CONF-890741--1

SAND--88-3089C

DE89 005918

Using Conjoint Meshing Primitives To Generate Quadrilateral And Hexahedral Elements In Irregular Regions*

Michael B. Stephenson† Ted D. Blacker
Sandia National Laboratories

January 6, 1989

DISCLAIMER

*This work was performed at Sandia National Laboratories and supported by the U.S. Department of Energy under contract DE-AC04-76DP00789.

†On leave from Brigham Young University, Department of Civil Engineering during 1988-1989.

DISCLAIMER

Abstract

The parameter-space mapping technique used in most finite element mesh generation programs requires that the geometry be subdivided into either rectangular areas in 2-D or rectilinear volumes in 3-D to produce, respectively, quadrilateral or hexahedral elements. Since subdivision occurs in parameter space, the edges bounding the areas and volumes need not be straight or parallel lines. Simple geometries may be subdivided easily while irregular features challenge the analyst.

This paper presents the formulation and application of *conjoint* meshing primitives that assist the analyst in meshing in and around irregular features. The term *conjoint* is applied to these primitives because each is composed of several rectangular areas or rectilinear volumes. Interval assignments may vary from side to side, within a set of constraints, for each primitive. Thus, all of them are useful for transitions between other regular areas or volumes.

In 2-D, the primitive areas are the triangle, pentagon, semi-circle, circle, and rectangular transition. These primitives are implemented in the Sandia meshing program **FASTQ** and are used in regular production work. They are also the basis for decomposition regions in the developing artificial intelligence **A**utomated **ME**shing **K**nowledge **S**ystem, **AMEKS**.

In 3-D, the primitive volumes are the 2-D primitives extended in depth together with the tetrahedon and two rectilinear transition volumes. These primitives are the foundation for extending **AMEKS** to 3-D.

INTRODUCTION

The finite element method is a fundamental simulation technique widely used in the engineering analysis community. This technique is capable of solving partial differential equations for very complex geometries and problems. However, successful use of the technique still requires significant expertise and time. Many researchers are investigating ways to further simplify or automate this modeling technique, thus allowing improved productivity, more accurate solutions, and use by less trained personnel.

Although there are a number of steps in the finite element modeling technique, often the most time consuming and expertise-intensive task faced by an analyst is the discretization of a general geometric definition of the problem into valid finite elements. The parameter-space mapping technique used in most finite element mesh generation programs requires that the geometry be subdivided into either rectangular areas in two dimensions (2-D) or rectilinear volumes in three dimensions (3-D) to produce, respectively, quadrilateral or hexahedral elements. Since subdivision occurs in parameter space, the edges bounding the areas and volumes need not be straight or parallel lines.

Not only is the mesh generation task tedious and error prone, but the accuracy and cost of the analysis directly depends on the size, shape, and number of elements in the mesh. Simple geometries may be subdivided easily into quadrilaterals or hexahedra while irregular features challenge the analyst. Experienced analysts mesh difficult geometry regions similarly. If this expertise can be captured in the definition of additional area and volume primitives, the productivity of experts and novices and the accuracy of their work would increase.

This paper presents the formulation and application of these additional primitives, *conjoint* meshing primitives, that assist the analyst in meshing in and around irregular features and that simplify the decomposition of complex shapes. The term *conjoint* is applied to these primitives because each is composed of several rectangular areas or rectilinear volumes. Interval assignments may vary from side to side, within a set of constraints, for each primitive. Thus, all of them are useful for transitions between other regular areas or volumes.

The body of this paper is organized in the following way. First, the 2-D primitive areas, the triangle, pentagon, semi-circle, circle, and rectangular transition, are discussed. These primitives are implemented in the Sandia meshing program **FASTQ** [1] and are used in regular production work. They are also the basis for decomposition regions in the developing artificial intelligence **A**utomated **ME**shing **K**nowledge **S**ystem, **AMEKS** [2,3,4]. Next, the 3-D primitive volumes are described. They are the 2-D primitives extended in depth together with the tetrahedon and two rectilinear transition volumes. These primitives are the foundation for extending **AMEKS** to 3-D and are currently being implemented in a new

3-D mesh generation program. Finally, example meshes generated using these 2-D and 3-D primitives are presented.

Two-Dimensional Primitives

The two-dimensional conjoint primitives described in the following paragraphs are the triangle, pentagon, rectangular transition, semi-circle, and circle. The triangle and pentagon primitives have unique solutions and are discussed first. The rectangular transition, semicircle and circle do not have unique solutions for subdivision and interval assignment. These are discussed at the end of this section.

Triangle Primitive

Figure 1 shows a triangle subdivided into three quadrilateral subareas. Obviously, a triangle may always be subdivided in this way to form three quadrilateral elements. This is very restrictive since each side of the triangle must have only two intervals. A more general interpretation of Figure 1 assigns additional intervals, n_1, n_2 and n_3, to each of the quadrilateral subareas. This allows for a different number of intervals on each of the three sides. The triangle primitive is then useful as a transition region.

Figure 1 suggests, therefore, the following matrix equation where m_1, m_2 and m_3 are the total number of intervals on the three sides of the triangle.

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix} = \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix} \tag{1}$$

Solving for the unknown n_i's gives:

$$\frac{1}{2} \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix} = \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix} \tag{2}$$

Since all of the n_i's must be greater than or equal to one, the following inequalities hold.

$$\begin{aligned} m_1 + m_2 &\geq m_3 + 2 \\ m_1 + m_3 &\geq m_2 + 2 \\ m_2 + m_3 &\geq m_1 + 2 \end{aligned} \tag{3}$$

Examining Figure 1 suggests a physical interpretation of Equation 3. The sum of the intervals of any two sides must be two greater than the number of intervals on the remaining side.

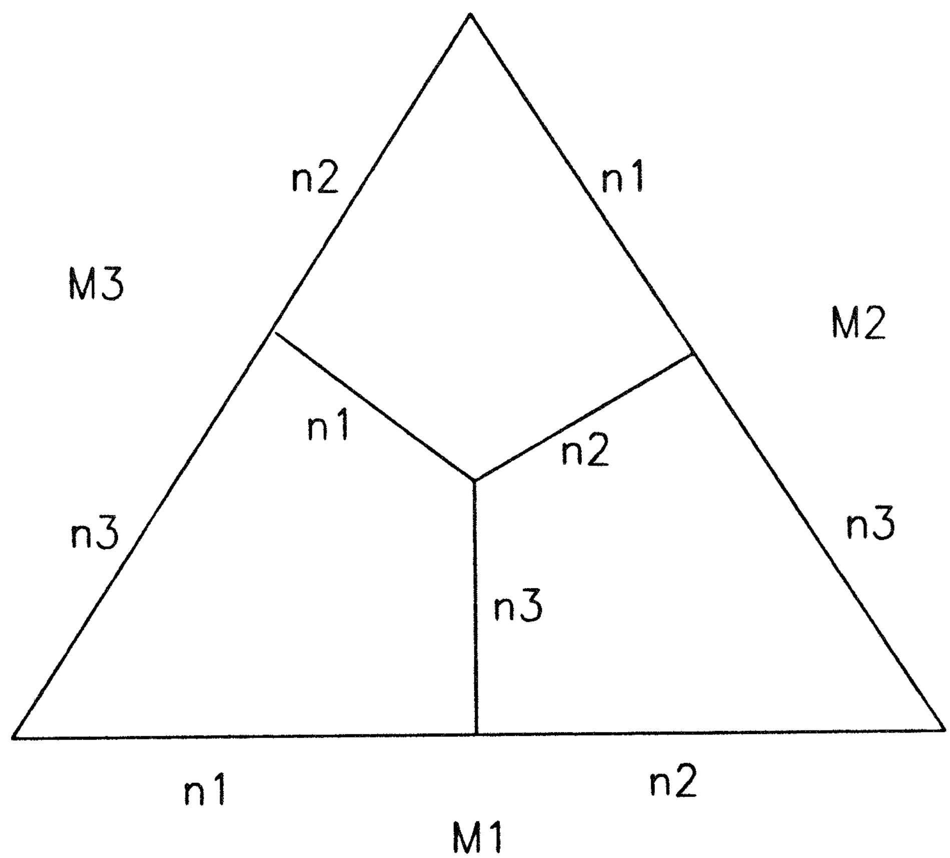

Figure 1: Triangle Primitive

A final condition dictated by Figure 1 is that the sum of the intervals must be even since the sum of m_1, m_2 and m_3 is equal to twice the sum of n_1, n_2 and n_3.

In summary, any triangular area may be mapped into a quadrilateral element mesh if the following conditions are meet.

1. Each side must have a minimum of two intervals.
2. The sum of the intervals around the perimeter is even.
3. The sum of the intervals for any two sides is two greater than the remaining side.

Pentagon Primitive

The derivation of the pentagon primitive closely follows that of the triangle primitive described above. Historically, it was the last of the two-dimensional primitives developed and is include here because of its similarity to the triangle primitive [5].

Figure 2 shows a pentagon primitive that may occur around a hole in a model. The pentagon is subdivided into five quadrilateral subareas, and two of these are combined along the dashed lines to form three meshable subareas.

If the n_i's are the number of intervals on the sides of the subareas, and if the m_i's are the number of intervals on the sides of the pentagon primitive, then Figure 2 suggests the following matrix equation.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{Bmatrix} = \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{Bmatrix} \tag{4}$$

Solving for the unknown n_i's gives:

$$\frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{Bmatrix} = \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{Bmatrix} \tag{5}$$

Since all of the n_i's must be greater than or equal to one, the following inequalities hold.

