CONNECTING SOLID FINITE ELEMENT MODELS THAT HAVE DISSIMILAR MESHES ON THE MATING SURFACES

John L. Bretl
Senior Project Engineer

Structural Dynamics Research Corporation
Milford, Ohio

ABSTRACT

This paper describes a method to connect two solid models along surfaces that have non-coincident meshes, i.e., the grid points and elements on the two surfaces do not line up. The essence of the method is that the displacement of each grid point on one of the surfaces is made dependent on the three or four surrounding grid points from the other surface. The MSC-NASTRAN interpolation constraint element, RBE3, is used. Element shape functions are used to determine the needed weight factors. The method allows two modelers to independently build complex solid models that eventually must be connected, without the constraint that the meshes on the connecting surfaces must match up. Also, the method is an easy way to transition from a coarse to a fine mesh, something that is normally quite difficult to do in a solid model.

INTRODUCTION AND PURPOSE

As the use of finite element analysis continues to grow, the types of structures being analyzed are no longer restricted to two-dimensional type structures that can be modeled with beam and shell elements. Many of the structures being analyzed today are such that they must be modeled with three-dimensional brick elements. Examples are heavy suspension parts such as steering knuckles and almost any engine part (blocks, heads, pistons, connecting rods, etc.). Most cast items would be modeled with brick elements because they are generally thick enough and non-uniform enough that beam or shell theory would not apply. In this discussion, let us refer to these brick element models as "solid models".

Building a solid model of a complex three-dimensional structure is significantly more difficult than modeling a complex shell-type structure. The restrictions on the modeler can be quite severe; for example:

- The elements cannot be too distorted or warped if reasonable accuracy is expected.
- Five-sided wedge elements will give poorer results than six-sided bricks.
- The mesh must be fine enough in areas of severe geometry changes to model the high stress gradients that exist there.
- The mesh should not be overly fine in any area where it is not needed because this may make the model too large and/or too expensive to run.
- The general mesh pattern in one area of a structure often conflicts with the pattern required in another area. The mesh must transition from one pattern to another.

As is inferred in the items above, mesh transitioning may be needed for two different reasons. The first is to transition from a fine mesh in an area of severe geometry or an area of particular concern to a coarse mesh where the geometry is uniform or, perhaps, accurate stress results are not needed. The second reason for mesh transitioning is to be able to model two parts of a structure that really have different geometries but must match up along a certain boundary. A good example of this is an engine block and head. Generally, the geometry on the block and head mating surfaces do not line up. Some of the water holes are common between the two but not all, and those that are common are generally not the same size and shape on the head as on the block. If the head and block are to be analyzed together, the mesh on the lower surface of the head must have the characteristics of the block geometry in addition to the geometry characteristics of the head itself. Likewise, the mesh on the upper surface of the block must include the head geometry in addition to its own. These restrictions force the mesh at the interface surface to be quite fine and, because this fine mesh must propagate into the block and head, the analysis may become quite expensive.

Mesh transitioning, whether needed to change from fine to coarse mesh or because of geometry, is much more difficult to accomplish in a solid model than in a shell model. It takes a significant amount of modeling time and often results in very badly distorted elements. A method is presented here that removes the mesh transitioning restrictions of solid models. It is easy to implement and does not restrict element shapes.

DESCRIPTION OF METHOD*

Consider the two simple structures in Figure 1. We want to mathematically connect the two pieces over their entire interface surface, like gluing the two pieces together. Note that, in general, the two meshes do not line up, i.e., for any grid point on one of the mating surfaces, there is no corresponding grid point on the other.

If the grid points had lined up, we could accomplish this "gluing" by simply making the displacement of each grid point on the first of the mating surfaces dependent on the displacement of its corresponding grid point on the other surface. This would be accomplished by writing constraint equations or, more easily, by using constraint "elements" such as MSC-NASTRAN's RBAR. Generally, however, the grid points do not line up. Therefore, the displacement of each grid point on the first surface must depend on the displacement of not one, but three or four of the grid points on the second surface.

Consider Figure 2. Each grid point A from the first mating surface falls inside (or on the boundary of) a quadrilateral or a triangle on the second mating surface. Then, the displacement of point A must be a weighted average of the displacements of the three or four surrounding grid points. MSC-NASTRAN's interpolation constraint element, RBE3, is ideally suited for this application. It makes the displacement of one grid point dependent, in a weighted average sense, on the displacements of the others.

The only issue to resolve, then, is how to determine the weight factors. Obviously, the weight factors must depend on the location of grid point A relative to the surrounding grid points. If A is close to grid point 1, for example, the weight factor for grid point 1 should be large and the others small. Clearly, in the limit, when A is coincident with grid point 1, the weight factor is unity for grid point 1 and zero for the others. As this suggests, and as shown in the theory that follows, the weight factors are actually the shape functions of the isoparametric formulation. Knowing the location of A relative to the surrounding grid points, we can determine the value of the shape functions for this location. These, then, are the weight factors needed.

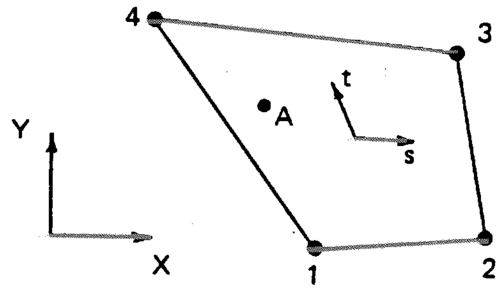
---

*The method is described here with reference to the MSC-NASTRAN finite element program, but is relevant for any finite element program that employs isoparametric solid elements. The ease of implementation, however, would depend on the particular program's method for inputting constraint equations.

THEORY

The theory is derived here only for linear isoparametric elements. However, it could be expanded to include higher order elements.

