URTIC

OF TH

DCUM

ARE

LEGI

LA-UR -82-1000

MASTER

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405 ENG-36

TITLE A MAPPING METHOD FOR GENERATING THREE-DIMENSIONAL MESHES: PAST AND PRESENT

AUTHOR(S) W. A. Cook and W. R. Oakes

SUBMITTED TO ASME Second International Computer Engineering Conference and Show, August 15-19, 1982, San Diego, California

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

ABSTRACT

Two transformations are derived in this paper. One is a mapping of a unit square onto a surface and the other is a mapping of a unit cube onto a three-dimensional region. Two meshing computer programs are then discussed that use these mappings. The first is INGEN, which has been used to calculate three-dimensional meshes for approximately 15 years. This meshing program uses an index scheme to number boundaries, surfaces, and regions. With such an index scheme, it is possible to control nodal points, elements, and boundary conditions. The second is ESCHER, a meshing program now being developed. Two primary considerations governing development of ESCHER are that meshes graded using quadrilaterals are required and that edge-line geometry defined by Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM) systems will be a major source of geometry definition. This program separates the processes of nodal-point connectivity generation, computation of nodal-point mapping space coordinates, and mapping of nodal points into model space.

INTRODUCTION

The purpose of this paper is to discuss mapping methods used to generate three-dimensional meshes. Mapping methods have been used by researchers for more than a decade to generate three-dimensional meshes (1)-(3).

The fundamental concepts of a meshing computer program that uses mapping methods are the surface and three-dimensional region mapping transformations. These are derived and discussed in the Generators section as the first author remembers their development during the years 1967 through 1972 (see the Appendix). This is accomplished by defining a body or normal coordinate system, then deriving two- and three-dimensional coordinate transformations between the body coordinate system and the cartesian coordinate system. This establishes a mapping of a unit square onto a surface and a unit cube onto a three-dimensional region.

The INGEN Meshing Computer Program section describes the techniques used to generate three-dimensional meshes with the INGEN computer program. INGEN and its predecessor have been generating three-dimensional meshes for the past 15 years.

Two meshes are shown that are generated with INGEN and displayed with GRAPE (4). To use any meshing computer program effectively for generating three-dimensional meshes, a graphics computer program is required to display the mesh. The ones used at the Los Alamos National Laboratory are MOVIE developed by Henry Christiansen of Brigham Young University and GRAPE developed by Bruce Brown of Lawrence Livermore National Laboratory (GRAPE is a modified version of MOVIE).

The ESCHER Meshing Computer Program section describes the techniques used to generate three-dimensional meshes with the ESCHER computer program. This is a new computer program under development at Los Alamos. This computer program also uses mapping methods and has its own graphics.

The following terms will be used in this paper.

Grid point - a calculated point that defines geometry.

Nodal point - a geometric point that defines an element in the finite-element method. Nodal points are defined using grid points.

Boundary edge - a line that defines the boundary of a surface. This need not be a discontinuous line.

GENERATORS

In this section, body coordinates are defined, and a transformation is derived for both a surface and a three-dimensional region.

Consider the two-dimensional region shown in Fig. 1. This region can be defined by two coordinate systems. One is the cartesian coordinate system and the other is a nonorthogonal coordinate system, called body or normal coordinates (ξ,η). The body or normal coordinates are normalized such that they vary between 0 and 1. Thus, when ξ or η is either 0 or 1, a boundary of the region is designated. A coordinate transformation between these two coordinate systems $x(\xi,\eta)$ and $y(\xi,\eta)$ can be defined as follows.

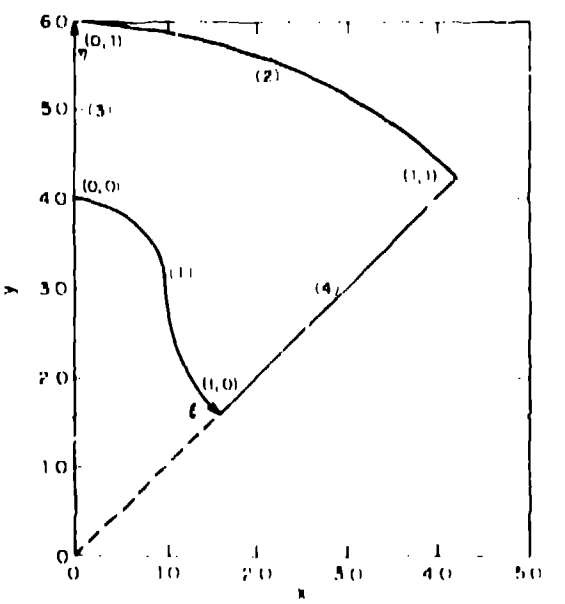

Fig. 1 Cartesian and body coordinate systems for a two-dimensional region having two straight boundary edges

- Four boundaries are designated for a region. Two opposite boundaries must be straight (Boundaries 3 and 4 in Fig. 1) and the remaining two may be curved (Boundaries 1 and 2 in Fig. 1). The curved boundaries are designated as $x(\xi,0)$, $y(\xi,0)$ for Boundary 1 and $x(\xi,1)$, $y(\xi,1)$ for Boundary 2.
- A linear transformation is defined as a transformation that defines η equal to a constant within the region as a linear combination of the two curved boundaries. Thus, constant values of η will evolve from the shape of one curved boundary to the shape of the other curved boundary as η varies from 0 to 1.

