Paving: A new approach to automated quadrilateral mesh generation

Article *in* International Journal for Numerical Methods in Engineering · September 1991

DOI: 10.1002/nme.1620320410

2 authors, including:

Ted D. Blacker
Sandia National Laboratories
46 PUBLICATIONS **1,923** CITATIONS

SEE PROFILE

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 32, 811–847 (1991)

PAVING: A NEW APPROACH TO AUTOMATED QUADRILATERAL MESH GENERATION*

TED D. BLACKER

Applied Mechanics Division III, Sandia National Laboratories, Albuquerque, New Mexico 87185, U.S.A.

MICHAEL B. STEPHENSON

Civil Engineering Department, Brigham Young University, Provo, Utah 84602, U.S.A.

SUMMARY

This paper presents a new mesh generation technique, *paving*, which meshes arbitrary 2-D geometries with an all-quadrilateral mesh. Paving allows varying element size distributions on the boundary as well as the interior of a region. The generated mesh is well formed (i.e. nearly square elements, elements perpendicular to boundaries, etc.) and geometrically pleasing (i.e. mesh contours tend to follow geometric contours of the boundary). In this paper we describe the theory behind this algorithmic/heuristic technique, evaluate the performance of the approach and present examples of automatically generated meshes.

1. INTRODUCTION

The finite element method is a fundamental numerical analysis technique in wide-spread use in the engineering community. It has long been recognized that this technique is a keystone in the design and understanding of complicated mechanisms and processes. However, one of the biggest obstacles that must be overcome for this method to be used is the discretization of a general geometry of the problem into a valid finite element mesh. Not only is the mesh generation task tedious and error prone, but the accuracy and cost of the analysis direcly depends on the size, shape and number of elements in the mesh.

Since automation of mesh generation has a potentially large payoff, extensive research continues in this area. Most of the research focuses on techniques such as the 2-D quadtree and 3-D octree,[1–3] triangulation[4,5] or substructuring.[6] In general, these techniques generate meshes using triangular elements that are less accurate and less versatile than quadrilateral elements. Some of these techniques have been extended to produce all-quadrilateral meshes. However, most commercial mesh generation codes still employ parameter space mapping[7,8] to generate an all-quadrilateral mesh.[9] This technique is not automatic, in that the analyst must decompose the geometry into regions that map well into parametric space. Even though the mapping technique is laborious and expertise-intensive, many skilled analysts prefer it over the other techniques for the following reasons:

- *Boundary sensitive.* Mesh contours closely follow the contours of the boundary. This characteristic is of particular importance since well shaped elements are usually desirable near the boundary.

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

0029–5981/91/120811–37$18.50

Received 15 April 1990
Revised 15 May 1991

- *Orientation insensitive.* Rotating or translating a given geometry does not change the resulting mesh topology. A mesh generated in a transformed geometry is equivalent to the original mesh transformed.
- *Few irregular nodes.* An interior mesh node is considered irregular if more or less than four elements are connected to it. This is a critical mesh topology feature because the number of elements sharing a node controls the final shape of the elements, even after smoothing.[10] Thus, a mesh with few irregular nodes, especially near the boundary where element shape is critical, is often preferred.

Noting these positive characteristics of mapping, we previously attempted to automate quadrilateral meshing by using an expert system, AMEKS,[11] to decompose a geometry into regions for mapping (similar to the approach of experienced analysts.) However, AMEKS suffered from a lack of robustness, long computer run times, and both software and hardware incompatibility with most engineering environments. These concerns led to the development of the paving technique.

The paving technique is designed to automatically generate an all-quadrilateral mesh with qualities similar to those of the mapped mesh technique but without decomposition of the geometry. This is accomplished by layering or paving the geometry with rows of quadrilateral elements from the boundary(s) toward the interior. The technique is relatively fast, robust and capable of handling mesh transitions from small to large elements much more elegantly than mapping techniques. Eventually this technique may be combined with automated decomposition and/or adaptive finite element error correction techniques to produce an *optimal* mesh for a finite element problem. The paving technique has been incorporated into an existing quadrilateral mesh generation program, FASTQ,[12] and is being used in a production environment.