Consider one surface of a solid element. (Consider, for the time being, that the surface is quadrilateral. The theory for a triangular surface is a degenerate case of the quadrilateral).



The displacement of any point A can be written in isoparametric formulation [1] * as:

$$
\begin{Bmatrix} u_A \\ v_A \\ w_A \end{Bmatrix} = \begin{bmatrix} W_1 & W_2 & W_3 & W_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_1 & W_2 & W_3 & W_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_1 & W_2 & W_3 & W_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{Bmatrix}
$$

*Numbers in square brackets are references listed at the end of the paper.

where u, v, w are displacements in the x, y, z directions, the W_i are shape functions defined as:

$$W_1 = 1/4\,(1-s)\,(1-t)$$

$$W_2 = 1/4\,(1+s)\,(1-t)$$

$$W_3 = 1/4\,(1+s)\,(1+t)$$

$$W_4 = 1/4\,(1-s)\,(1+t)$$

and s, t are isoparametric coordinates defined to be +1 or −1 at the corner grid points.

The W_i are exactly the weight factors that we need for the weighted average input on the RBE3 element. For example, if A coincides with grid point 1 ($s = t = -1$) then $W_1 = 1$ and $W_2 = W_3 = W_4 = 0$. Likewise, if A coincides with any of the other corners, that weight factor is one and the others are zero. If A lies somewhere inside the quadrilateral, then s and t are less than one, and all the W_i have values between zero and one. The easiest example of that is $s = t = 0$, then all $W_i = 1/4$. That is to say, when A is at the center of the quadrilateral, its displacement depends equally on all four corner grid points.

For a general location of A, s and t are not known. How, then, do we determine the values of the four W_i?

We need four independent equations. The first two come from the fact that the elements are isoparametric, i.e., the same shape functions describe the element geometry as define the displacement field. The two equations are:

$$x_A = W_1 x_1 + W_2 x_2 + W_3 x_3 + W_4 x_4 \quad (1)$$

$$y_A = W_1 y_1 + W_2 y_2 + W_3 y_3 + W_4 y_4 \quad (2)$$

where the x_i and y_i are local x and y coordinates of the grid points. The other two equations are inherent in the form of the shape functions:

$$1 = W_1 + W_2 + W_3 + W_4 \quad (3)$$

$$W_1 * W_3 = W_2 * W_4 \quad (4)$$

These four equations can be solved for the four W_i. Equation (4) makes the set of equations nonlinear and they must be solved iteratively. Details of this are in the Appendix. This iterative solve introduces only a slight complication; it always converges and it converges very quickly.

The equations for a triangle are obtained by dropping the last term of equations (1), (2), and (3) and eliminating equation (4). This makes a linear set of equations that can be solved in one pass.

IMPLEMENTATION

A FORTRAN program was written to implement the method. Two files are required as input:

File 1: This input file contains the geometry of all the grid points on both of the mating surfaces.

File 2: This input file relates a single point on the first surface to three or four points on the second surface. In other words, this file tells the program that grid point A from the first surface "falls inside of" grid points 1, 2, 3, (4) of the second surface.

Which surface is the "first" surface, i.e., which grid points should be made dependent? The answer is that if a fine mesh is being connected to a coarse mesh, the grid points of the fine mesh should be dependent on the grid points of the coarse mesh. If the two meshes are about equal, either can be made dependent on the other. All dependent points should be on one surface, though. There should not be dependent points on both surfaces.

Figure 3 shows a flowchart of the program. Note that the program sets up a local coordinate system for each quadrilateral or triangle. This is done because the method is formulated with the quadrilateral or triangle in an xy plane. In general, then, although point A and the three of four corner points should be coplanar, they need not lie in a plane parallel to any of the model's three global Cartesian planes.

The output from the program is a file of RBE3 cards. For the general case, where point A falls somewhere inside a quadrilateral, the RBE3 card has the form:

RBE3	el. no.	X	grid A	123	W_1	123	grid 1	W_2	+BE3
+BE3	123	grid 2	W_3	123	grid 3	W_4	123	grid 4	

It is the same for a triangle except fields 7, 8, and 9 of the continuation card are left blank because there is no grid point 4.

There are two special cases that need to be considered. The first is when grid point A falls on the boundary of a quadrilateral or triangle, say, directly between corner grid points 1 and 2. For this case the program correctly computes that W_1 and W_2 are non-zero and W_3 and W_4 are zero, i.e., the displacement of A depends only on the displacements of grid points 1 and 2. For this case the RBE3 card must take a special form:

RBE3	el. no.	X	grid A	123	W_1	123456	grid 1	W_2	+BE3

+BE3	123	grid 2							

One obvious difference from the general case shown above is that no information is included for grid points 3 and 4. But note also field 7 of the parent card. All six degrees-of-freedom (dof) are included instead of just the translational dof. The reason is that the independent dof on an RBE3 card (grid 1, 2, . . .) must describe all six rigid body motions. For three or more non-colinear points this can be done with only translational dof. But with only two points as in this case, using only translational dof fails to account for rotation about a line through the two points. Therefore, for this case we include all six dof at grid point 1.

The other special case is when grid point A is coincident with one of the corner grids, say, grid point 1. Then the program correctly computes $W_1 = 1$ and $W_2 = W_3 = W_4 = 0$. For this case an RBE3 is not needed because the displacement of A depends, not on a group of grid points, but on only grid point 1. A standard RBAR is written for this case.