This linear transformation is

$$x(\xi,\eta) = (1-\eta)\,x(\xi,0) + \eta\,x(\xi,1) \text{ and}$$
$$y(\xi,\eta) = (1-\eta)\,y(\xi,0) + \eta\,y(\xi,1) \quad . \tag{1}$$

Adding the z coordinate $z(\xi,\eta)$ makes the transformation valid for a three-dimensional surface. Thus,

$$z(\xi,\eta) = (1-\eta)\,z(\xi,0) + \eta\,z(\xi,1) \quad . \tag{2}$$

Using this transformation for a surface generator, a computer program was developed for generating three-dimensional meshes at Thiokol Corporation in 1967 (see the Appendix).

A mesh was calculated for the region shown in Fig. 1, and this mesh is shown in Fig. 2 ($\Delta\xi = 1/12$ and $\Delta\eta = 1/10$).

In 1969, we wanted to generalize this tranformation. Thus, the criteria we adopted were as follows.

- All four boundaries can be curved, and all four boundaries will be equally represented in the transformation.
- If any two opposite sides are straight, the transformation will reduce to the transformation represented by Eqs. (1) and (2).
- For constant ξ or constant η, the interior shapes will be a combination of the boundaries. Thus, the interior points for ξ constant will be a combination of the boundaries $x(0,\eta)$, $y(0,\eta)$, $z(0,\eta)$, and $x(1,\eta)$, $y(1,\eta)$, and $z(1,\eta)$. For η constant, the interior points will be a combination of the boundaries $x(\xi,0)$, $y(\xi,0)$, $z(\xi,0)$ and $x(\xi,1)$, $y(\xi,1)$, $z(\xi,1)$. When either ξ or η is constant, the interior points will evolve from the shape of one boundary to the shape of the opposite boundary.

Using the first transformation (Eq. (1)) and Fig. 3, the transformation can be derived for four curved boundaries as

$$x(\xi,\eta) = (1-\eta)\,x(\xi,0) + \eta\,x(\xi,1) - (1-\xi)\,\Delta x(0,\eta) + \xi\,\Delta x(1,\eta) \quad ,$$

$$y(\xi,\eta) = (1-\eta)\,y(\xi,0) + \eta\,y(\xi,1) + (1-\xi)\,\Delta y(0,\eta) - \xi\,\Delta y(1,\eta) \quad ,$$

where

$$\begin{aligned}
\Delta x(0,\eta) &= [(1-\eta)\,x(0,0) + \eta\,x(0,1)] - x(0,\eta) \quad , \\
\Delta y(0,\eta) &= y(0,\eta) - [(1-\eta)\,y(0,0) + \eta\,y(0,1)] \quad , \\
\Delta x(1,\eta) &= x(1,\eta) - [(1-\eta)\,x(1,0) + \eta\,x(1,1)] \quad , \text{ and} \\
\Delta y(1,\eta) &= [(1-\eta)\,y(1,0) + \eta\,y(1,1)] - y(1,\eta) \quad .
\end{aligned} \tag{3}$$

Simplifying

$$\begin{aligned}
x(\xi,\eta) = {} & (1-\eta)\,x(\xi,0) + \eta\,x(\xi,1) + (1-\xi)\,x(0,\eta) \\
& + \xi\,x(1,\eta) - (1-\xi)(1-\eta)\,x(0,0) \\
& - (1-\xi)\eta\,x(0,1) - \xi(1-\eta)\,x(1,0) \\
& - \xi\eta\,x(1,1) \quad ,
\end{aligned} \tag{4}$$

$$\begin{aligned}
y(\xi,\eta) = {} & (1-\eta)\,y(\xi,0) + \eta\,y(\xi,1) + (1-\xi)\,y(0,\eta) \\
& + \xi\,y(1,\eta) - (1-\xi)(1-\eta)\,y(0,0) \\
& - (1-\xi)\eta\,y(0,1) - \xi(1-\eta)\,y(1,0) \\
& - \xi\eta\,y(1,1) \quad .
\end{aligned}$$

Again, when

$$\begin{aligned}
z(\xi,\eta) = {} & (1-\eta)\,z(\xi,0) + \eta\,z(\xi,1) + (1-\xi)\,z(0,\eta) \\
& + \xi\,z(1,\eta) - (1-\xi)(1-\eta)\,z(0,0) \\
& - (1-\xi)\eta\,z(0,1) - \xi(1-\eta)\,z(1,0) \\
& - \xi\eta\,z(1,1) \quad ,
\end{aligned} \tag{5}$$

these transformation equations can be applied to a three-dimensional surface. Notice that these transformation equations satisfy the criteria we adopted for a generalized surface transformation.