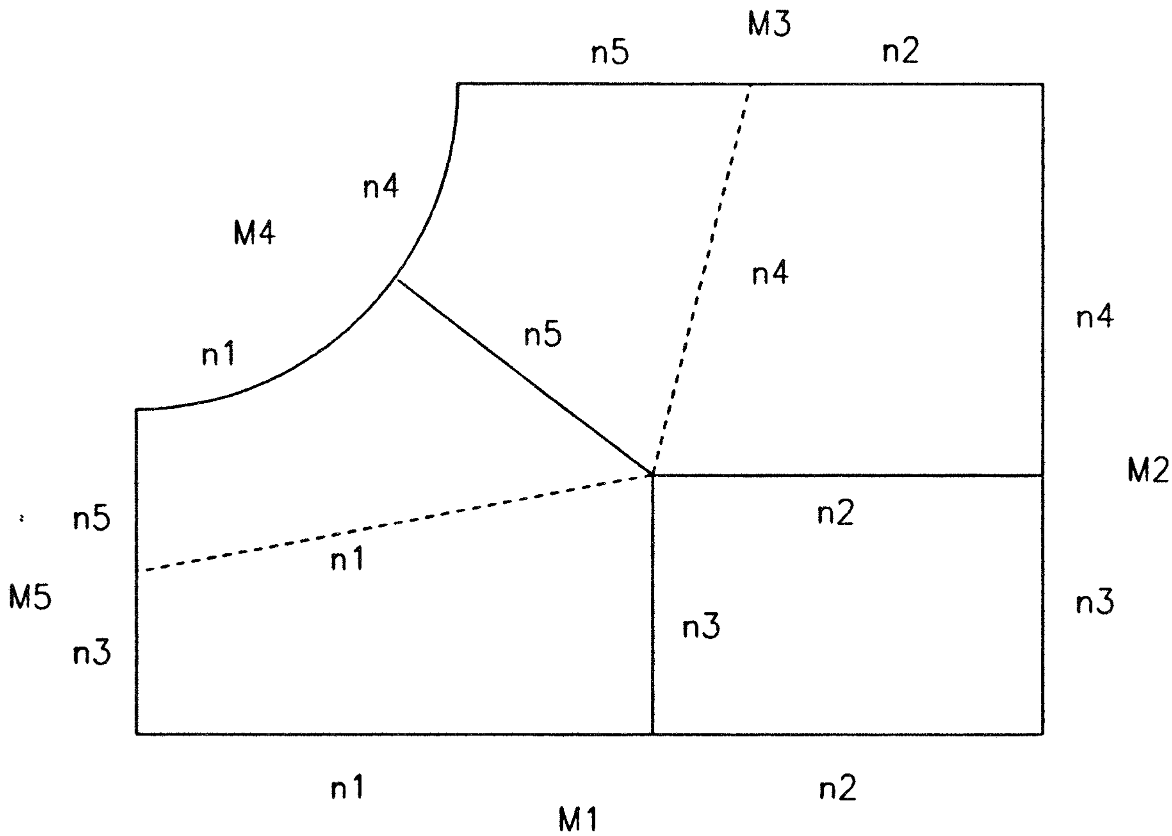

Figure 2: Pentagon Primitive

$$
\begin{aligned}
m_1 + m_4 + m_5 &\geq m_2 + m_3 + 2 \\
m_1 + m_2 + m_3 &\geq m_4 + m_5 + 2 \\
m_1 + m_2 + m_5 &\geq m_3 + m_4 + 2 \\
m_2 + m_3 + m_4 &\geq m_1 + m_5 + 2 \\
m_3 + m_4 + m_5 &\geq m_1 + m_2 + 2
\end{aligned} \tag{6}
$$

Examining Figure 2 suggests another physical interpretation of Equation 6 similar to the one for the triangle. The sum of the intervals of any three adjacent sides must be two greater than the sum of the remaining two sides.

Like the triangle, a final condition dictated by Figure 2 is that the sum of the intervals must be even since the sum of the m_i's is equal to twice the sum of the n_i's.

In summary, any pentagon area may be subdivided into quadrilateral elements if the following conditions are meet.

1. Each side must have a minimum of two intervals.
2. The sum of the intervals around the perimeter is even.
3. The sum of the intervals for any three adjacent sides is two greater than the sum of the intervals for the remaining two sides.

N-Sided Primitive

The derivation technique used for the triangle and pentagon may be extended to *n-sided* polygons. Each of these primitives has one irregular node at the interior point where the subareas all meet. At the irregular node, the average corner angle is 120° for a triangle primitive and 72° for a pentagon primitive. The size of this angle will determine whether other primitives with more sides generate usable meshes.

Rectangular Transition Primitive

Figure 3 shows a rectangular transition region subdivided into six quadrilateral subareas. It can be thought of as two triangle primitives, back-to-back, with the mid-side point pulled out to form a corner. This subdivision scheme allows for the greatest flexibility in the assignment of intervals. Each side may have a different number of intervals as long as the conditions derived below are met.

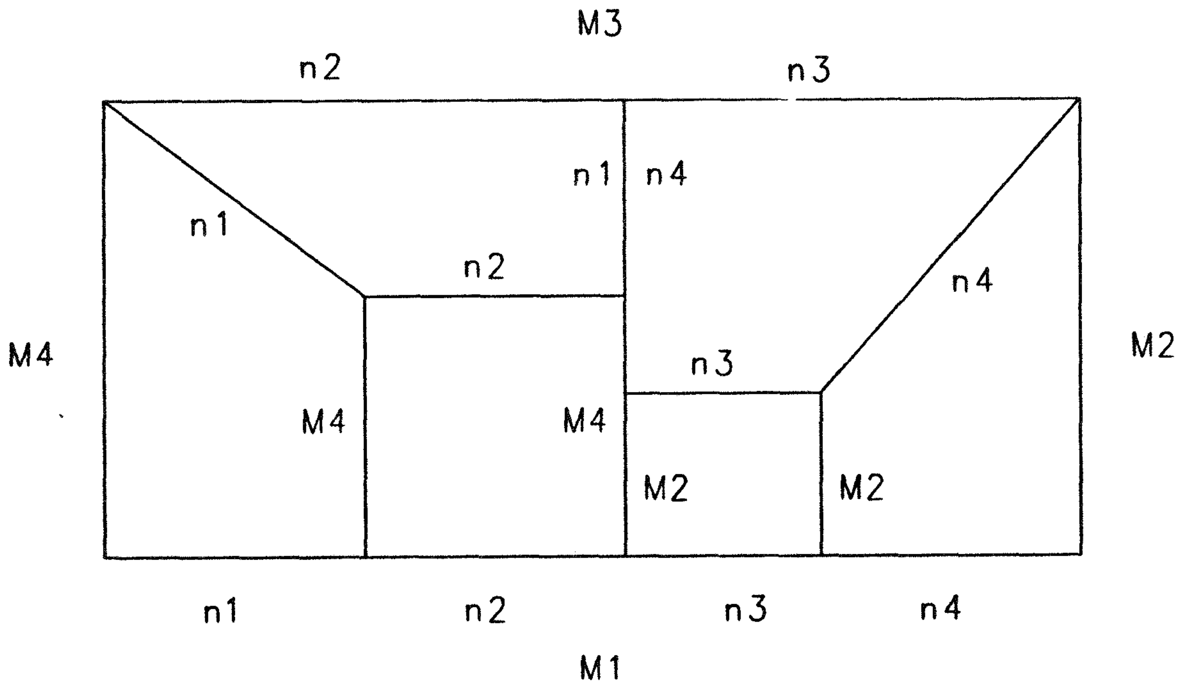

Figure 3: Rectangular Transition Primitive

The first two equations below are obvious from a review of Figure 3. The requirement that the interval assignments on either side of the mid-line must be equal gives us the third equation.

$$n_1 + n_2 + n_3 + n_4 = m_1 \tag{7}$$
$$n_2 + n_3 = m_3 \tag{8}$$
$$n_1 + m_4 = n_4 + m_2 \tag{9}$$

Equation 8 is used to write Equation 7 in terms of n_1, n_4, m_1, and m_3. Combining this equation with Equation 9 leads to the solution below for n_1 and n_4.

$$\begin{aligned} n_1 &= \frac{m_1 + m_2 - m_3 - m_4}{2} \geq 1 \\ n_4 &= \frac{m_1 + m_4 - m_2 - m_3}{2} \geq 1 \end{aligned} \tag{10}$$

The interval assignments for n_2 and n_3 are not unique. The only requirement is that they sum to m_3. On the other hand, a desirable characteristic of the mesh is a nearly vertical mid-line. If the interval spacing is uniform along the bottom and the top, the ratio below will force a vertical mid-line.

$$\frac{n_1}{n_4} = \frac{n_2}{n_3} \tag{11}$$

If the ratio of Equation 11 is enforced, then a unique solution exists. In terms of n_1, n_4 and m_3, n_2 and n_3 are

$$\begin{aligned} n_2 &= \frac{n_1}{n_1 + n_4} m_3 \\ n_3 &= \frac{n_4}{n_1 + n_4} m_3 \end{aligned} \tag{12}$$

Unfortunately, the exact solution forces n_2 and n_3 to be a common multiple of n_1 and n_4 respectively (see Equation 11). As before, the values of the n_i's must be positive integers which further limits the utility of an *exact* solution. On the other hand, if the vertical mid-line condition is relaxed so that instead of being equal the ratios are approximately equal, a useful solution results by rounding the values of n_2 and n_3 to the nearest integer.