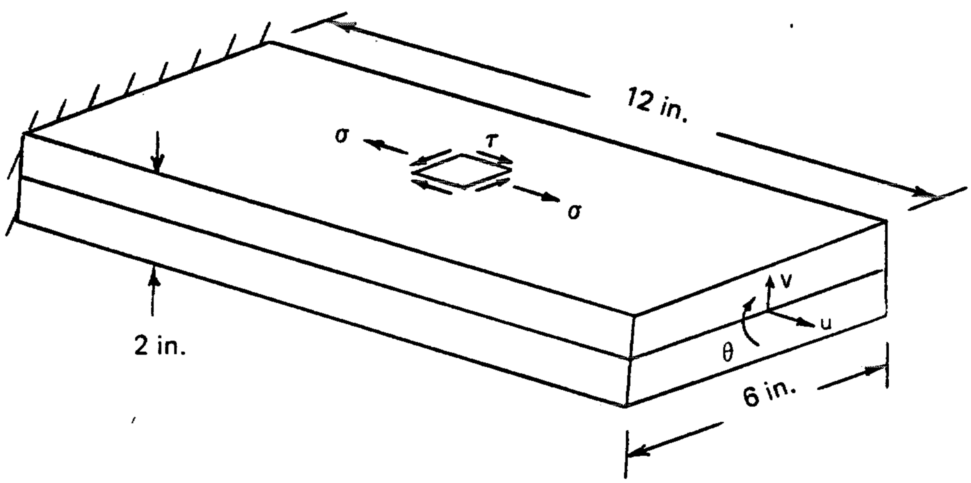
EXAMPLES

Figure 4 shows models of two thick plates, one with a uniform mesh and one with an irregular mesh. These two plates are "glued" together using the method described here. The first test is to make sure that rigid body motion causes no strain energy in the structure. Figure 5 shows plots of rigid body translation and rotation. For each case strain energy was computed in all elements and found to be zero.

The next test is to deform the structure. Three load cases were used: axial load, pure bending, and twist. The deformation shapes are shown in Figure 6. For comparison purposes two other models were also run. These were like the model of Figure 4 except that, in the first, both top and bottom plates have a uniform mesh and, in the second, both plates have the irregular mesh of the bottom plate in Figure 4. For these two models no special attachment was needed between the two plates because the meshes matched up exactly.

The table below compares results for the three models to theoretical results. Model 2 used the method described in this paper. The table shows that all results agree very closely with theory. Also, the Model 2 results are closer to the theoretical results than Model 3; this shows that any variance in Model 2 results from the theoretical results is due to basic finite element approximations and the irregularity of the mesh, not to the RBE3 method employed.

Model	Mesh		Results (per 10^5 lb force or 10^5 lb-in moment)					
			Axial Pull		Bending		Twist	
	Top Plate	Bottom Plate	u (in)	σ (psi)	v (in)	σ (psi)	θ (deg)	τ (psi)
1	Uniform	Uniform	.00333	8,333	.0600	−25,000	.436	15,100
2	Uniform	Irregular	.00333	8,333	.0600	−25,000	.431	14,805
3	Irregular	Irregular	.00333	8,333	.0602	−25,025	.431	14,333
Theory			.00333	8,333	.0600	−25,000	.454	15,605



E = 30E6 psi
ν = 0.25

Figure 7 shows a case of how the method was employed on project work. The engine bulkhead shown was analyzed for stresses around some internal bolt holes (not visible in the figure). A fine mesh was required in that region to handle the stress gradients involved. In addition, a large part of the engine bulkhead structure had to be modeled so that it would react properly to the crankshaft loads applied to it. This type of situation, namely, an area of very fine mesh needed somewhere in a large solid model, would normally lead to a very difficult mesh transitioning job for the modeler and the transition area would probably include many badly shaped elements. To avoid that, the RBE3 method was used to "glue" the fine mesh to the coarse mesh. The result was that the time to build the model was greatly reduced and very good results were obtained.

CONCLUSION

A method has been introduced that allows the finite element analyst to connect two solid models with dissimilar meshes on the mating surfaces. The method facilitates transitioning from one mesh to another. This mesh transitioning may be needed to change from fine to coarse mesh, or may be required between two areas where geometries call for different meshes. The method also allows two modelers to work independently on separate areas of a structure, without the requirement that the meshes match up at the boundary between the two areas.

The method has been found to yield strain-free rigid body motion, as required. Use of the method on simple stretching, bending, and twisting test cases leads to no degradation of results. Also, the method has been used in project work on a very large analysis, with good success.

REFERENCES

1. Cook, R. D., Concepts and Applications of Finite Element Analysis, 2nd Ed., John Wiley & Sons, New York, 1981, pp. 113–119.

2. Dahlquist, G., and Bjorck, A., Numerical Methods, Prentice Hall, Englewood Cliffs, N.J., 1969, pp. 249–251.

APPENDIX

Iterative Solution to Obtain Weight Factors for the Case where Grid Point A Falls Inside a Quadrilateral

As shown in the theory section, the four equations that determine the four weight factors for a given location of point A inside a quadrilateral are:

$$x_1 W_1 + x_2 W_2 + x_3 W_3 + x_4 W_4 = x_A \qquad (1)$$

$$y_1 W_1 + y_2 W_2 + y_3 W_3 + y_4 W_4 = y_A \qquad (2)$$

$$W_1 + W_2 + W_3 + W_4 = 1 \qquad (3)$$

$$W_1 W_3 - W_2 W_4 = 0 \qquad (4)$$

Equation (4) makes the set of equations nonlinear because it involves products of the variables. A nonlinear set of equations requires an iterative solution. There are many iterative methods available. The Newton-Raphson Method [2] is probably the most popular and is used here.