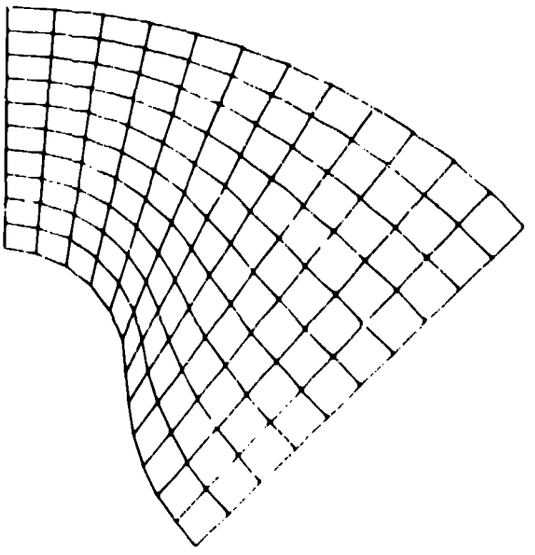

Fig. 2 A two-dimensional mesh of Fig. 1

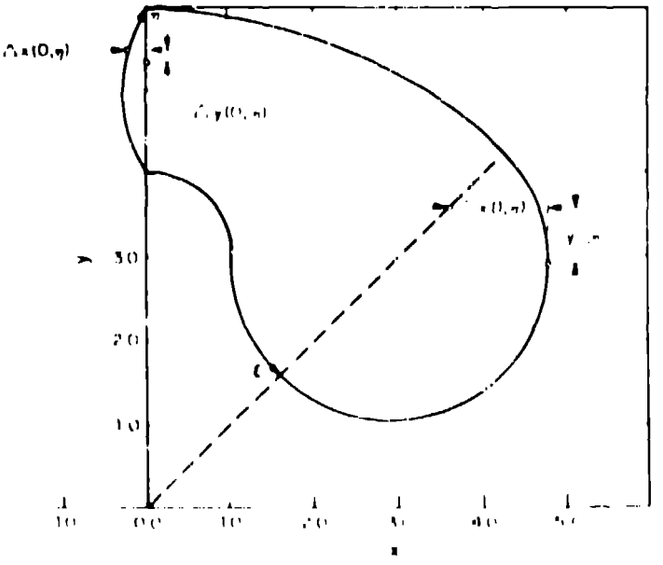

Fig. 3 Cartesian and body coordinate system for a two-dimensional region

A mesh was calculated for the region in Fig. 3 and this mesh is shown in Fig. 4 ($\Delta\xi$ = 1/12 and $\Delta\eta$ = 1/14).

Given the grid points on the boundary edges, the body coordinates (ξ,η) can be calculated. This is accomplished by calculating the length of the lines between grid points and then normalizing with respect to the total boundary edge length. Thus, ξ and η vary between 0 and 1 on each boundary edge. This can be considered a mapping of a unit square onto a two-dimensional surface.

Consider the three-dimensional region shown in Fig. 5. Again, this region can be defined by both the cartesian coordinate system and the body coordinate system (ξ,η,γ). A linear coordinate transformation between these two coordinate systems ($x(\xi,\eta,\gamma)$, $y(\xi,\eta,\gamma)$, and $z(\xi,\eta,\gamma)$) can be defined using the following criteria.

- All 12 boundary edges can be curved and each is equally represented.
- The three-dimensional transformation reduces to surface transformations similar to Eqs. (4) and (5) when any one of the body coordinates is either 0 or 1.

Thus, a linear transformation that satisfies these criteria is

$$\begin{aligned} x(\xi,\eta,\gamma) = {} & (1-\eta)(1-\gamma)\,x(\xi,0,0) \\ & + \eta(1-\gamma)\,x(\xi,1,0) + (1-\eta)\gamma\,x(\xi,0,1) \\ & + \eta\gamma\,x(\xi,1,1) + (1-\xi)(1-\gamma)\,x(0,\eta,0) \\ & + \xi(1-\gamma)\,x(1,\eta,0) + (1-\xi)\gamma\,x(0,\eta,1) \\ & + \xi\gamma\,x(1,\eta,1) + (1-\xi)(1-\eta)\,x(0,0,\gamma) \\ & + \xi(1-\eta)\,x(1,0,\gamma) + (1-\xi)\eta\,x(0,1,\gamma) \\ & + \xi\eta\,x(1,1,\gamma) + C(\xi,\eta,\gamma) \quad , \end{aligned}$$

where

$$\begin{aligned} C(\xi,\eta,\gamma) = {} & -2(1-\xi)(1-\eta)(1-\gamma)\,x(0,0,0) \\ & -2\xi(1-\eta)(1-\gamma)\,x(1,0,0) \\ & -2(1-\xi)\eta(1-\gamma)\,x(0,1,0) \\ & -2\xi\eta(1-\gamma)\,x(1,1,0) \\ & -2(1-\xi)(1-\eta)\gamma\,x(0,0,1) \\ & -2\xi(1-\eta)\gamma\,x(1,0,1) \\ & -2(1-\xi)\eta\gamma\,x(0,1,1) \\ & -2\xi\eta\gamma\,x(1,1,1) \quad . \end{aligned} \tag{6}$$