In this paper, we first present the theory of paving. Since paving is accomplished with an interdependent series of steps, the theory is explained with an introductory overview of each step, a general schematic showing how the steps interact, and finally a thorough explanation of each step in turn. Following the theory presentation, we next evaluate the performance of the technique both qualitatively with several examples and quantitatively using several evaluation criteria. Finally, we end with a summary of our conclusions.

2. PAVING THEORY

The paving technique is based on iteratively layering or paving rows of elements to the interior of a region's boundary(s). As shown in the brief sequence in Figure 1, these rows eventually fill the region from the boundary(s) inward. As rows begin to overlap or coincide at the interior of the geometry, they are carefully connected together to form a valid quadrilateral mesh. In this section we first define the terminology and the input requirements for paving, second, give an overview of each of the paving steps, and third, briefly describe how these steps interact (i.e. the control sequence) during the paving process. From there, each of the steps is elaborated on, including (in order of discussion) row choice, closure checks, row generation, smoothing, seaming, row adjustment, perimeter intersection handling and cleanup of the completed mesh.

2.1. Terminology and input requirements

Paving begins with the input of one or more ordered, closed loops of connected nodes that, as shown in Figure 2, form the boundary of the region to be meshed. These initial loops will be referred to as the permanent boundary of the region. The connectivity and location of nodes on

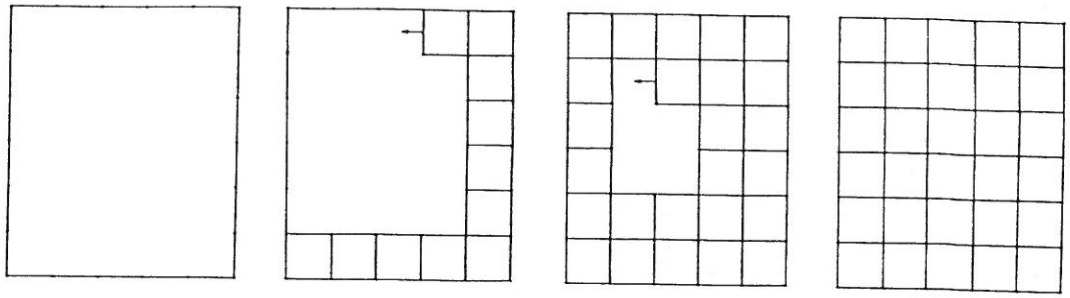

Figure 1. Example of a simple paving sequence

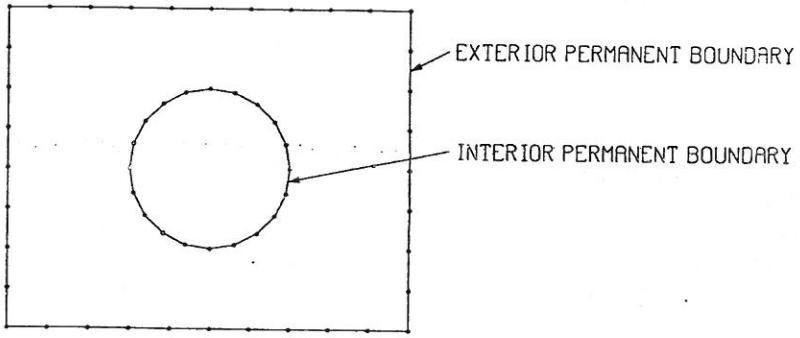

Figure 2. Example of the permanent boundary input for paving

the permanent boundary are not allowed to change during paving. This ensures mesh compatibility with adjacent regions.

Permanent boundaries are categorized as either exterior or interior boundaries. There is always only one exterior permanent boundary for a region. It must be a loop of nodes that is non-intersecting and completely encloses the region to be meshed. It is ordered counterclockwise. Interior permanent boundaries define voids or holes in the region being meshed. They are always ordered clockwise. There may be any number of interior permanent boundaries but each must be completely enclosed within the exterior permanent boundary. No permanent boundary may intersect any other permanent boundary.