We rewrite equations (1) — (4) as:

$$f_1 = x_1 W_1 + x_2 W_2 + x_3 W_3 + x_4 W_4 - x_A \quad (1)$$

$$f_2 = y_1 W_1 + y_2 W_2 + y_3 W_3 + y_4 W_4 - y_A \quad (2)$$

$$f_3 = W_1 + W_2 + W_3 + W_4 - 1 \quad (3)$$

$$f_4 = W_1 W_3 - W_2 W_4 \quad (4)$$

The equations are satisfied if $f_1 = f_2 = f_3 = f_4 = 0$, or in matrix notation:

$$\{f\} = 0$$

Expanding each f_i into a Taylor series, dropping higher order terms, and writing as an iterative equation:

$$\{f\} = \{f\}^k + [f']^k \left(\{W\}^{k+1} - \{W\}^k\right) = 0$$

The superscripts k and k+1 are iteration numbers and $\{W\}$ is a vector of the variables W_1, W_2, W_3, W_4. Matrix $[f']$ is a 4 x 4 matrix of partial derivatives:

$$[f'] = \frac{\partial f_i}{\partial W_j} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \\ W_3 & -W_4 & W_1 & -W_2 \end{bmatrix}$$

Then, at each iteration, the linear set of equations to be solved is:

$$[f']^k \{W\}^{k+1} = [f']^k \{W\}^k - \{f\}^k$$

The process begins with initial guesses of $W_1 = W_2 = W_3 = W_4 = 1/4$ and continues until $\{f\} = 0$ and $\{w\}^{k+1} = \{w\}^{k}$. As long as point A is inside the quadrilateral the iteration always converges and converges quickly. It takes at most five iterations even when point A is on or near the boundary of the quadrilateral.

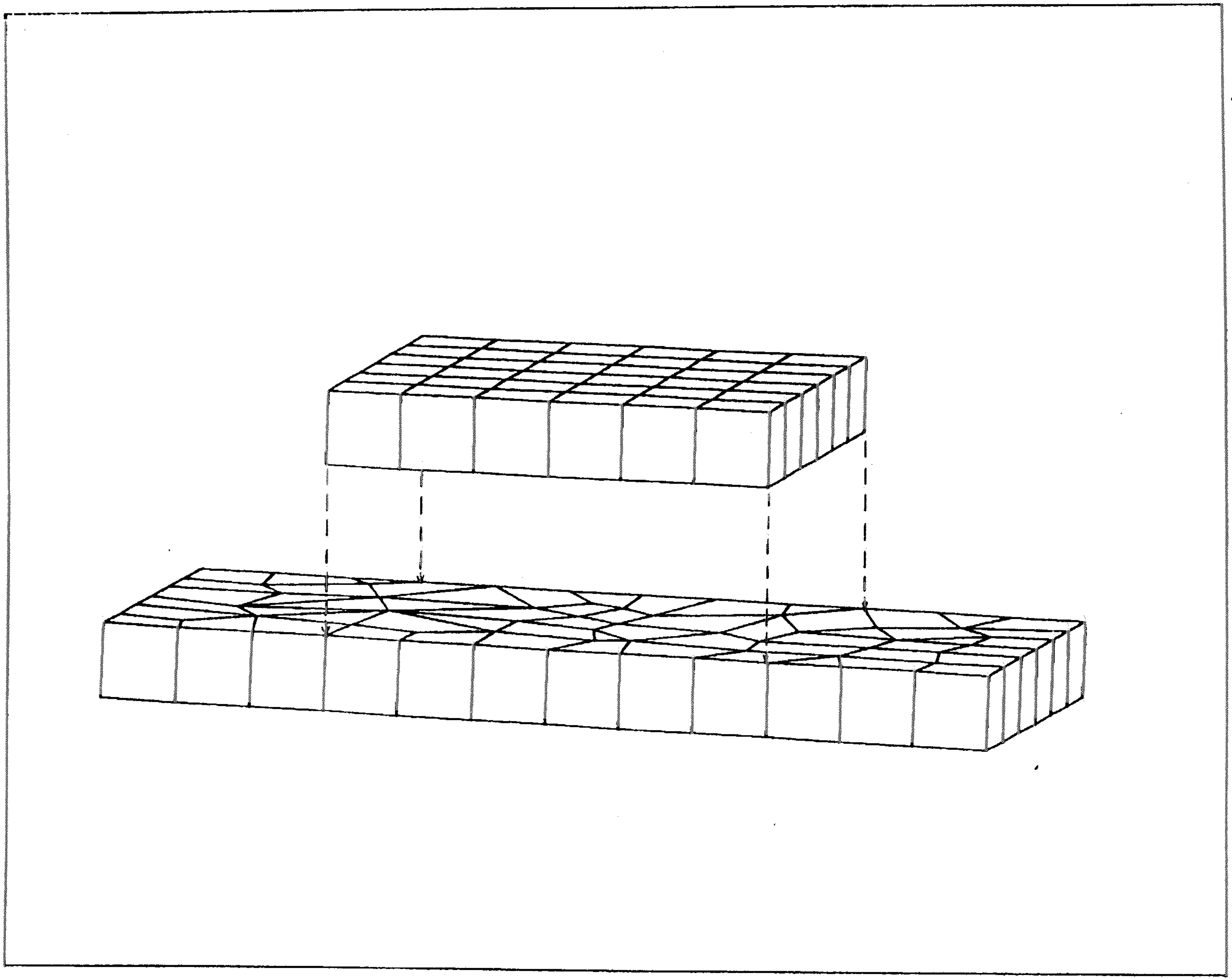
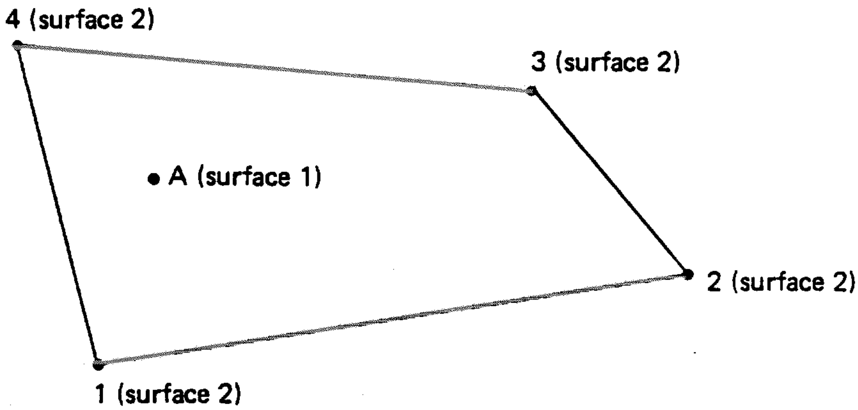
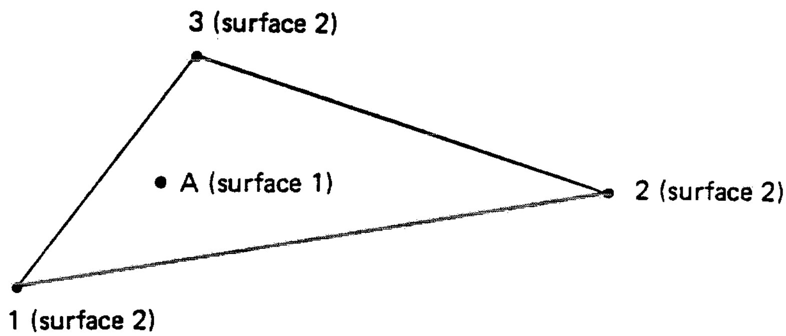