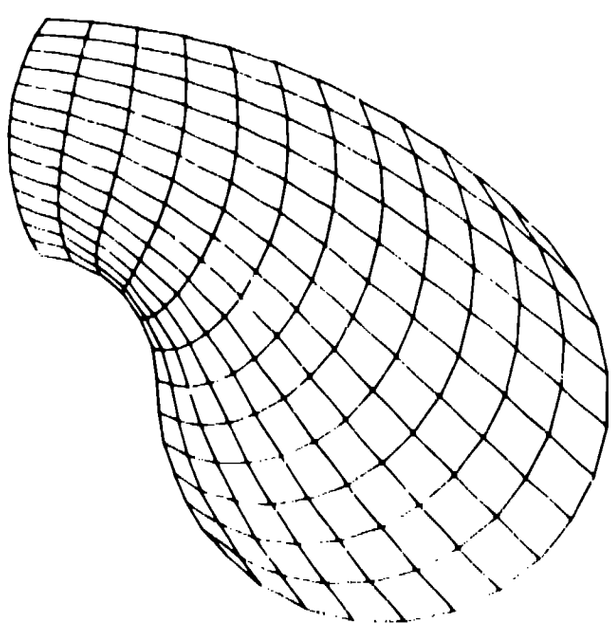

Fig. 4 A two-dimensional mesh of Fig. 3

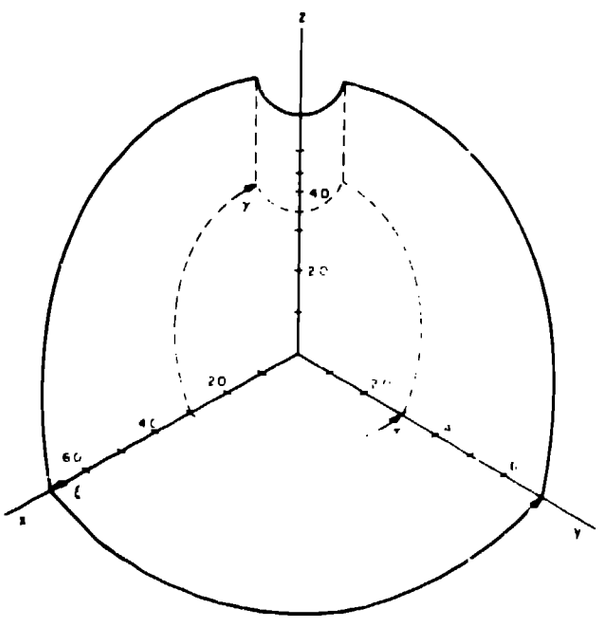

Fig. 5 Cartesian and body coordinate systems for a three-dimensional region

Also, $y(\xi,\eta,\gamma)$ and $z(\xi,\eta,\gamma)$ are similar. This transformation can be written in terms of surfaces as follows:

$$\begin{aligned} x(\xi,\eta,\gamma) = {} & 1/2\,[(1-\eta)\,x(\xi,0,\gamma) + \eta\,x(\xi,1,\eta) \\ & + (1-\xi)\,x(0,\eta,\gamma) + \xi\,x(1,\eta,\gamma) \\ & + (1-\gamma)\,x(\xi,\eta,0) + \gamma\,x(\xi,\eta,1)] \\ & + 1/4\,C(\xi,\eta,\gamma) \quad , \end{aligned} \tag{7}$$

where $y(\xi,\eta,\gamma)$ and $z(\xi,\eta,\gamma)$ are similar.

Figure 6 shows a mesh that was calculated for the three-dimensional region shown in Fig. 5 ($\Delta\xi$ = 1/10, $\Delta\eta$ = 1/10, and $\Delta\gamma$ = 1/10).

This transformation, Eq. (7), can be considered as a mapping from a unit cube just as the surface transformation was a mapping from a unit square.

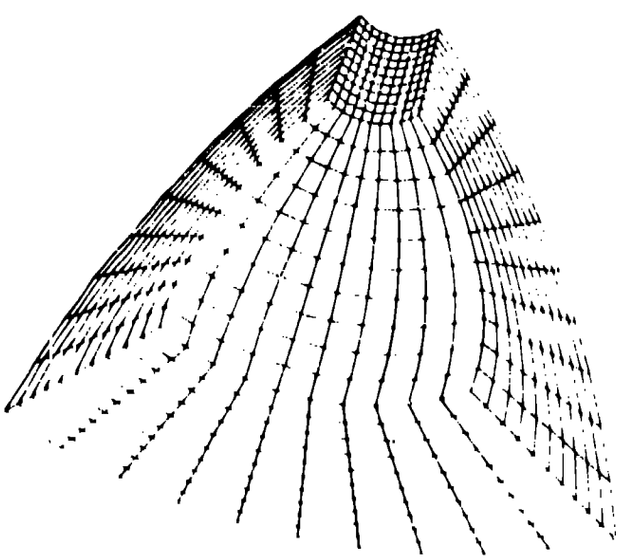

Fig. 6 A three-dimensional mesh of Fig. 5

In 1973, W. J. Gordon of General Motors Research Laboratories informed the first author that these transformation equations had been derived by S. A. Coons several years earlier (5).

We rely on user intervention to correct mapping errors rather than insist on perfect mapping algorithms. Occasionally, the mapping of a part into the real space will have undesired results. Some grid points may be generated outside the intended part boundaries. The user corrects such a problem by either redefining the region of error by subdividing the region into simpler regions or by introducing auxiliary constraints on the mapping as suggested by Gordon and Hall (2). For an example, see Fig. 7.

THE INGEN MESHING COMPUTER PROGRAM

INGEN (6) and its predecessor have been in existence for 15 years. See the Appendix for a history of this computer program. In this section, we discuss the philosophy used to develop this mesher, describe the methods used for calculating body coordinates on a unit square and in a unit cube, and define an index scheme and discuss its use. Also, two meshes are shown.