During the mesh generation process, the paving technique always operates on boundaries of connected nodes referred to as paving boundaries. Paving boundaries are transient in nature and change as the mesh is generated. As shown in Figure 3, the paving boundaries (highlighted in bold lines) always bound the existing mesh. Initially each paving boundary is identical to a permanent boundary. Elements that are added to the mesh eventually separate the paving boundary(s) from their respective permanent boundary(s).

As with permanent boundaries, paving boundaries are categorized as either exterior or interior paving boundaries. An exterior paving boundary is paved counterclockwise and progresses inward from the exterior boundary. It is always ordered counterclockwise. Although initially there is only one exterior paving boundary in a region, during paving any number may form. Likewise, an interior paving boundary is paved clockwise and progresses outward from an interior permanent boundary.

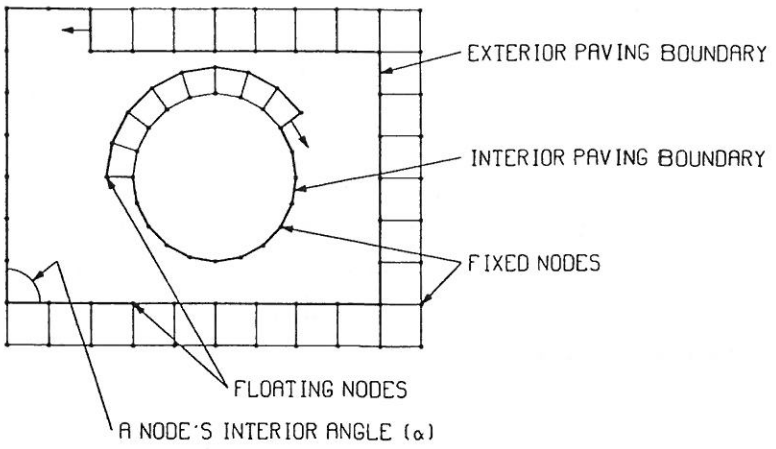

Figure 3. Example of interior and exterior paving boundaries

The nodes in the mesh are also categorized as shown in Figure 3. A paving node is defined as any node on a paving boundary. A fixed node is defined as any node on a permanent boundary. A floating node is defined as any node not on a permanent boundary. Thus, a paving boundary may consist of fixed nodes, floating nodes, or a combination of fixed and floating nodes. Every paving boundary node also has an interior angle which is the angle between a line connecting the node to the preceding node and one connecting it to the next node in the paving boundary, as shown in Figure 3.

Each paving boundary must always contain an even number of nodes. This is a necessary condition when generating an all-quadrilateral mesh. This condition is enforced by controlling the operations performed during paving.

The size of the elements in the mesh is determined by the spacing of nodes on the paving boundary as it propagates. The spacing on a paving boundary is, of course, initially defined by the fixed node spacings on the corresponding exterior boundary. Every attempt is made to maintain this spacing or size as the boundary progresses. Thus, the user can control the element size by modifying the spacing of initial nodes on a permanent boundary. This spacing need not be uniform, and usually is not, as it is often desirable to mesh a region with a gradation of element sizes.

2.2. *Overview of paving steps*

The propagation of a paving boundary involves a number of operations which must be tightly controlled to ensure mesh validity and quality. Each of the paving operations is explained in detail in subsequent sections but may be summarized as follows:

1. *Row choice.* The beginning and ending node of the next sequence or row of elements to be added is found.
2. *Closure check.* A check is made to make sure that more than six nodes remain in the paving boundary. Specific closure techniques are used to conclude meshing for paving boundaries of six or fewer nodes.
3. *Row generation.* The next row of elements identified in the *row choice* step is incrementally added to the boundary.

4. *Smooth.* Floating nodes are adjusted to improve mesh quality and boundary smoothness.
5. *Seam.* Small interior angles in the paving boundary are *seamed* or closed by connecting opposing elements.
6. *Row adjustment.* The new row is adjusted by placing *tucks* or *wedges* into the row to correct for elements becoming too small or too large.
7. *Intersection.* The paving boundary is checked for intersections with itself or with other paving boundaries. Intersections are connected to form new, often separate, paving boundaries.
8. *Cleanup.* The completed mesh is adjusted where element deletion and/or addition improves the overall mesh quality.