Figure 1
Two Structures with Dissimilar Meshes to be Connected



(a) Grid Point A Inside a Quadrilateral



(b) Grid Point A Inside a Triangle

Figure 2
Grid Point A and the Surrounding Three or Four Grid Points

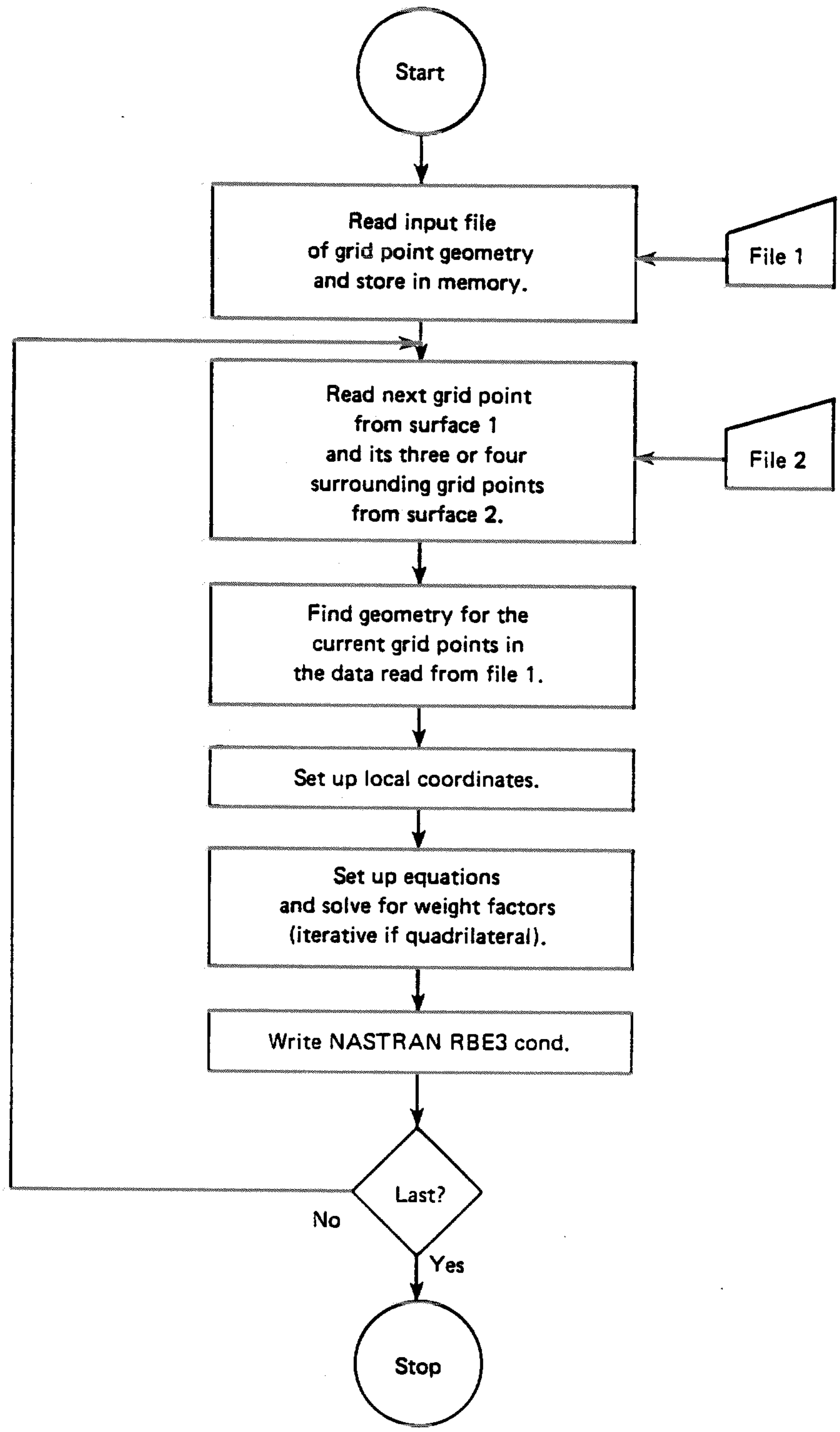


Figure 3
Flow Chart of FORTRAN
Program Used to Compute Weight
Factors and Write RBE3 Cards