The philosophy involved in developing a nodal-point mesh is to distribute the grid points according to the anticipated variation of the field variables. For large gradients, the grid points should be dense, and for small gradients, the grid points should be sparse. Therefore, when generating a mesh, it is necessary to make an estimate of how the field variables change in the different regions of the problem. The user of the INGEN computer program establishes mesh grading for his gradient estimate by generating the boundary edges of the mesh with the desired spacing of grid points using the boundary-edge generators, and then by using surface and volume (three-dimensional region) generators, both of which preserve this spacing. The surface-grid-point generator preserves this spacing by using the grid points as they are distributed along the boundary edges as the criteria for spacing the surface grid points. Similarly, the volume-grid-point generator calculates the interior grid points using the surface grid points as the criteria for spacing the interior grid points. Using these generated grid points, the desired nodal points and elements are then calculated.

The unit square concept in the Generators section is very useful in determining ξ and η for grid points on the surface. The approach that was used in (1) was to draw straight lines between opposite boundaries of the unit square, and the point of intersection established a ξ and η for a grid point (see Fig. 8). This approach requires the limitation of having the same number of grid points on opposite boundaries. For details, see (1). Once the surface body coordinates are defined, the surface grid points are calculated with transformation Eqs. (4) and (5). The unit cube concept in the Generators section is very useful to find coordinates ξ,η,γ for calculating interior grid points for a three-dimensional region. The approach that was used in (1) was to draw a straight line between opposite boundaries of the unit cube and the point that was the minimum distance to the three lines establishes a ξ,η,γ for a grid point (see Fig. 9). With the body coordinates defined, the interior grid points are calculated with Eq. (7).

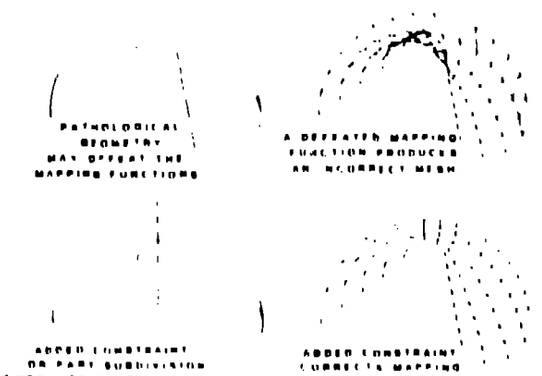

Fig. 7 User intervention corrects mapping errors

The index scheme is an integral part of the INGEN meshing computer program. If i index counts the grid points in the ξ coordinate direction, j index counts those in the η coordinate direction, and k index counts those in the γ coordinate direction. Then when only one of the indexes changes, a boundary edge is designated. When two of the indexes change, a surface is represented, and when all three indexes change, a three-dimensional region is designated. This makes it very easy to program problem constraints or boundary conditions that are needed along a boundary edge, on a surface, or throughout a three-dimensional region. Also, with this index scheme, it is possible to check to see if the input is complete for the boundary edge generators, surface generators, and volume generators. Another use of this index scheme is to decide when to form nodal points from grid points. For example, consider the three-dimensional parabolic element.

- When all three indexes are odd, the grid point is a corner nodal point.
- When two indexes are even and one index is odd, there is no nodal point at that point.
- When all three indexes are even, the grid point is a center point and may or may not be a nodal point as desired.
- When two indexes are odd and one even, the grid point is a midside nodal point and may be eliminated or centered as desired.
- Depending upon which index (i, j, or k) is the even index, the midside nodal points may be eliminated on a preferential basis.

Two meshes will be shown that have been generated using the INGEN meshing computer program and displayed with the graphics computer program GRAPE.

The first of these meshes is a 30-deg segment of a prestressed concrete reactor vessel designed by General Atomic of La Jolla, California, a few years ago. This mesh was generated with INGEN and is shown in Fig. 10. The horizontal and vertical piping are of special interest in this model. This model was analyzed for stress distribution using the NONSAP-C computer program (7).

Another mesh generated with INGEN is a three-dimensional model of a controlled thermonuclear reactor experiment. This experiment is being studied at Los Alamos and is shown in Fig. 11. This model was analyzed for magnetic field distribution using the NONSAP computer program (8). Of special interest is the geometric spacing of the nodal points in this mesh.

THE ESCHER MESHING COMPUTER PROGRAM

In this section, we discuss the use of body coordinate transformation functions in the mesh generator ESCHER. The choice of using a mapping method to develop ESCHER followed an analysis of our current and future meshing needs. It is therefore beneficial to discuss the concepts controlling the development of ESCHER that led to the choice of a mapping