In summary, any rectangular transition area may be subdivided into quadrilateral elements if the following conditions are meet.

1. The sum of the intervals around the perimeter is even and is greater than or equal to eight.
2. The sum of the intervals of the side with the greatest number and of one adjacent side must be greater than or equal to the sum of the intervals of the other two sides plus two (see Equation 10).

Semi-Circle Primitive

The semi-circle primitive is a rectangular transition primitive fit into a region with only two sides. The formulation is very much the same except that where the rectangular transition required one assumption to solve the problem, the semi-circle requires three. Again the first two equations are obvious from Figure 4. They are simply the sum of the intervals along the two lines that bound the semi-circle. The third equation enforces the condition that the number of intervals on either side of the vertical mid-line must be the same just like the rectangular transition.

$$\begin{aligned} n_1 + n_2 + n_3 + n_4 &= m_1 \\ n_2 + n_3 + n_5 + n_6 &= m_2 \\ n_1 + n_6 &= n_4 + n_5 \end{aligned} \tag{13}$$

Since there are six unknowns and only three equations, this system has no unique solution. Three additional equations can be written that will provide the necessary constraints to solve for the intervals along the remaining sides. In the equations below, n_2 and n_3 are assumed to be the average interval assignment if m_1 is sufficiently large. Remember that there are eight intervals around the primitive. Otherwise, they are simply half the base less one. The value of n_6 is half the difference of m_2 and n_2 and n_3. These values are all rounded to the nearest integer.

Let:

$$n_2 = \begin{cases} \frac{m_1+m_2}{8} & \text{if } 3m_1 - m_2 \geq 8 \\ \frac{m_1}{2} - 1 & \text{otherwise} \end{cases} \tag{14}$$

$$n_3 = n_2 \tag{15}$$

$$n_6 = \frac{m_2 - 2n_2}{2} \tag{16}$$

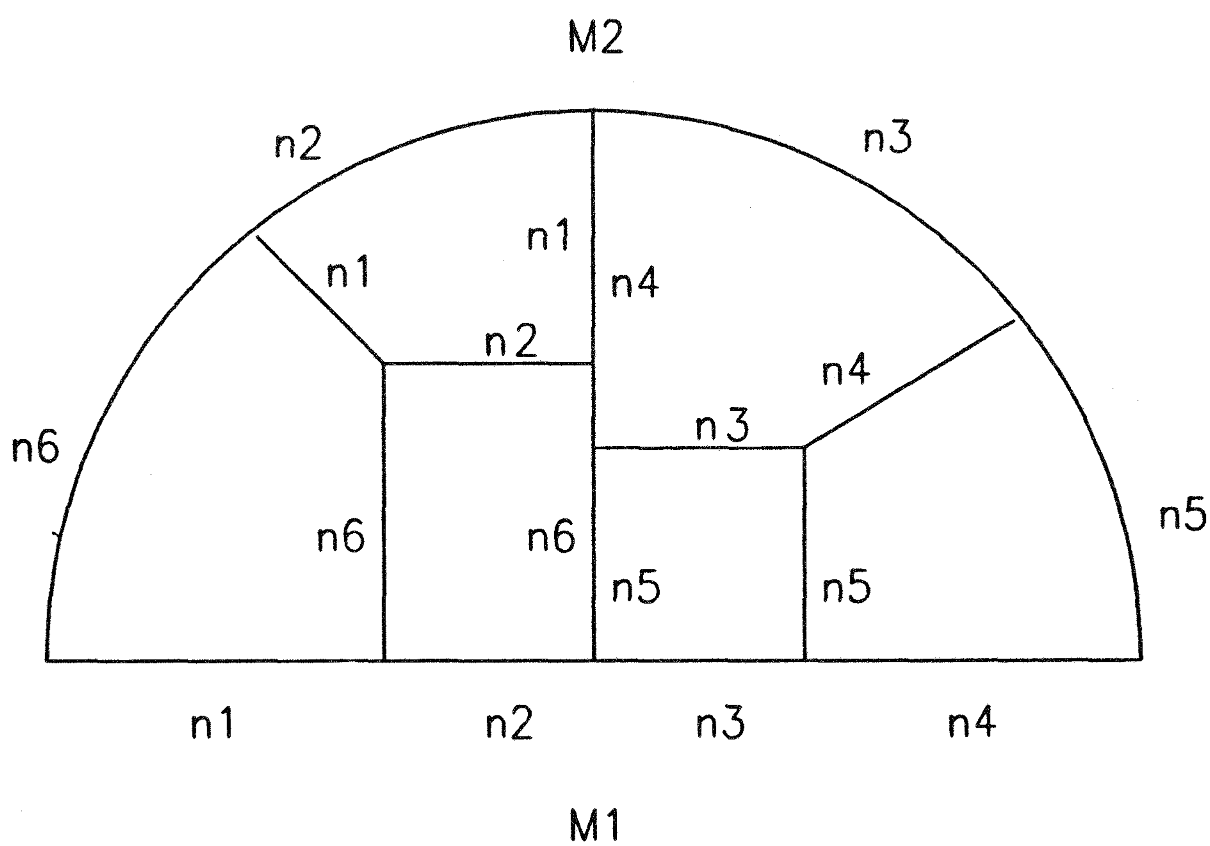

Figure 4: Semi-Circle Primitive

Now that n_2, n_3 and n_6 are known, Equation 13 may be rewritten as:

$$\begin{aligned} n_1 + n_4 &= m_1 - 2n_2 \\ n_5 &= m_2 - 2n_2 - n_6 \\ n_1 - n_4 - n_5 &= -n_6 \end{aligned} \tag{17}$$

Solving for n_1, n_4 and n_5 results in the following:

$$\begin{aligned} n_1 &= \frac{m_1 + m_2}{2} - 2n_2 - n_6 \\ n_4 &= \frac{m_1 - m_2}{2} + n_6 \\ n_5 &= m_2 - 2n_2 - n_4 \end{aligned} \tag{18}$$

In summary, any semi-circle area may be subdivided into quadrilateral elements if the following conditions are meet.

1. The sum of the intervals around the perimeter is even.
2. The intervals for the side with the great number must be greater than or equal to those for the other side which must be greater than or equal to four (see Figure 4).

Circle Primitive

The circle primitive is quite simple as seen in Figure 5. In fact, no additional code was written to implement this primitive in **FASTQ** whose meshing capabilities are derived from an earlier code, **QMESH** [6]. A rectangle is meshed in the circle, and this mesh is then forced toward the center by adding one or more element *necklaces*. The only equation for the primitive is:

$$2n_1 + 2n_2 = m_1 \tag{19}$$

Equation 19 requires m_1 to be even. The values of n_1 and n_2 are abritrary but should be chosen to reflect the general shape of the body being meshed. For example, if the region being meshed is a *perfect* circle, then the values of n_1 and n_2 should be equal. On the other hand, if the circle being meshed is squashed, then the values of n_1 and n_2 should be adjusted so that the intervals are evenly distributed in the *long* and *short* directions.

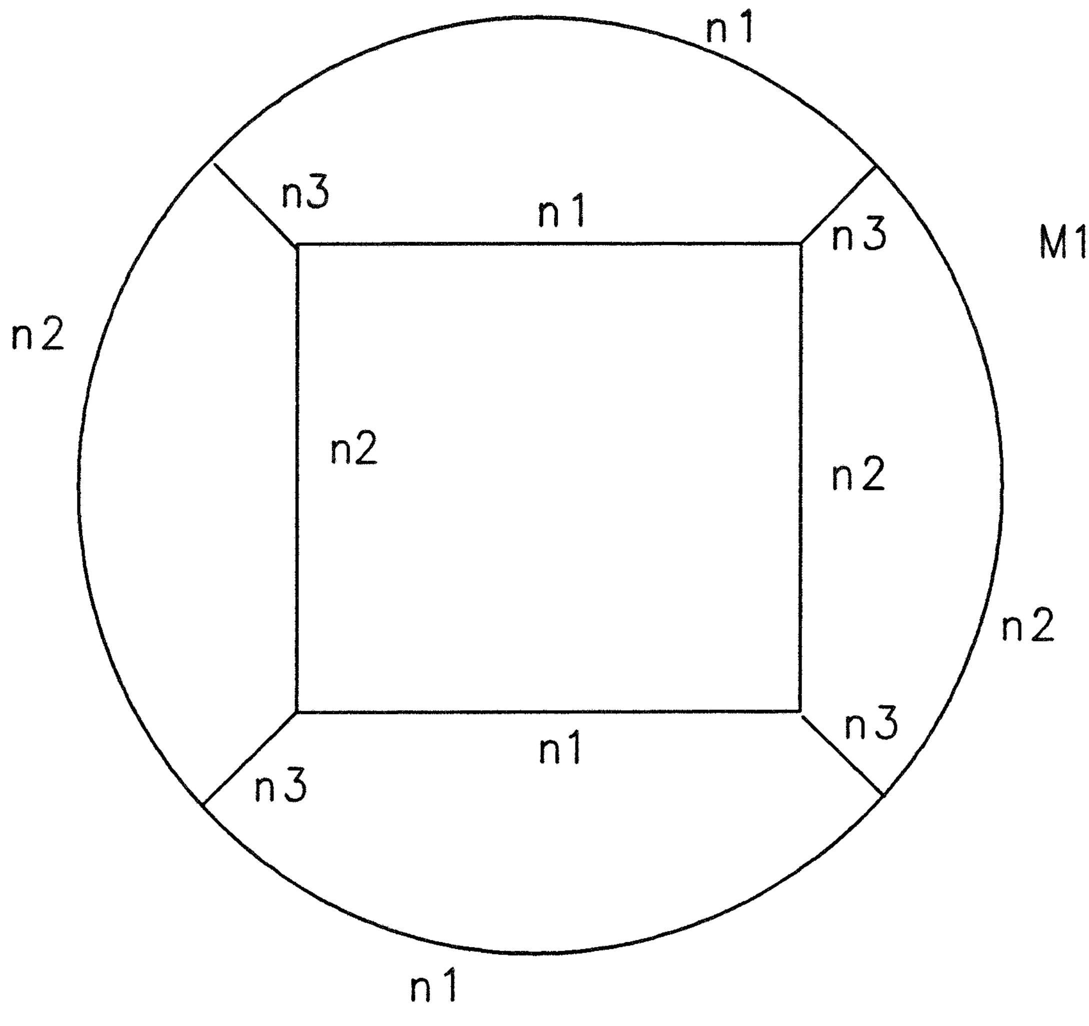

Figure 5: Circle Primitive

The only constraints for circle primitive is as follows:

$$n_3 \geq 1$$

In other words, the number of rows of elements between the circumference and the interior rectangle is abritrary. If it is greater than or equal to one, then no irregular nodes or ill formed elements will occur on the region boundary. If n_3 is large, then the elements in the middle of the circle become small compared to those on the boundary. A value of two or three for n_3 produces acceptable meshes.

Three-Dimensional Primitives

The 3-D primitives described below include all of the 2-D primitives discussed previously extended in depth together with the purely 3-D primitives. The first set of primitives is known as $2\frac{1}{2}$–D primitives because they are 2-D except for being translated or rotated through space.

$2\frac{1}{2}$–D Primitives

The $2\frac{1}{2}$–D primitives include the wedge (triangle), the cylinder (circle), the half-cylinder (semi-circle), the regular hexahedral transition (rectangular transition), and the pentagon sweep (pentagon). No additional equations are necessary to implement these primitives. The 2-D primitives are simply translated and/or rotated as required to generate the requested number of elements. Examples of the $2\frac{1}{2}$–D primitives are provided later in this paper.

Tetrahedron Primitive

The derivation of the tetrahedron primitive is similar to the derivation of the triangle and pentagon primitives in 2-D. In Figure 6, p_1, p_2, p_3, and p_4 are the tetrahedron vertices, and m_1, m_2, m_3, m_4, m_5, and m_6 are the interval numbers on each of the tetrahedron's sides. The intervals need not be equal on any of the sides, although other constraints apply, which makes this primitive useful as a transition region.

Figure 7 shows the tetrahedron divided into four *logical* hexahedra. They are called *logical* because they may have non-planar faces.

Suppose that the proposed subdivision is one like Figure 7 where n_1, n_2, n_3, and n_4 are the interval subdivisions on the respective hexahedra's sides. A system of six equations in four unknowns can be derived similar to the 2-D cases described previously.

In matrix form, the equation is:

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{Bmatrix} = \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{Bmatrix} \tag{20}$$

It is known that a system of equations $\mathbf{Ax} = \mathbf{B}$ has a solution if and only if the rank of the augmented matrix $[\mathbf{A} : \mathbf{B}]$ is the same as the rank of the coefficient matrix $\mathbf{A}$ [7].