TOP PLATE UNIFORM, BOTTOM PLATE IRREGULAR

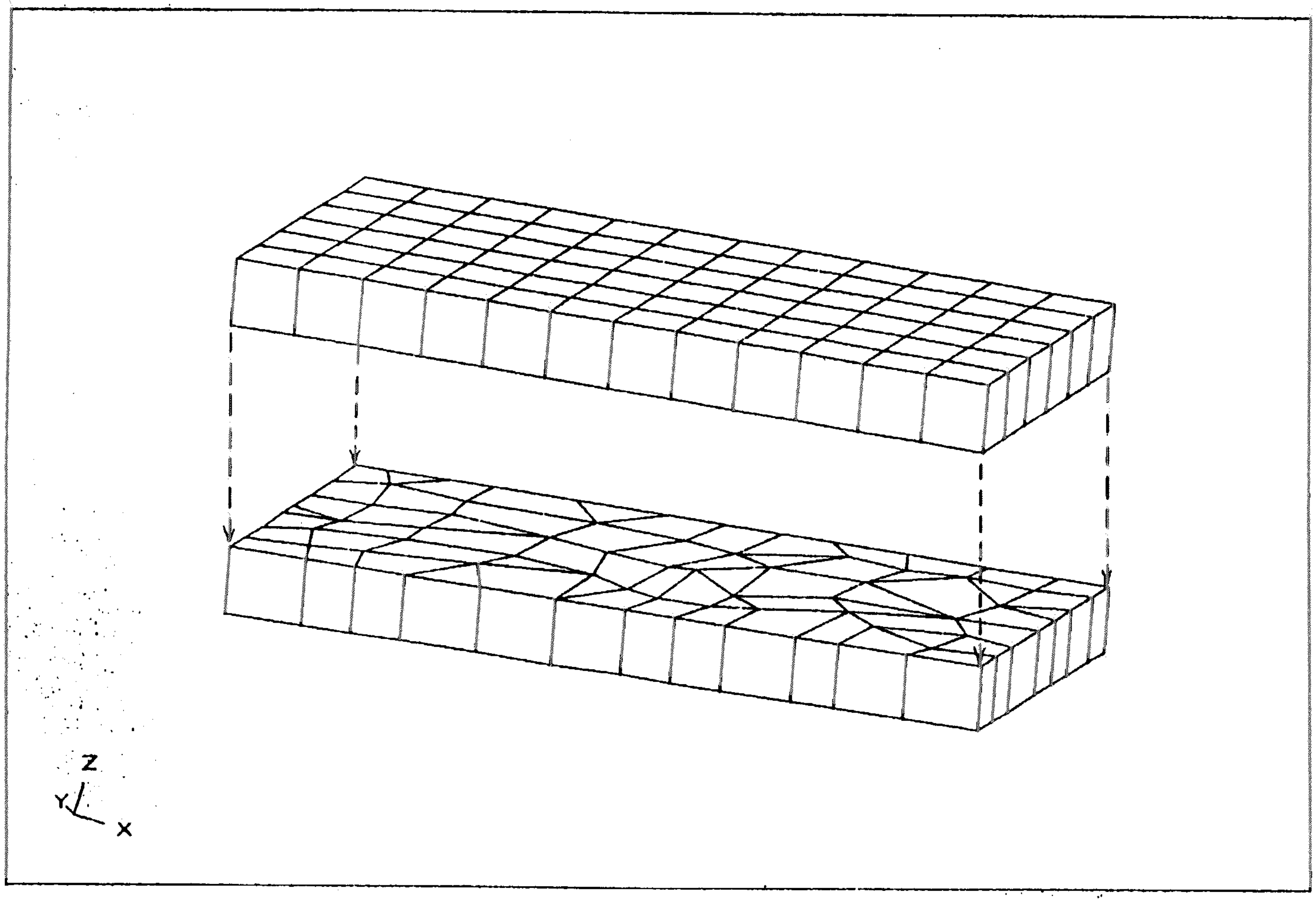
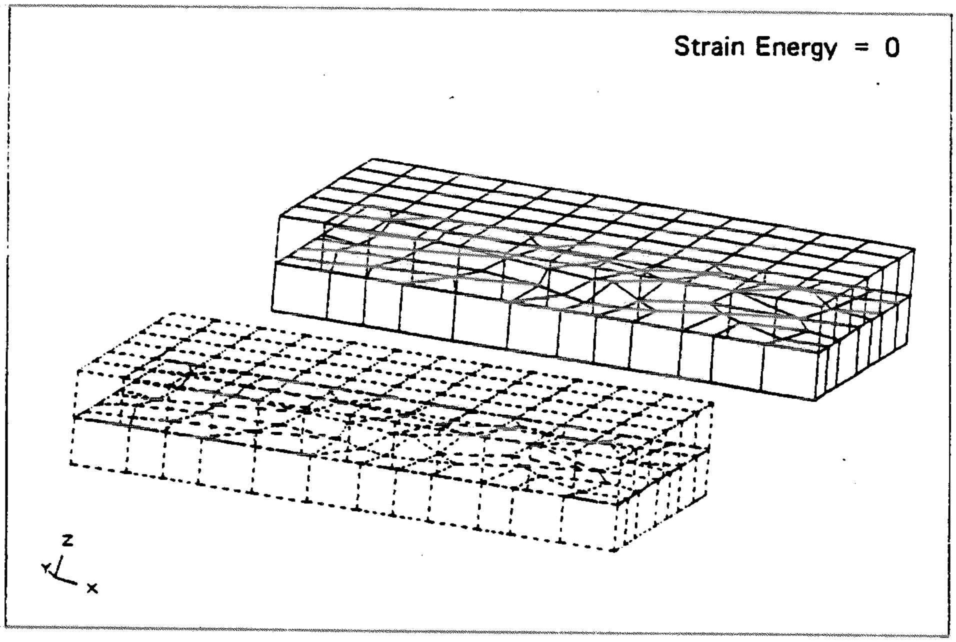
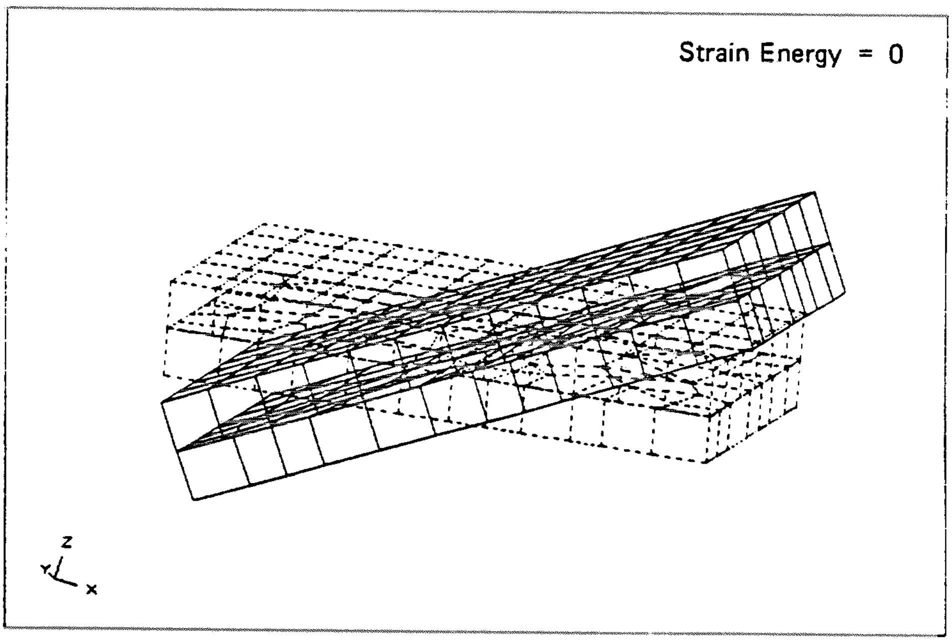