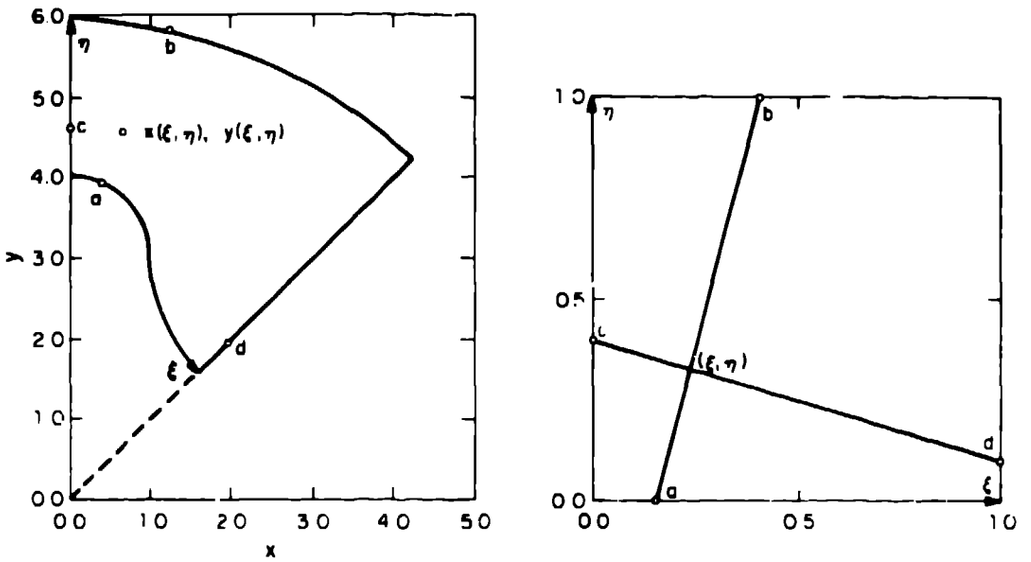

Fig. 8 Body coordinate system mapped on a unit square

a – Point on the $\eta = 0$ surface
b – Point on the $\eta = 1$ surface
c – Point on the $\gamma = 0$ surface
d – Point on the $\gamma = 1$ surface
e – Point on the $\xi = 0$ surface
f – Point on the $\xi = 1$ surface

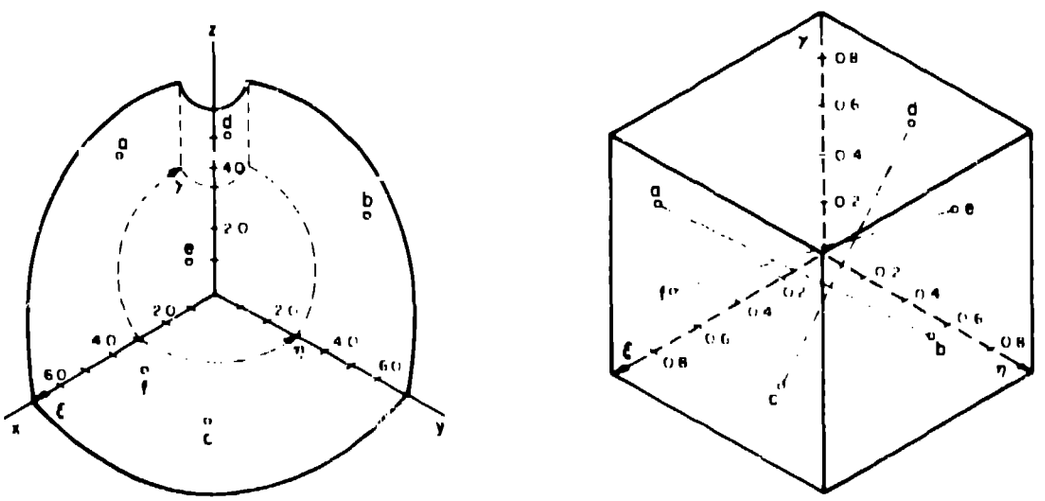

Fig. 9 Body coordinate system mapped on a unit cube

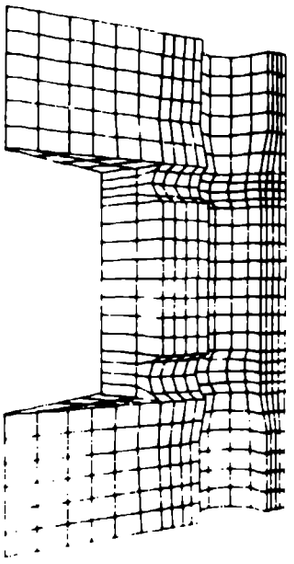

Fig. 10 Prestressed concrete reactor vessel (30-deg segment) mesh (INGEN)

method. We emphasize our philosophy of edge-surface geometry specification and its relationship to our application. We present examples of new mesh grading methods that rely on body coordinate transformation functions.

ESCHER is an interactive computer program for generating two- and three-dimensional meshes from geometry data received from CAD/CAM systems, other geometry input files, and interactive input. Although the interface system is still under development, it is our intent to use the Initial Graphics Exchange Specification (IGES) (9) file format to transmit many of the required geometry data to ESCHER. We have chosen the IGES format because we believe that it will be supported by most of the CAD/CAM systems that we will interface to for the next 7 to 10 years. Specifically, we expect CAD/CAM geometry to be expressed as edge lines during this time. A second design objective for ESCHER is to provide more flexible mesh grading techniques than those available in the mesh generators that we now use. This is required so that we can better distribute grid points according to anticipated variation of field variables. We require that a nonambiguous meshing be established even for regions in which different numbers of elements are specified along region boundaries. The anticipation of receiving edge-line geometry specification from CAD/CAM systems and the ability to define flexible grading techniques influenced us in choosing a mapping method for ESCHER.