2.3. Control sequence

The paving process is an iterative method which uses the steps outlined above. The control of these steps is critical to the formation of a high quality mesh. The control sequence used for paving is diagrammed in Figure 4.

As can be seen in Figure 4, these steps sometimes interact in complex ways. For instance, during a row generation, it may be necessary to suspend creation of the row at critical points, smooth, look for seams and intersections, and then continue with the row generation if appropriate. Also, whenever an intersection occurs, the mesh is smoothed and the boundary is seamed on both sides before continuing to the choice of the next row. Iteration continues until all paving boundaries close, thus completing the mesh.

2.4. Row choice

The first step in the paving technique is determining the location for the next row of elements. It is critical that elements be placed by rows and that those rows be chosen carefully in order to maintain boundary sensitivity, minimize the number of irregular nodes and control the placement

```
repeat
     perform ROW CHOICE
     repeat
          ADD ROW portion
          SMOOTH row portion
          SEAM boundary
          if INTERSECTION occurs then
               connect overlaps
               SEAM boundary
          end if
     until complete row is added
     perform ROW ADJUSTMENT
     if INTERSECTION occurs then
          connect overlaps
          SEAM boundary
     end if
until CLOSURE CHECK is positive
CLEAN-UP MESH
```

Figure 4. Paving control sequence

of irregular nodes. A row is defined by an appropriate beginning and ending paving node. This involves a classification of the paving nodes on the paving boundary, a careful watch for paving boundaries which form simple shapes (e.g. rectangle) or what we have termed primitive regions and the consecutive insertion of new rows. Each of these will be described in turn.

Node classification. To begin the row choice process, each paving node on the current paving boundary is first given an angle status and finally a node classification. The angle status is a categorization that is based solely on the size of the node's interior angle, α (see Figure 3), and the setting of angle tolerances, $\sigma_1, \ldots, \sigma_6$. The angle tolerances represent an allowable deviation from an expected angular value. The angle status categories are:

- *Row end* ($\alpha \leqslant \pi/2 + \sigma_1$). A row end node is a node at which a row terminates. Only one new element will be inserted at a row end node, as shown in Figure 5.
- *Ambiguous row end/side* ($\pi/2 + \sigma_1 < \alpha \leqslant \pi - \sigma_2$). An ambiguous row end/side node may function as either a row end or a row side.
- *Row side* ($\pi - \sigma_2 < \alpha \leqslant \pi + \sigma_3$). A row side node is a node along which the row continues normally. Two new elements are attached to each row side node, as shown in Figure 5.
- *Ambiguous row side/corner* ($\pi + \sigma_3 < \alpha \leqslant 3\pi/2 - \sigma_4$). An ambiguous row side/corner node may function as either a row side or a row corner.
- *Row corner* ($3\pi/2 - \sigma_4 < \alpha \leqslant 3\pi/2 + \sigma_5$). A row corner node is a node at which the row turns a logical corner. Three new elements are attached to each row corner, node as shown in Figure 5.
- *Ambiguous row corner/reversal* ($3\pi/2 + \sigma_5 < \alpha \leqslant 2\pi - \sigma_6$). An ambiguous row corner/reversal node may function as either a row corner or a row reversal.
- *Row reversal* ($\alpha > 2\pi - \sigma_6$). A row reversal node is a node at which the row turns two sequential logical corners, or reverses its direction. Four new elements are attached to each row reversal node, as shown in Figure 5.

These classifications are summarized in Figure 6. All nodes are eventually classified as either a row end, side, corner, or reversal node. As can be seen in Figure 6, the chances of a paving node falling into one of the ambiguous angle status categories are high. These ambiguities are resolved by looking for simple shapes or primitives, or by avoiding irregular node generation as explained below.