Figure 4
Simple Example Problem
Two Thick Plates with Dissimilar Meshes
Connected by RBE3 Method

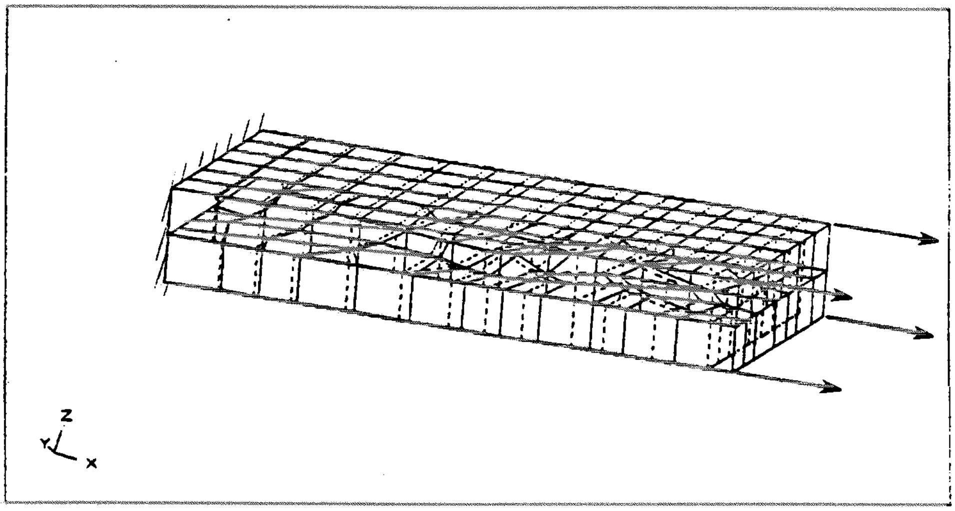


(a) Translation

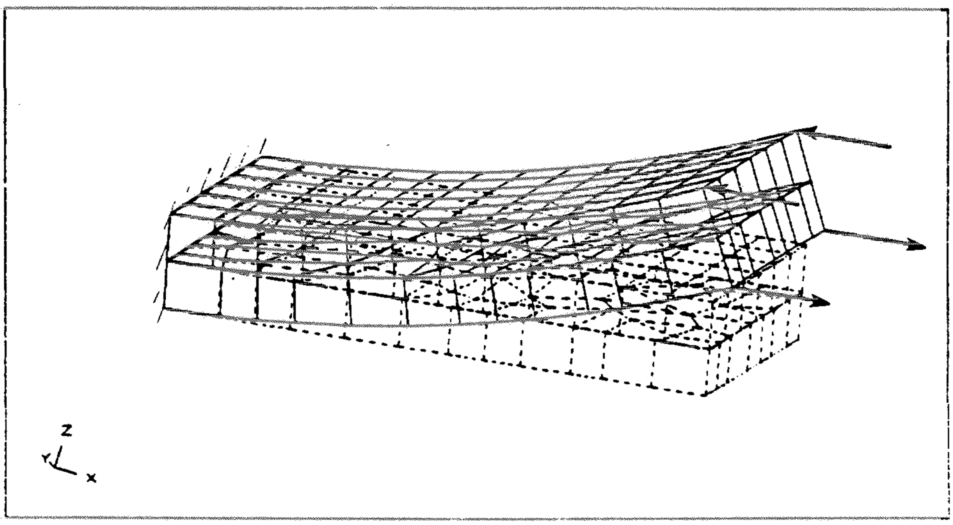


(b) Rotation

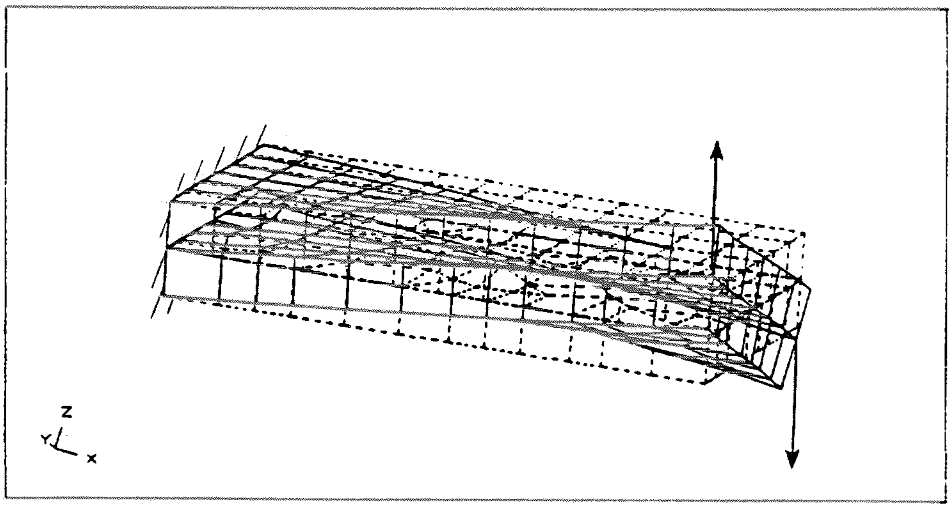
Figure 5