The first design criterion that resulted in the choice of a mapping method for ESCHER is that geometry will be specified as edge lines. The edge lines come from CAD/CAM systems, from geometry input files, and from interactive input. Mapping methods, which require edge geometry to generate region geometry, are the logical choice for geometry completion under these conditions. However, there is a fundamental limitation to geometry specification with edge-defined geometry. In reality, an edge is the intersection of surfaces; a surface is not an intersection of edges. In practice, this limitation has not been severely restrictive. The user defines a part by specifying the edges surrounding the part and by specifying the number of elements required along each edge. We believe that automation is required for this step so that the users' input will be minimized. For a given topology, there is a prescribed order in which edges must be specified. Figure 12 presents the patterns that can be used to combine edges into parts. These patterns are reduced to quadrilaterals and cuboids before the mapping method is used to mesh the part. The special purpose 3-D volumes have not yet been developed. ESCHER creates an internal description of a part by computing vertex points at the intersections of the specified edge lines. Edges are then defined to be parametric equations of the edge lines defined between the vertex points (the parameter being normalized arc length). The vertex locations thus become corners of a unit square or a unit cube for the mapping function to be used for internal part geometry definition. This parameterization, along with the mapping transformations specified in the Generators section, provides a complete description of the edge-defined part.

The second design criterion that resulted in the choice of a mapping method for ESCHER is that more flexible mesh grading techniques than are now available to us are required. Our implementation of mesh grading is inspired by (and resembles somewhat) a technique by Imafuku, Kodera, and Sayawaki (10). The difference from their technique is caused by our desire to grade with quadrilaterals. We follow a three-step procedure in generating a graded mesh: (1) we generate node connectivity, (2) we compute nodal coordinates in the mapping space, and (3) we map the nodal coordinates into the model space.

Flexibility of mesh grading within ESCHER results from the separation of nodal connectivity generation into a distinct step. We can easily insert new grading algorithms based on nodal connectivity. It is an exercise in induction to show

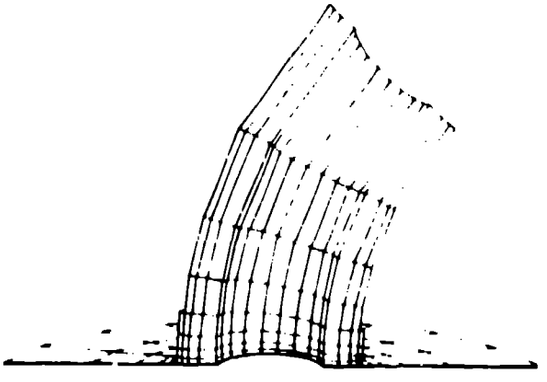

Fig. 11 Controlled thermonuclear reactor experiment mesh (INGEN)

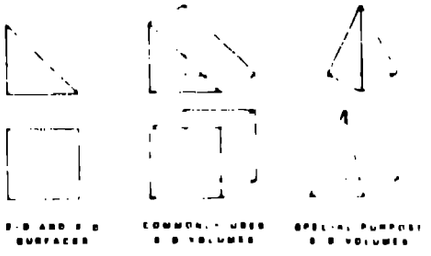

Fig. 12 ESCHER part topologies

that any region to be meshed with quadrilaterals must have an even number of element-edge to region-edge contacts. Within this restriction, there are a great number of useful mesh grading possibilities. Figure 13 presents a sample of the mesh grading capabilities that we have developed to date. Three-dimensional topologies considered so far are replications of the two-dimensional topologies into the third dimension. Figure 14 presents a typical graded mesh for a three-dimensional part. In generating these mesh connectivities, we maintain two topological attributes of the nodal points: corner-node and edge-node. We must know corner-node positions in mapping space and we must know to which edge an edge-node belongs. Additional information is not needed to compute nodal coordinates in the mapping space.

The second step in generating a mesh on a part is computing the nodal coordinates in the mapping space. We do this by relaxing the nodal coordinates in the mapping space so that each nodal position is the average of its neighbor nodal positions. Constraints on this process are that corner-nodes are not allowed to move and edge-nodes must remain on the associated edge.

The third step in generating the graded mesh is to map the nodal coordinates of the part from the mapping space into the model space. Direct implementations of the mappings of the Generators section are used without relying on surface generators for three-dimensional mapping.

We rely on interactivity to correct model geometry faults. Baumgart (11) has suggested a method for creating shapes by using what he calls Euler objects. He bases his construction on Euler's rule specifying the number of faces, edges, and vertices in a polyhedron. We have used some of his ideas for data management within the program, but have left the user as the final judge of geometric correctness and completeness.

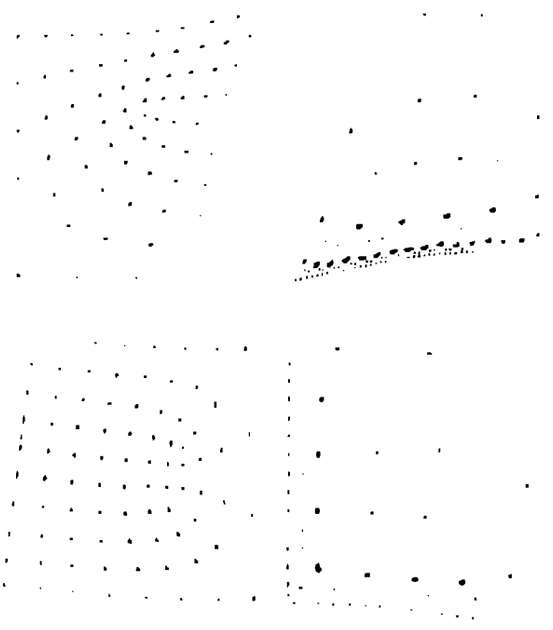

Fig. 13 Mesh grading algorithms offer gradual or rapid transition of element density