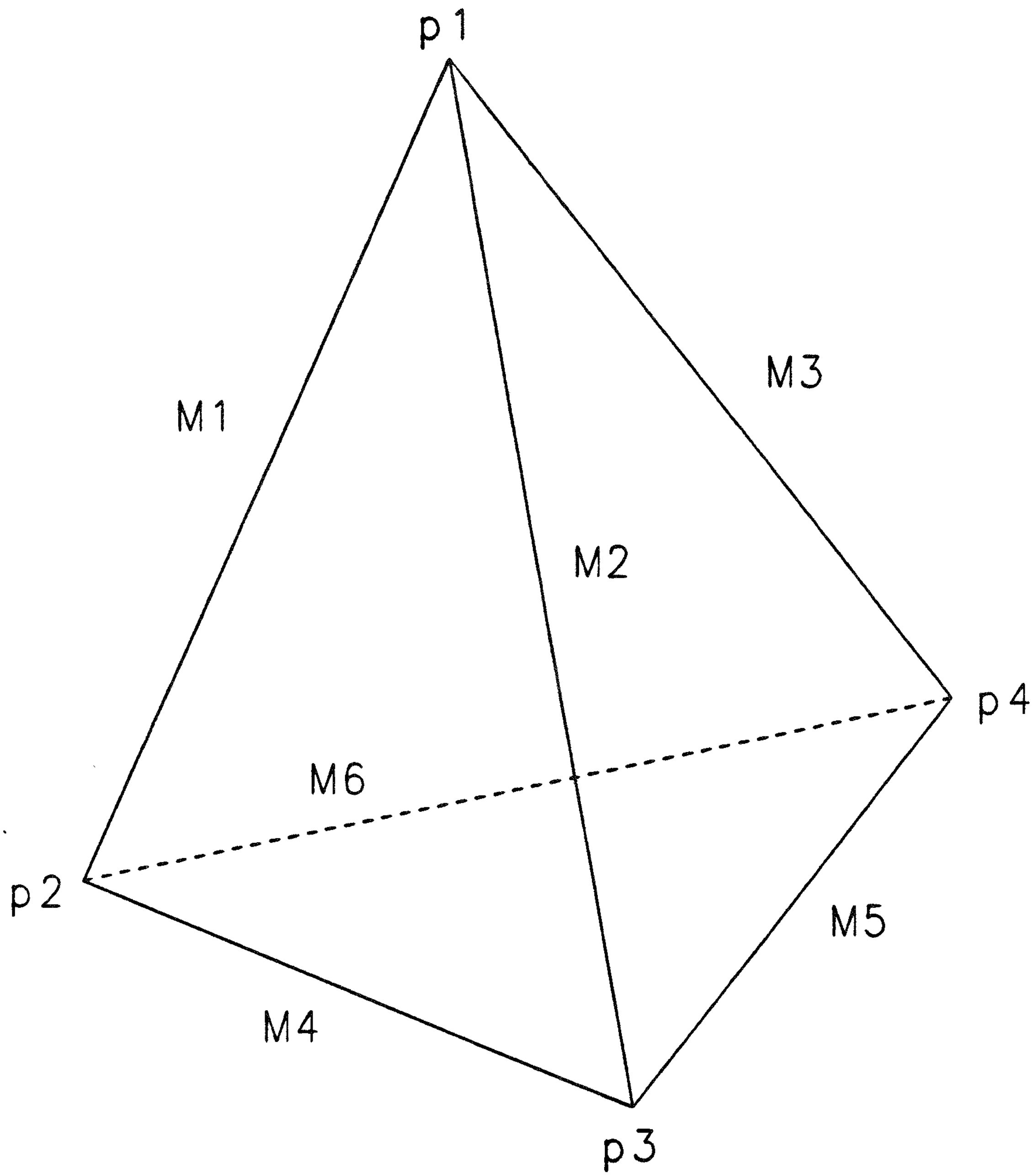

Figure 6: A Tetrahedron with Non-Uniform Intervals

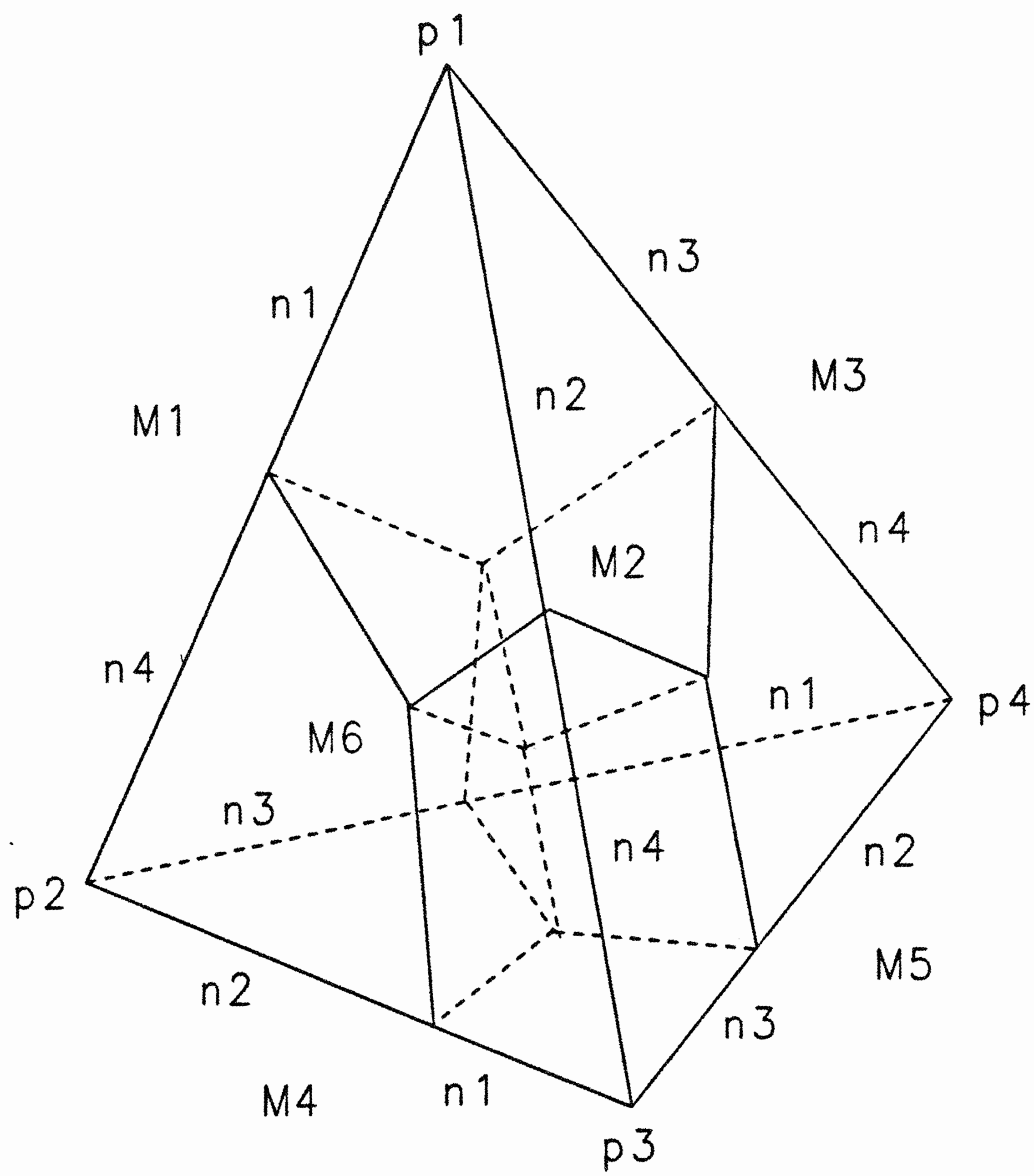

Figure 7: A Tetrahedron Filled with Logical Hexahedra

Because elementary row operations do not change the row space of a matrix, they are applied on this system to find its rank value. After these transformations the augmented matrix is

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & \frac{m_1+m_4-m_2}{2} \\ 0 & 1 & 0 & 0 & \frac{m_2+m_4-m_1}{2} \\ 0 & 0 & 1 & 0 & \frac{2m_3+m_4-m_1-m_2}{2} \\ 0 & 0 & 0 & 1 & \frac{m_1+m_2-m_4}{2} \\ 0 & 0 & 0 & 0 & m_5 - m_1 + m_3 - m_4 \\ 0 & 0 & 0 & 0 & m_6 - m_1 + m_2 - m_4 \end{array}\right] \tag{21}$$

The coefficient part of the augmented matrix has rank four. This means that there are two dependent equations in the system that can be represented in terms of the other four. For the system to be solvable, the B vector must have rank 4 also. Therefore, the following equations must be satisfied.

$$\begin{aligned} m_5 + m_1 - m_3 - m_4 &= 0 \\ m_6 + m_2 - m_3 - m_4 &= 0 \end{aligned} \tag{22}$$

From the last two equalities, one can infer that

$$m_5 + m_1 = m_3 + m_4 = m_6 + m_2 \tag{23}$$

This is one of the compatibility conditions. In other words, if the sum of opposite side intervals (i.e. through the tetrahedron) are equal, then the tetrahedron may be subdivided in the proposed manner (see Figure 7).

Solving the reduced system gives

$$\left\{\begin{array}{c} n_1 \\ n_2 \\ n_3 \\ n_4 \end{array}\right\} = \left[\begin{array}{c} \frac{m_1+m_4-m_2}{2} \\ \frac{m_2+m_4-m_1}{2} \\ \frac{2m_3+m_4-m_1-m_2}{2} \\ \frac{m_1+m_2-m_4}{2} \end{array}\right] \tag{24}$$

As with the other primitives, the n_i's must be integers.

$$\begin{aligned} n_1 &= \tfrac{m_1+m_4-m_2}{2} & \geq 1 \\ n_2 &= \tfrac{m_2+m_4-m_1}{2} & \geq 1 \\ n_3 &= \tfrac{2m_3+m_4-m_1-m_2}{2} & \geq 1 \\ n_4 &= \tfrac{m_1+m_2-m_4}{2} & \geq 1 \end{aligned} \tag{25}$$

Therefore, if the sum of the intervals around each of the triangular faces of the tetrahedron are even, Eq. 24 will have a unique solution, and the tetrahedron may be subdivided. This is the other compatibility condition.

Hexahedral Transition Primitives

This section describes two hexahedral transition primitives. Each primitive is named for the number of edges whose interval assignment may deviate from that usually associated with a regular hexahedron. Other transition primitives are surely possible. In fact, the regular hexahedral transition, a $2\frac{1}{2}$-D primitive, is another example.

One-Edge Hexahedral Transition

A regular hexahedron meshing primitive may have different interval assignments in each of the three parameteric directions. The one-edge hexahedral transition allows one edge of the primitive to vary from the norm. Figure 8 shows a one-edge transition primitive. In this primitive, a smaller hexahedron is centered on the edge that has more intervals than it would if it were a regular hexahedron. This is the *irregular* edge. The interior faces of the smaller hexahedron are connected to the primitive's faces by the diagonal lines. Thus, there are five hexahedrons within the one-edge primitive.

The interval assignments for the primitive are m_1, m_2, m_3, and m_4. The only unknown is n_1 which is the number of intervals on either side of the smaller hexahedron. Since $2n_1 + m_1$ must equal the number of intervals along the top edge, n_1 is simply

$$n_1 = \frac{m_2 - m_1}{2} \tag{26}$$

The only condition for using this primitive is that the difference between the irregular edge and the other edges in the same parametric direction is equal and even.

Three-Edge Hexahedral Transition

The three-edge or corner hexahedral transition is similar to the one-edge transition except that the small hexahedron is placed at a corner as shown in Figure 9 This primitive consists then of four hexahedra. With the smaller hexahedron in the corner, the three edges that join at the corner may have more intervals assigned to them than in a regular hexahedral primitive.