Rigid Body Motions for Simple Example Problem



(a) Axial Load



(b) Bending



(c) Twist

Figure 6
Elastic Deformations for Simple Example Problem

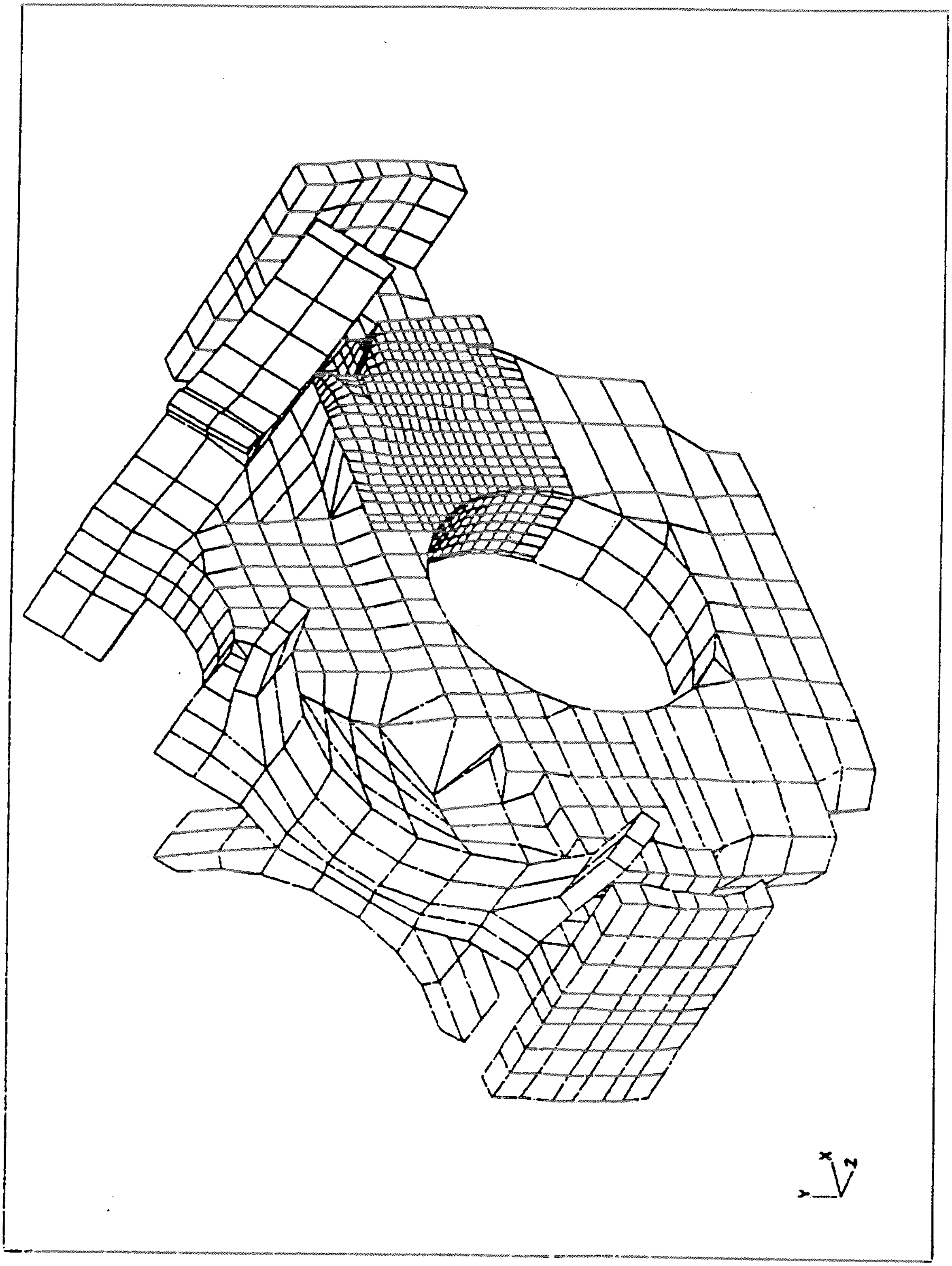


Figure 7
Model of Engine Bulkhead Showing Use of
RBE3 Method to Transition from Fine to Coarse Mesh