 3: Interpolation on Mirror and Mount

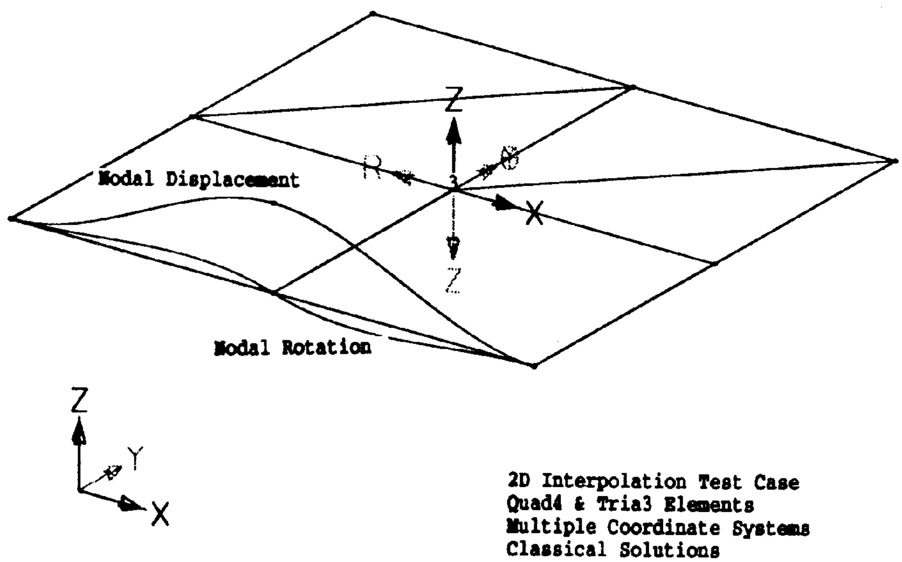


Figure 4: Plate Bending Interpolation Test Case

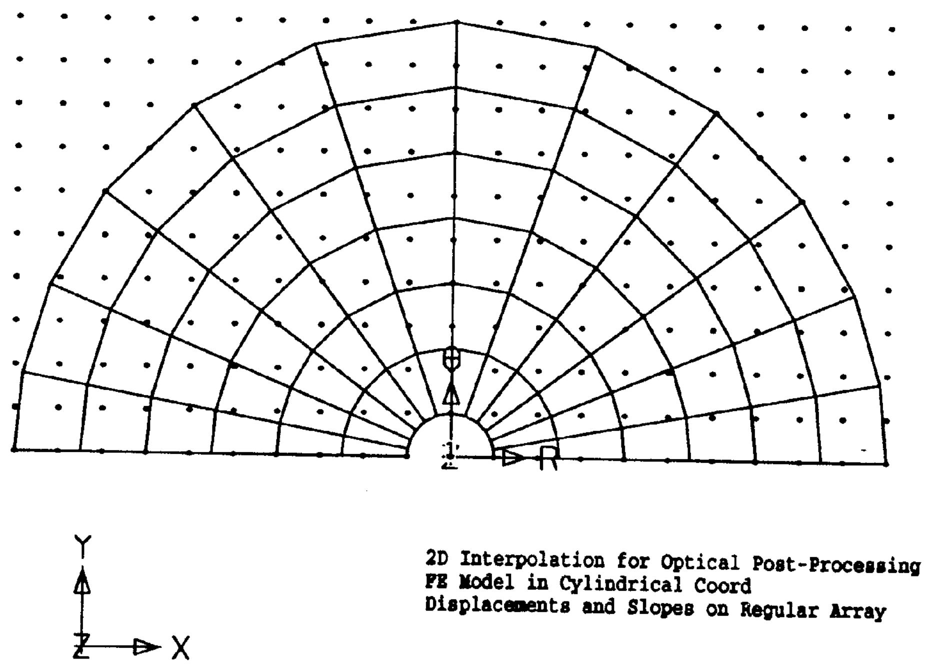


Figure 5: Optical Post-Processing Array

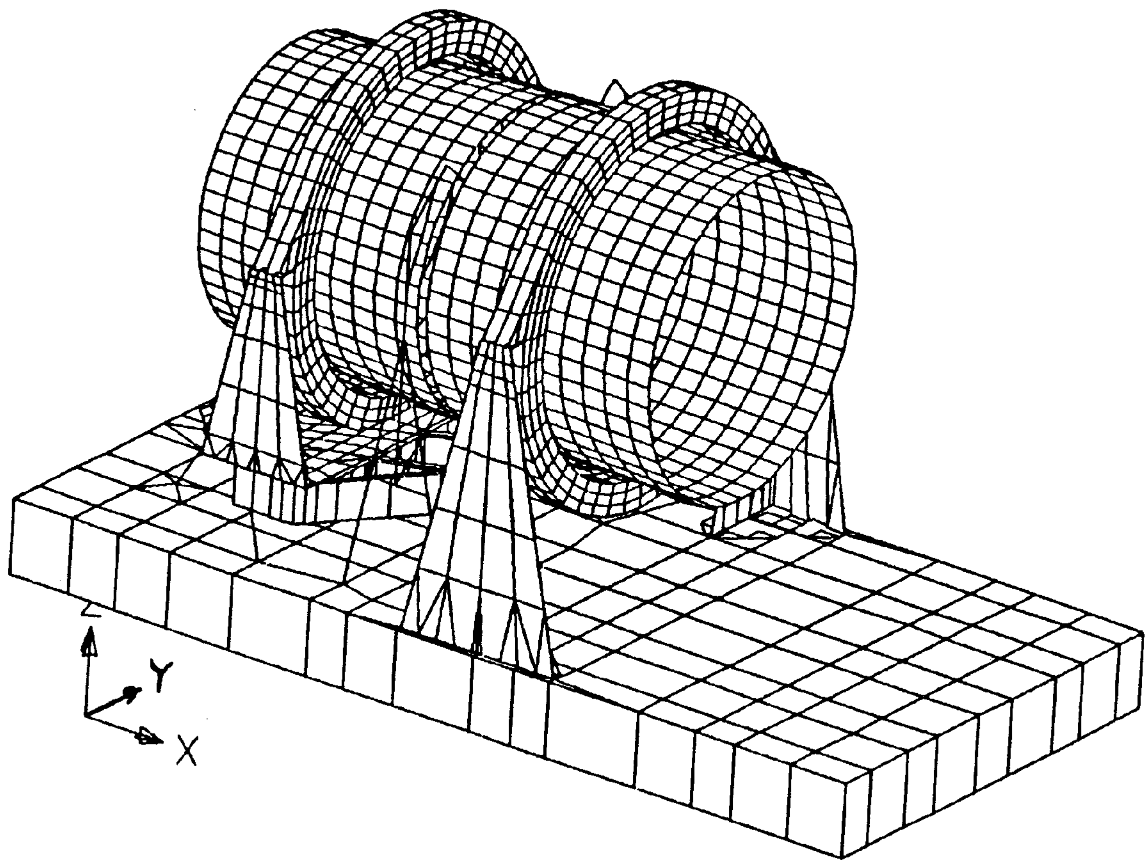


Figure 6: Two Mirror Ray Trace Model

APPENDIX

Triangular Plate Bending Behavior:

The equations used for triangular plate behavior are given in Yang [2] in Section 12.4.4 using area coordinates ξ_j

$$w(\xi_1,\xi_2,\xi_3) = \Sigma\ [\, f_j W_j + g_j W_{xj} + h_j W_{yj} + e_j\{w_{nj} - \widetilde{W}_{nj}\}\,]$$

where the nodal displacement and slopes are W_j, W_{xj}, W_{yj} and coefficients are given by:

$$f_1 = \xi_1 + \xi_1^2\xi_2 + \xi_1^2\xi_3 - \xi_1\xi_2^2 - \xi_1\xi_3^2$$

$$g_1 = (x_2 - x_1)\,(\xi_1^2\xi_2 + 0.5\,\xi_1\xi_2\xi_3) + (x_3 - x_1)\,(\xi_1^2\xi_3 + 0.5\,\xi_1\xi_2\xi_3)$$

$$h_1 = (y_2 - y_1)\,(\xi_1^2\xi_2 + 0.5\,\xi_1\xi_2\xi_3) + (y_3 - y_1)\,(\xi_1^2\xi_3 + 0.5\,\xi_1\xi_2\xi_3)$$