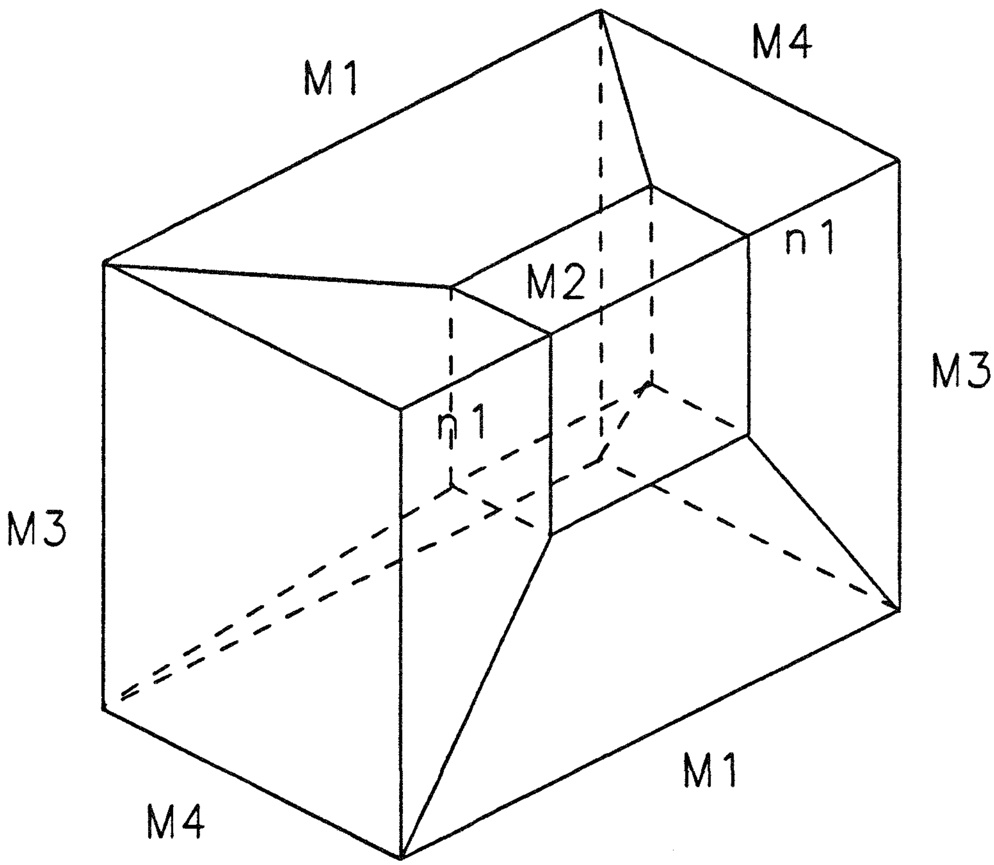

Figure 8: A One-Edge Hexahedral Transition

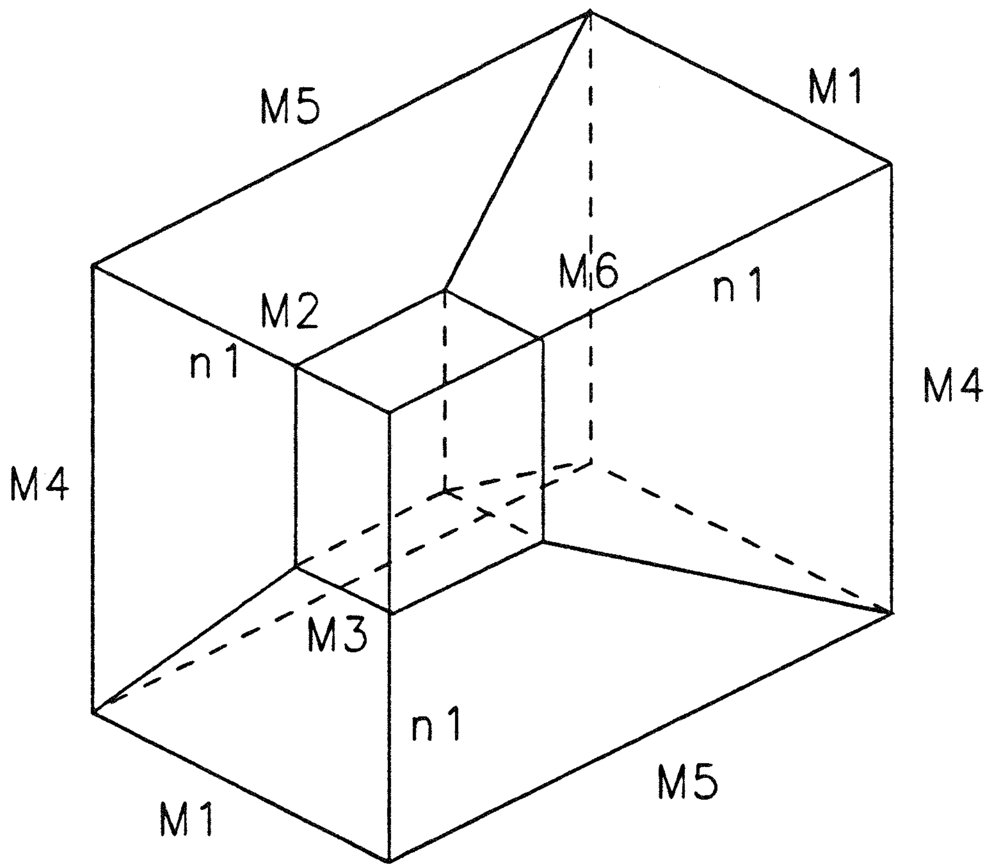

Figure 9: A Three-Edge Hexahedral Transition

The following equations summarize the conditions that must be met to mesh this primitive.

$$
\begin{aligned}
n_1 + m_1 &= m_2 \\
n_1 + m_4 &= m_3 \\
n_1 + m_5 &= m_6
\end{aligned}
\tag{27}
$$

Solving each of these equations for n_1 gives

$$m_2 - m_1 = m_3 - m_4 = m_6 - m_5 \tag{28}$$

Therefore, the condition for using this primitive is that the three differences between the irregular edge and all the other edges in each parameteric direction must be equal.

Examples

The examples below illustrate the utility of the various primitives described in the preceeding paragraphs. The 2-D examples are more complex than the 3-D examples. The 2-D primitives were implemented in 1987 and early 1988, and as such, they have a much longer history and track record. On the other hand, the 3-D primitives are just now being implemented and used.

The 2-D examples were generated and plotted using **FASTQ**. The $2\frac{1}{2}$-D examples were generated with **FASTQ** and **GEN3D**. **GEN3D** is a Sandia code for translating and/or rotating 2-D meshes to form 3-D meshes. The 3-D examples were generated and plotted with **MOVIE-STAR** [8,9].

2-D Examples

Figure 10 shows all of the 2-D primitives meshed individually. Each row is a different primitive, and each column shows the primitive applied to another shape.

The top row shows the standard rectangular mesh produced by **FASTQ**. Parameter-space mapping is very adept at meshing such non-rectangular shapes as the ring in the middle column and the lazy "S" in the right column.

Beginning with the second row, the conjoint primitives are the triangle, the semi-circle, the circle, the pentagon, and the rectangular transition. Each primitive is easily identified by the number of irregular nodes and the number of elements connected to the irregular node. For example, the triangle has one irregular node where three elements meet while the pentagon has one irregular node where five elements meet. The semi-circle and the rectangular transition have two irregular nodes where three elements meet. Remember that these primitives are really just two back-to-back triangles. Finally, the circle has four irregular nodes where three elements meet.

The meshes are well formed with few bad corner angles even for areas that certainly do not fit the geometric definitions of triangle, pentagon, etc. The mesh smoothing algorithms available in **FASTQ** are used to provide the best possible mesh for each of these primitives.

Figure 11 shows four different bodies composed of multiple meshing areas. The left column shows the decomposition of the body into suitable regions. The right column shows the resulting mesh. The top row shows the triangle and the rectangular transition blending areas of different intervals together. The second row illustrates the application of the pentagon primitive around holes as well as the circle primitive.

The body of the third row uses two semi-circle primitives to mesh its upper portion and