Most paving boundaries contain two row end nodes where the next row can start and end. However, there are three special cases of paving boundaries that do not contain two row end

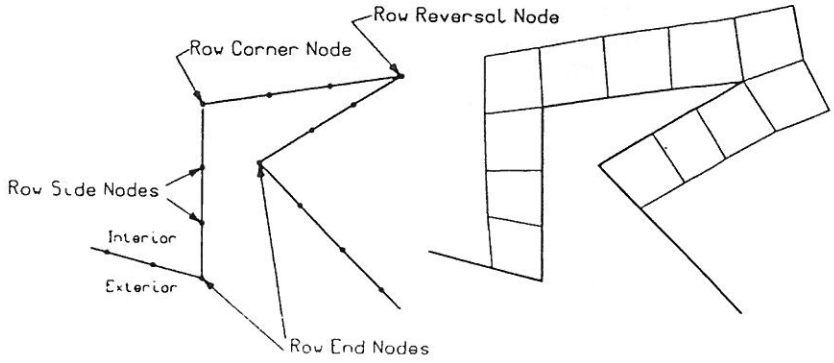

Figure 5. Example boundary node classifications for a new boundary row

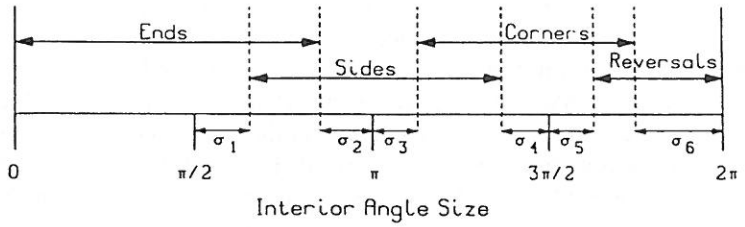

Figure 6. Angle status categories based on paving node's interior angle

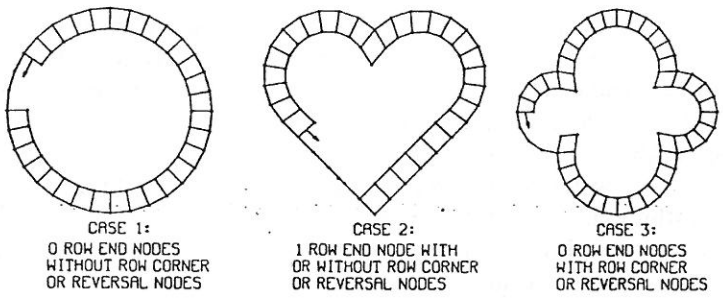

Figure 7. Handling of special case paving boundaries for row choice

nodes and must be handled differently, as shown in Figure 7. The elements shown in Figure 7 indicate how the first row would be inserted in these special case geometries. Case 1 is a paving boundary with only row side nodes (e.g. a circle). In this case the node with the smallest angle is chosen as both the beginning and ending node in the new row to be generated. Case 2 is a paving boundary with only one row end node and any number of row side, corner, or reversal nodes (e.g. a heart shape). In this case the row end node becomes both the start and finish for the new row. Case 3 is a paving boundary with no row end nodes but with some row corners and/or row reversals (e.g. a club shape). In this case the row corner or row reversal node with the smallest interior angle is chosen as the row start and finish node.

Primitive classification. The second factor considered when determining the next row to be paved is the formation of simple or primitive shapes.[13]* Determination of primitive shapes involves the formation of a more global perspective of the current paving boundary. If primitive shapes are identified from this global perspective, the placement of new rows can be adjusted so as to complete the paving of the primitive with a minimum of irregular nodes. The primitives currently used in paving are the rectangle, the triangle and the semicircle. The search for primitives is based on possible combinations of all row ends and row sides in the paving boundary. Since a primitive, by definition, does not contain any row corners or reversals, boundaries that contain row corner or reversal nodes are not evaluated as primitives.

Primitive classification begins with a combinatorial search of all possible row end and row side node interpretations. All nodes which have been categorized as ambiguous row end/side nodes are evaluated as being either a corner or a side for each of the possible interpretations. In order to

* Primitive classifications are considered only when dealing with an exterior paving boundary

preserve the possibility of a primitive shape, any ambiguous row side/corner nodes are always interpreted as sides. Although it is possible that a given region may have a huge number of interpretations, in practice the maximum number is around 10 and the average is 1 or 2.