Permute subscripts to obtain f_2, f_3, etc. The last coefficient is:

$$e_j = (8A / L_{12})\,(\xi_1^2\xi_2^2\xi_3) / (\xi_3 + \xi_1) / (\xi_3 + \xi_2)$$

where L_{12} = length of side 1-2

A = area of the triangle

The subscript n refers to slope normal to an edge. w_{nj} is the average of the two nodal normal slopes on edge j. $\widetilde{W}_{nj}$ is found from differentiating the following function and evaluating it at the midside of edge j.

$$\widetilde{W}(\xi_1,\xi_2,\xi_3) = \Sigma\ [\, f_j W_j + g_j W_{xj} + h_j W_{yj}]$$

These equations represent a 9 degree-of-freedom conforming triangle.

Rectangular Plate Bending Behavior:

The equations used for rectangular plate behavior are given in Yang [2] in Section 12.3.3 where ξ,η are standard isoparametric coordinates. These equations represent a 4 node 16 degree-of-freedom conforming rectangle.

$$w(\xi,\eta) = \Sigma\ [\,G_j(\xi)\,G_j(\eta)\,W_j + (a/2)\,H_j(\xi)\,G_j(\eta)\,W_{xj}$$

$$+ (b/2)\,G_j(\xi)\,H_j(\eta)\,W_{yj} + (ab/4)\,H_j(\xi)\,H_j(\eta)\,W_{xyj}\,]$$

where

$$G_j(\xi) = (-\xi_j\xi^3 + 3\xi_j\xi + 2) / 4$$

$$H_j(\xi) = (\xi^3 + \xi_j\xi^2 - \xi - \xi_j\,) / 4$$

The nodal displacement and slopes are W_j, W_{xj}, W_{yj} and

a = length in the x direction
b = length in the y direction

Tetrahedron Equations:

The tetrahedron is the primary element for linear interpolation in a general solid. The 4 nodal coordinates can be represented in geometric or parametric space as:

	Geometric			**Parametric**		
Node	**X**	**Y**	**Z**	ξ	η	ζ
1	x1	y1	z1	0	0	0
2	x2	y2	z2	1	0	0
3	x3	y3	z3	0	1	0
4	x4	y4	z4	0	0	1

The shape functions are

$$N_1 = 1 - \xi - \eta - \zeta$$

$$N_2 = \xi$$

$$N_3 = \eta$$

$$N_4 = \zeta$$

The Jacobian matrix is

$$[J] = \begin{bmatrix} (x2-x1) & (x3-x1) & (x4-x1) \\ (y2-y1) & (y3-y1) & (y4-y1) \\ (z2-z1) & (z3-z1) & (z4-z1) \end{bmatrix}$$

When testing to see if a point x_p is located within element *m*, convert x_p to element *m*'s parametric coordinates, then the point must pass the following 4 tests to be contained within *m*

$$0 < \xi < 1$$
$$0 < \eta < 1$$
$$0 < \zeta < 1$$
$$0 < \xi+\eta+\zeta < 1$$

Cylinder Equations:

The cylinder is used for all 1D elements for linear interpolation in a general solid. The 2 nodal coordinates can be represented geometric or parametric space as:

	Geometric			Parametric		
Node	**X**	**Y**	**Z**	ξ	η	ζ
1	x1	y1	z1	0	0	0
2	x2	y2	z2	1	0	0

The shape functions are

$$N_1 = 1 - \xi$$
$$N_2 = \xi$$

The radius of the cylinder is

$$R = \text{sqrt}(A/\pi)$$

The transformation matrix is

$$[J] = \begin{bmatrix} lx & mx & nx \\ ly & my & ny \\ lz & mz & nz \end{bmatrix}$$

where the direction cosines of the vector along the member are

$lx = x2 - x1$ $\quad$ $ly = y2 - y1$ $\quad$ $lz = z2 - z1$

and the other vectors of length R are normal to the member using the $\bar{v}$ vector:

$\bar{m} = R^* (\bar{l} \times \bar{v})$ $\quad$ $\bar{n} = R^*(\bar{l} \times \bar{m})$

When testing to see if a point x_p is located within element *m*, convert x_p to element *m*'s parametric coordinates, then the point must pass the following 2 tests to be contained within *m*

$$0 < \xi < 1$$
$$0 < \text{sqrt}(\eta^2 + \zeta^2) < 1$$