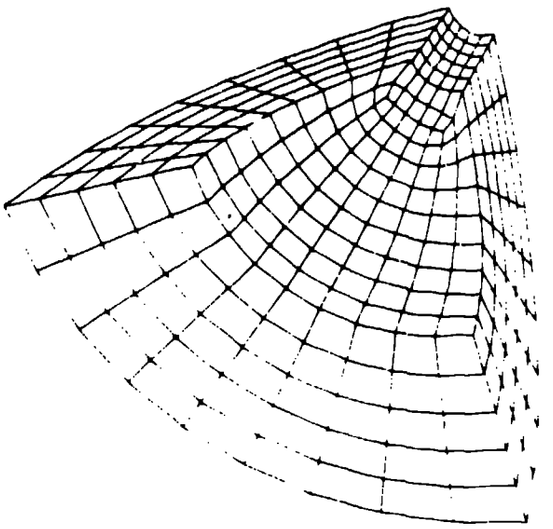

Fig. 14 A three-dimensional mesh using one-direction refinement

REFERENCES

1. Cook, W. A., "Body Oriented (Natural) Coordinates for Generating Three-Dimensional Meshes," Int. J. Num. Meth. Engng. 8, 27-43 (1974).

2. Gordon, W. J. and Hall, C. A., "Construction of Curvilinear Coordinate System and Application to Mesh Generation," Int. J. Num. Meth. Engng. 7, 461-477 (1973).

3. Barnhill, R. E., Birkhoff, G., and Gordon, W. J., "Smooth Interpolation in Triangles," J. Approx. Theory 8, 114-128 (1973).

4. Brown, B. E., "Displaying the Results of Three-Dimensional Analysis Using GRAPE," Lawrence Livermore National Laboratory report UCID-18067 (October 1979).

5. Coons, S. A., "Surface for Computer-Aided Design of Space Forms," Project MAC, MIT (1964). Revised to MAC-TR-41 (1967). Available through CFSTI, Sills Building, 5285 Port Royal Road, Springfield, VA 22151, USA.

6. Cook, W. A., "INGEN: A General-Purpose Mesh Generator for Finite Element Codes," Los Alamos Scientific Laboratory report LA-7139-MS (March 1978).

7. Anderson, C. A., Smith, P. D., and Carruthers, L. M., "NONSAP-C: A Nonlinear Stress Analysis Program for Concrete Containments Under Static, Dynamic, and Long-Term Loadings," NUREG-CR-0416, LA-7496-MS, Rev. 1 (January 1982).

8. Bathe, K. J., Wilson, E. L., and Iding, R. H., "NONSAP - A Structural Analysis Program for Static and Dynamic Response of Nonlinear Systems," Department of Civil Engineering, University of California report UCSESM 74-3 (1974).

9. Brauner, K., Bodnar, I., Fournier, L., Kelly, J. C., Kennicott, P., Liewald, M.,

Moore, D., and Wellington, J., "Final Draft Sections 1, 2, 3, and 4 Proposed American National Standard Engineering Drawing and Related Documentation Practices Digital Representation for Communication of Product Definition Data," IGES Y14.26M Response Committee (January 1982).

10. Imafuku, I., Kodera, Y., and Sayawaki, M., "A Generalized Automatic Mesh Generation Scheme for Finite-Element Method," Int. J. Num. Meth. Engng. 15, 713-731 (1980).

11. Baumhart, B. G., "Winged Edge Polyhedron Representation," Stanford University Press Computer Science Department, STAN-CS-320 (1972).

APPENDIX

HISTORY OF THE MESHING COMPUTER PROGRAM INGEN

The INGEN computer program had its origin in 1967 at Thiokol Corporation, Wasatch Division, Brigham City, Utah. D. H. Lee, a scientific programmer, and W. A. Cook, a mathematician, developed a three-dimensional meshing computer program for generating finite-element data for stress analysis of Minuteman rocket motors. This mesh had several limitations, and in 1970 the Air Force Rocket Propulsion Laboratory, Edwards Air Force Base, California, under Contract F04611-70-C-0068, financed the development of another meshing computer program. This new computer program was again developed by Lee and Cook. Later, E. C. Dickson, also a scientific programmer at Thiokol Corporation, added several significant features to this computer program. In 1974, Cook became a staff member at the Los Alamos National Laboratory and brought the Thiokol meshing computer program with him. V. J. Orlicky of the Los Alamos National Laboratory converted this meshing computer program to the CDC 7600 computer and named it INGEN. During the last eight years, several additions have been made to INGEN by Cook, P. D. Smith, and L. M. Carruthers. These have been supported by the US Nuclear Regulatory Commission, Division of Reactor Safety; Energy Research and Development Administration, (Contract W-7405-ENG.36); and Department of Energy, Division of Military Applications. The development during the last eight years has been a small effort, and in 1982, the computer program is only 4200 lines (including comments).