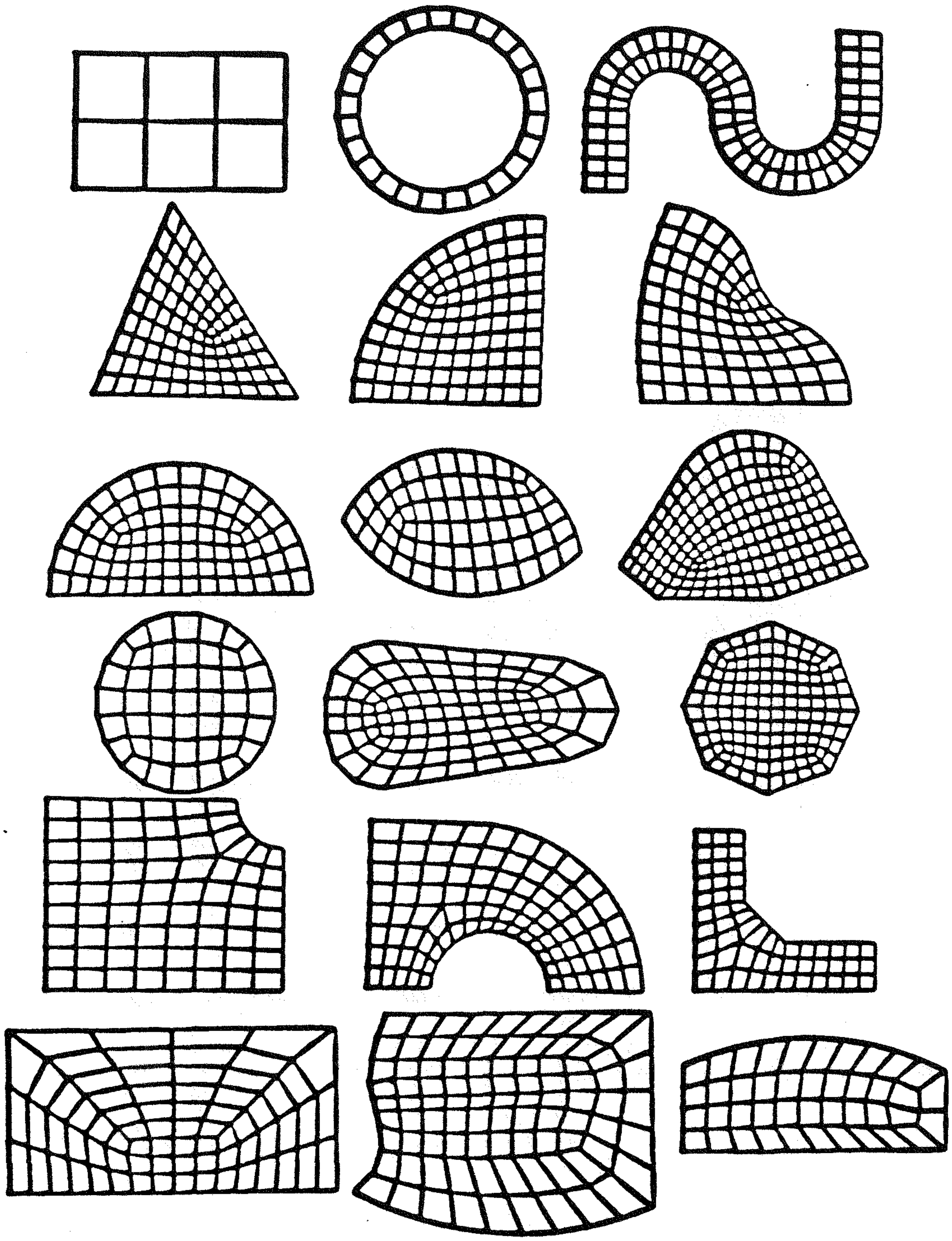

Figure 10: Two Dimensional Meshed Primitives

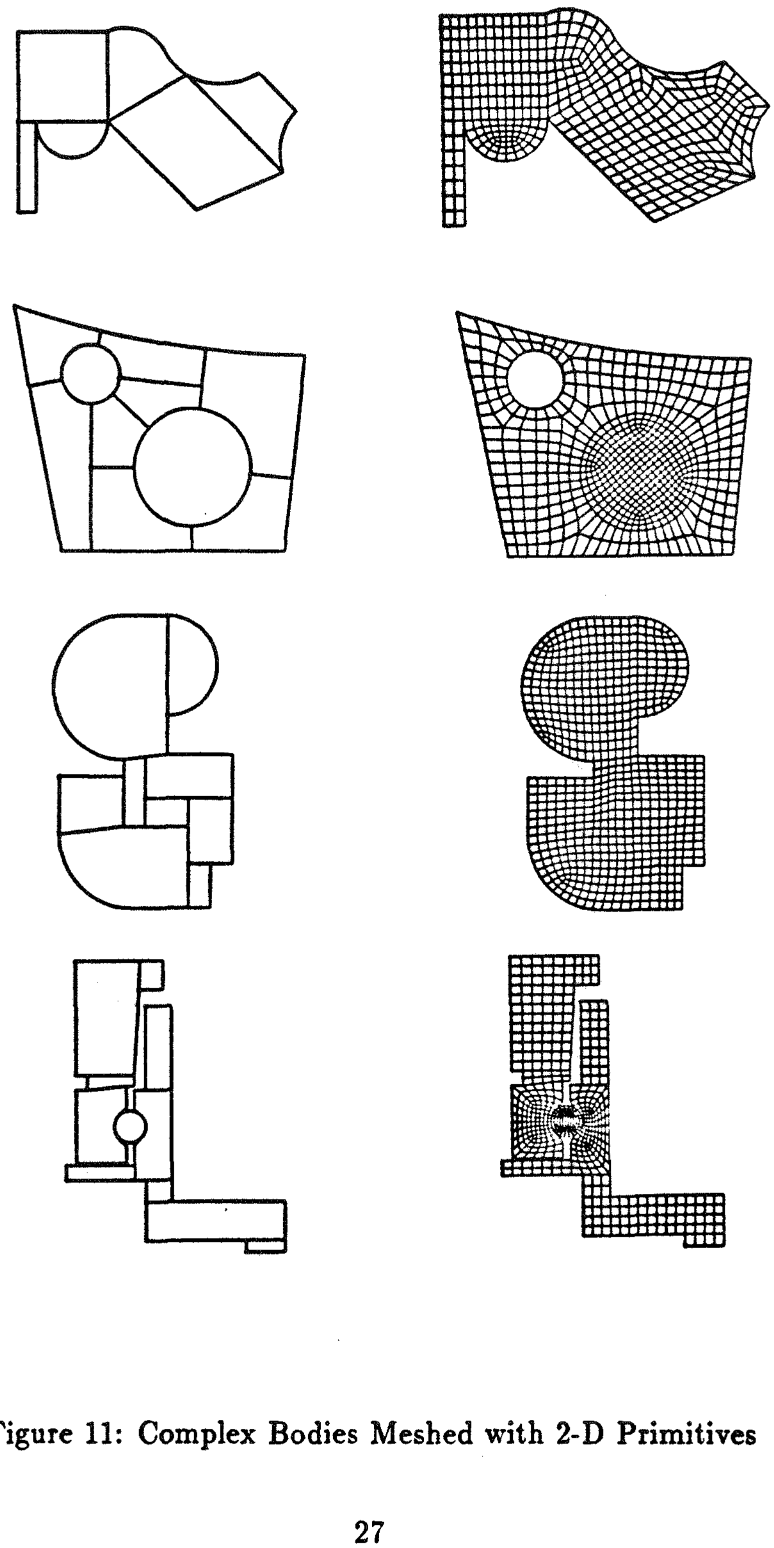

Figure 11: Complex Bodies Meshed with 2-D Primitives

the triangle primitive to mesh its lower left corner. All the rest of its areas are meshed with the standard rectangle. The last object uses the standard rectangle together with two rectangular transition primitives and a circle primitive. The transition primitives are used to match the fine mesh around the circle without forcing small elements throughout the model.

$2\frac{1}{2}$-D Examples

Figure 12 illustrates the use of the 2-D primitives in three dimensions. These are the $2\frac{1}{2}$-D primitives. The top row shows a triangle primitive translated, on the left, and rotated, on the right, to form a wedge. The second row shows a pentagon primitive translated and a rectanglar transition primitive rotated to form, respectively, a pentagon sweep and a regular hexahedral transition. The last row shows a semi-circle primitive rotated to form a curved half-cylinder and a circle primitive translated to form a cylinder.

The rectangular transition and circle primitives are rotated in Figure 13 to generate a sector of a toroidal object. Many 3-D objects can be modelled using only these $2\frac{1}{2}$-D primitives.

3-D Examples

The 3-D primitives in Figure 14 are not as smooth as those of the sections above. **MOVIE-STAR** does not have smoothing algorithms like **FASTQ**.

The top row shows the tetrahedron primitive, the middle row shows the three-edge hexahedral transition, and the last row shows the one-edge hexahedral transition. The left and right columns show the subdivided primitive and the resulting mesh with hidden surfaces removed. Note that the interval assignment differs on each of the visible edges of the meshed tetrahedron.

Unlike the 2-D primitives where only the number of intervals need match along an edge, the topology of the matching faces between adjacent 3-D primitives must be the same. In other words, the derivation of the grid on the face of a tetrahedron must match that of the wedge with the same interval assignments if these two primitives are to be used together.

Smoothing techniques like those in **FASTQ** must be adapted to 3-D to improve mesh quality. Calculation of internal subdivision points and smoothing is more complicated in 3-D since points and nodes must remain on the surface of the model. Obviously, this is not a consideration in planar, 2-D models.