Each combination of boundary interpretation is evaluated for quality and validity as a primitive. The quality is based on the formation of irregular nodes along the existing boundary, the deviation of the angles from perfect end or side nodes and the number of irregular nodes to be expected within the interior of the resulting primitive as it is meshed. The validity depends on the requirements of each primitive, as documented in Reference 13. The primitive interpretation that is valid and has the highest quality is chosen as the correct interpretation.

For example, the paving boundary shown in Figure 8(a) contains two row end nodes, nodes 1 and 3, two ambiguous row end/side nodes, nodes 2 and 4, and the rest row side nodes. With this set of nodes, there are four possible interpretations:

1. Nodes 2 and 4 may both be classified as row side nodes. In this case a valid semicircle primitive (two end nodes) would be formed. Its quality would be affected by the fact that the interior angles of side nodes 2 and 4 are significantly less than π, the fact that node 4 would become an irregular node, and the fact that a semicircle forms two additional irregular nodes, as shown in Figure 8(b).
2. Node 2 may be classified as a row end node, and node 4 as a row side node. In this case an invalid triangle primitive (three end nodes) is formed.
3. Node 2 may be classified as a row side node, and node 4 as a row end node. Again, an invalid triangle primitive (three end nodes) is formed.
4. Nodes 2 and 4 may both be classified as row end nodes. In this case a valid rectangle primitive (four end nodes) would be formed. Its quality would be affected by the fact that the interior angles of end nodes 2 and 4 are significantly greater than $\pi/2$ and the fact that node 2 would be an irregular node. However, the rectangle does not form any additional irregular nodes, as shown in Figure 8(c), and in this case would be the best interpretation.

If an acceptable primitive evaluation exists, that evaluation is used in determining the placement of the next row. Node classifications are set to those used in the primitive interpreta-

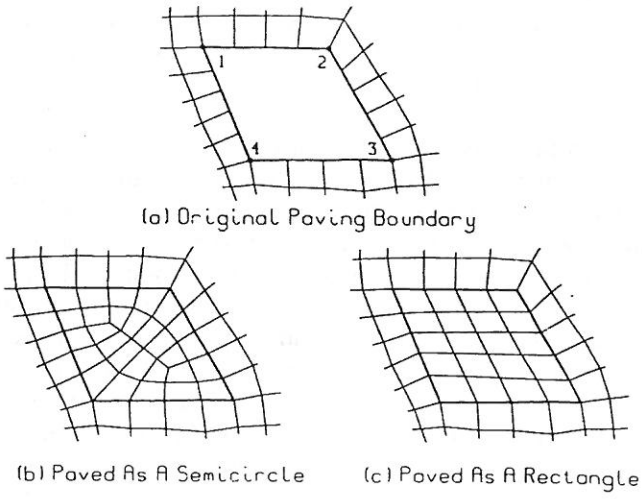

Figure 8. Example of multiple primitive region interpretations for paving

tion and the row end nodes for the next row chosen so as not to invalidate the primitive while maintaining sequential ordering as described in Section 2.4.

This evaluation technique is only a limited global view of the boundary since it does not contain information as to the overall shape, the position and orientation of various boundaries with respect to each other, the proximity of possible overlaps, etc. Because of the limitations in this simple evaluation technique, boundary shapes are often misinterpreted. However, since the interpretation mechanism is repeated as each row is added, the evaluations are self-correcting.

For example, the paving boundary in Figure 9(a) is initially misinterpreted as a rectangle. Adding the first row of elements creates a self-intersecting paving boundary that is subsequently corrected, as shown in Figure 9(b). This forms two exterior paving boundaries which are both correctly interpreted as triangle primitives. These paving boundaries are then filled with only one irregular node each, as shown in the completed mesh in Figure 9(c). As shown with this example, incorrect interpretations of complex shapes as being primitive do not adversely affect the deposition of rows of elements. On the other hand, correct interpretations of simple shapes often minimize the number of irregular nodes.

When an acceptable primitive evaluation does not exist, all ambiguous nodes are classified based on angle size and the possibility of generating an irregular node. For example, if classifying an ambiguous side/corner node as a side node would form an irregular node while a corner classification would not, the corner classification is allowed a greater angular tolerance.

Consecutive ordering of rows. The final concern in the choosing of rows is that the paving boundary be propagated successively inward away from all permanent boundaries. This is accomplished by preferring new rows that are sequential to (i.e. topologically *next to*) the last row installed in the paving boundary. Rows are initially installed around exterior paving boundaries sequentially counterclockwise. After each complete traversal of the exterior paving boundary, each interior paving boundary is paved with one complete traversal sequentially clockwise.

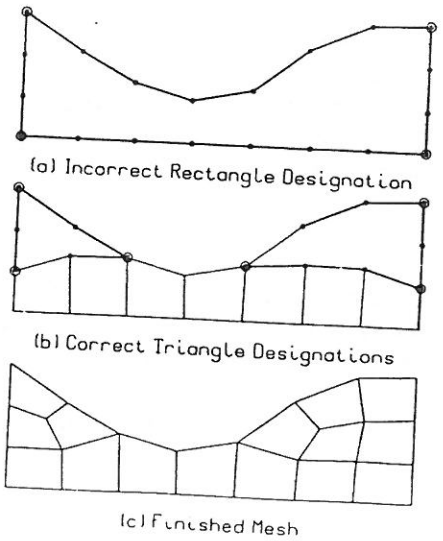

Figure 9. Example of incorrect primitive region interpretation

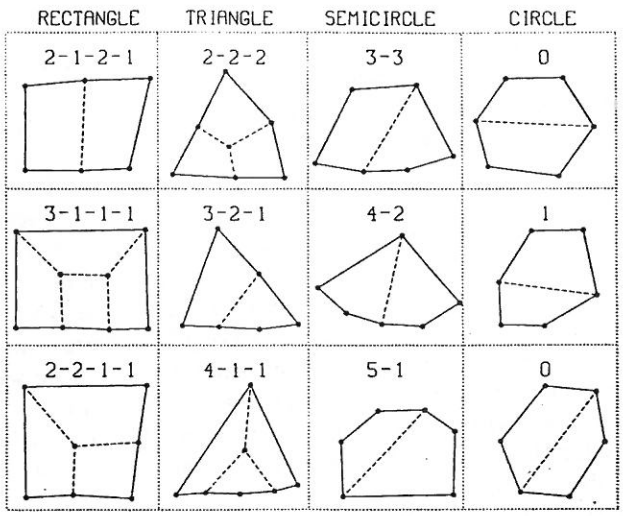

Figure 10. Closure of boundaries with six nodes

Although rows may not form true rings inside the boundary, the sequential ordering of row placement generally forces boundaries to close toward the interior of the region. This tends to place irregular nodes interior to the geometry, away from the more critical boundary elements.

2.5. *Closure check*

Before any new rows of elements are added to the mesh, a check is made to see if the paving boundary contains six or less nodes. If the boundary contains zero nodes, paving of the boundary has been completed. If the boundary contains two nodes, paving can be completed with a simple seam, as explained in Section 2.8. If the boundary contains four nodes, one additional quadrilateral element is added to close the boundary. Closure for boundaries containing six nodes is based on the configuration of those nodes and may close to form two, three, or four new elements, as shown in Figure 10. The numbers above each case in the figure indicate how many element edges are on each side of the shape (e.g. in a 3–2–1 triangle, the triangle's first side is formed from the combination of three element sides, the second from two element sides, and the third from one element side). Some of the elements shown in Figure 10 are poor quadrilateral elements. However, because closures usually form in the centre of the mesh, subsequent smooths and cleanup almost always improve or eliminate the problem elements.

2.6. *Row generation*

Once the limits of the new row have been chosen, actual generation can begin. Row generation is accomplished by projecting new nodes away from each sequential boundary node at an appropriate angle and distance. This projection is based on the node classification (Section 2.4) and the distance between immediately preceding and subsequent boundary nodes. Elements and nodes are always generated sequentially along the row. Projections from row side, corner and reversal nodes, as well as from one additional special case, are described in the present section.† Finally, row termination is discussed.