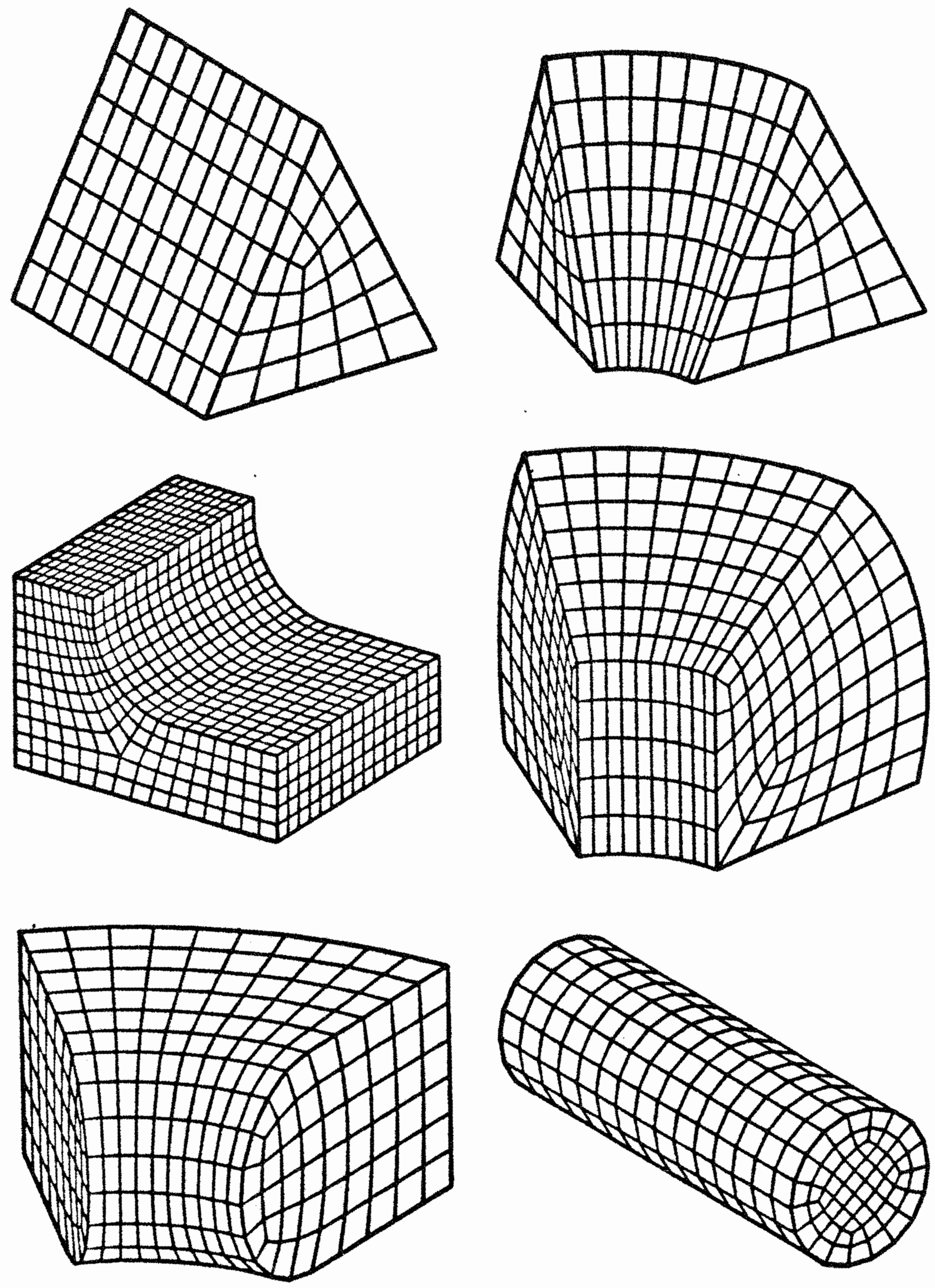

Figure 12: Two and One Half Dimensional Mesh Primitives

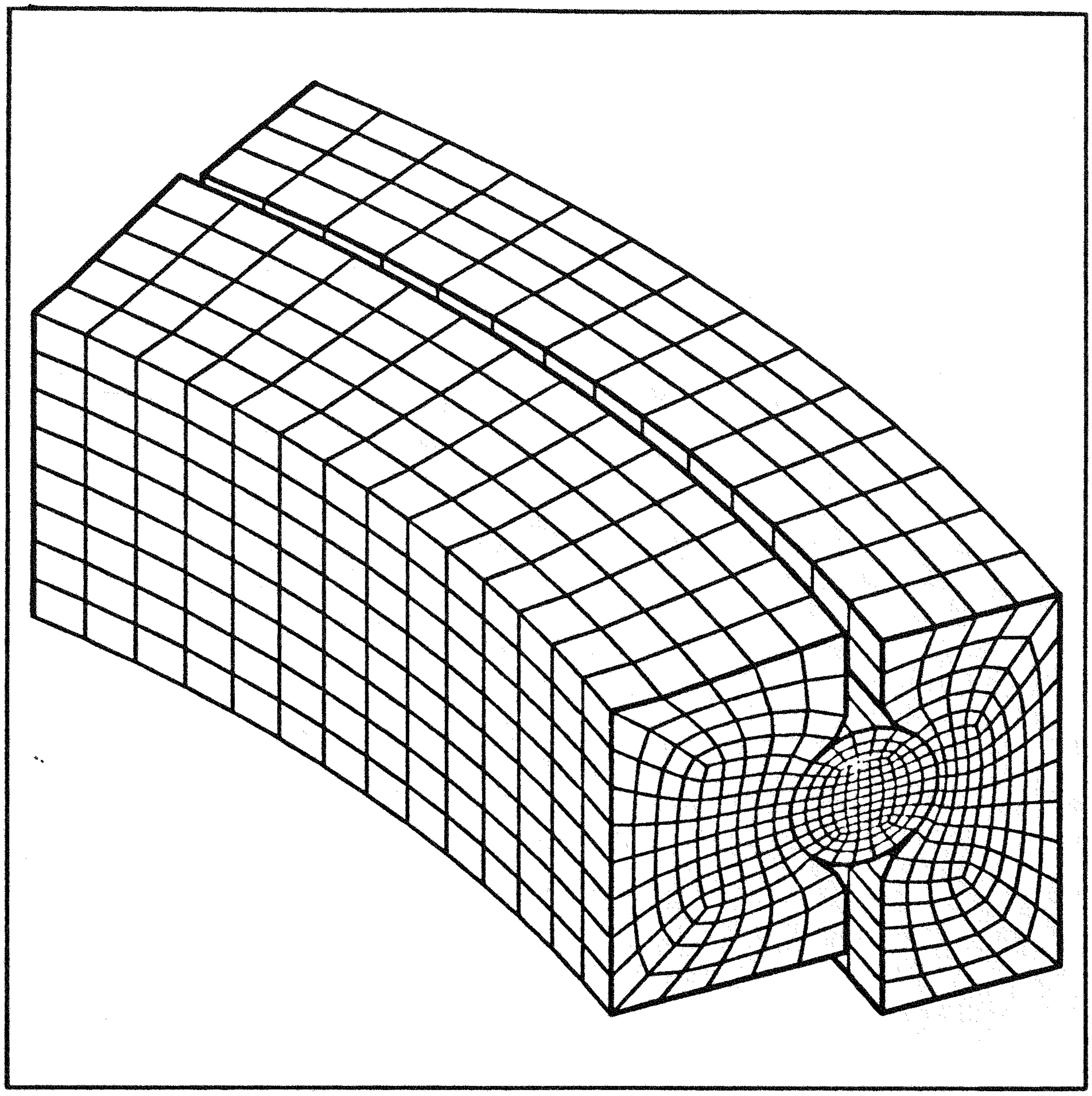

Figure 13: Complex 3-D Body Mesh with $2\frac{1}{2}$ -D Primitives

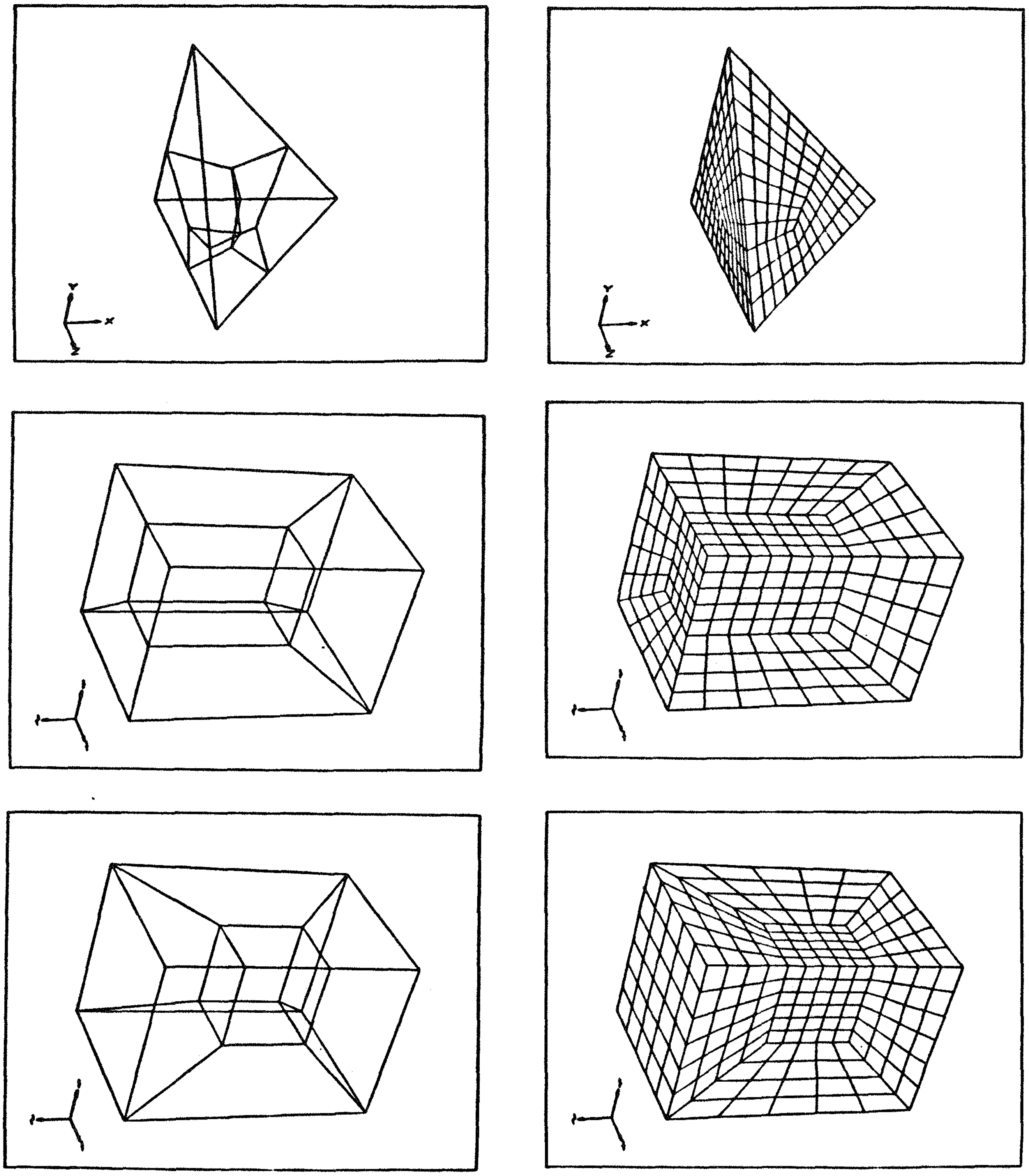

Figure 14: Three Dimensional Meshed Primitives

Conclusions

The 2-D primitives derived in this paper have proven useful in production work. They provide the analyst an easy way of decomposing a body without the necessity of further subdividing the area into quadrilaterals. The **FASTQ** smoothing algorithms are applied to the entire primitive to produce a superior mesh. If an analyst were to subdivide the region into quadrilaterals without the aid of the conjoint primitives, only the individual quadrilateral areas would be smoothed. The extension of these primitives to $2\frac{1}{2}$-D makes them useful in modelling a large number of 3-D problems.

The true 3-D primitives have not yet proven their utility since they have only recently become available. Further work will result in the addition of a hemisphere and a sphere primitive. A pentahedron primitive is also under investigation.

References

[1] T. D. Blacker, *FASTQ Users Manual Version 1.2,* Sandia Report No. SAND88-1326, Sandia National Laboratories, Albuquerque, NM, 1988.

[2] T. D. Blacker, J. L. Mitchiner, L. R. Phillips, Y. T. Lin, *Knowledge System Approach to Automated Two-Dimensional Quadrilateral Mesh Generation,* Proceedings of ASME International Computers in Engineering Conference, Vol.3, pp. 153-162, 1988.

[3] T. D. Blacker, M. B. Stephenson, J. L. Mitchiner, L. R. Phillips, Y. T. Lin, *Automated Quadrilateral Mesh Generation: A Knowledge System Approach* , ASME Paper No. 88-WA/CIE-4.

[4] L. R. Phillips, J. L. Mitchiner, T. D. Blacker, Y. T. Lin, *A Knowledge System for Automatic Finite Element Mesh Generation: AMEKS,* Proceedings of the First Annual International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Vol. 2, pp. 668-678, June 1988.

[5] D. H. Recalde, *The Pentagon and the Tetrahedron Decomposition Algorithms for Automatic Finite Element Meshing,* Unpublished Masters Project, Brigham Young University, Department of Civil Engineering, August 1988.

[6] R. E. Jones, *QMESH: A Self-Organizing Mesh Generation Program,* Sandia Report No. SLA-73-1088, Sandia National Laboratories, Albuquerque, NM, July 1974.

[7] A. Yaqub and H. Moore, *Elementary Linear Algebra with Applications,* Addison-Wesley Publishing Company, Inc. New York, p. 140, 1980.

[8] M. B. Stephenson and S. E. Benzley, *MOVIE-STAR Users Manual,* Engineering Computer Graphics Laboratory, Department of Civil Engineering, Brigham Young University, Provo, Utah, 1988.

[9] M. B. Stephenson and S. E. Benzley, *MOVIE-STAR Reference Manual,* Engineering Computer Graphics Laboratory, Department of Civil Engineering, Brigham Young University, Provo, Utah, 1988.