† Row end nodes form the basis for either beginning or ending a row and as such they are not projected

Row side node projection. The projection from a row side node is shown in Figure 11. Let the existing side node be N_i. The new node, N_j, is placed at the tip of a vector V whose origin is at N_i. V is oriented to bisect the interior angle, α, at N_i. Let d_1 be the distance from N_i to N_{i-1} and d_2 the distance from N_i to N_{i+1}. Then the length of the vector V, $|\mathbf{V}|$, is given by

$$|\mathbf{V}| = \frac{(d_1 + d_2)/2}{\sin(\alpha/2)} \tag{1}$$

The sine term in equation (1) tends to smooth out otherwise *bumpy* rows. One new element defined by the nodes N_i, N_{i-1}, N_{j-1} and N_j is added to the mesh.

As can be seen in Figure 11, the projection from a row side node (N_i) usually transforms the node into a row end node, effectively moving the row end forward one step.

Row corner node projection. The projection from a row corner node is shown in Figure 12. Let the existing corner node again be N_i. Three new nodes, N_j, N_k and N_l, are placed at the tips of

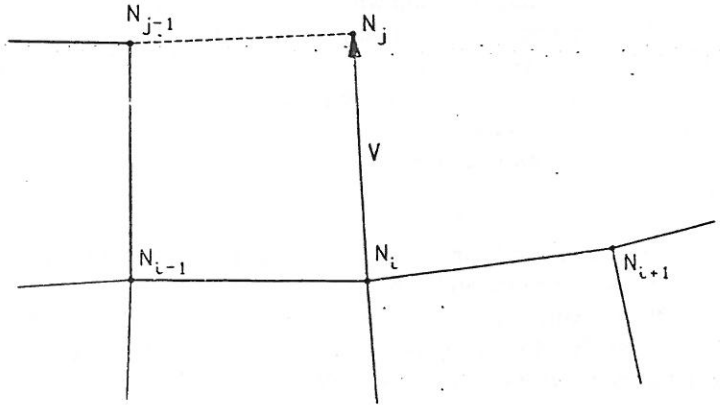

Figure 11. Projection of a new node from a row side node

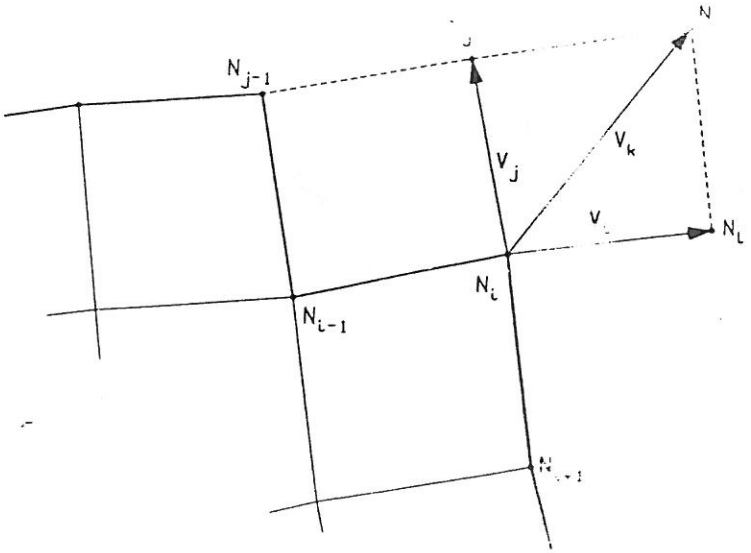

Figure 12. Projection of new nodes from a row corner node

vectors $\mathbf{V}_j$, $\mathbf{V}_k$ and $\mathbf{V}_l$ respectively. All three vectors originate at N_i. $\mathbf{V}_j$, $\mathbf{V}_k$ and $\mathbf{V}_l$ are oriented at respectively the clockwise 1/3, 1/2 and 2/3 bisections of the interior angle of N_i. As before, let d_1 be the distance from N_i to N_{i-1} and d_2 the distance from N_i to N_{i+1}. The magnitudes of the vectors are given by

$$|\mathbf{V}_j| = \frac{(d_1 + d_2)/2}{\sin(\alpha/3)} \tag{2}$$

$$|\mathbf{V}_k| = \sqrt{2}|\mathbf{V}_j| \tag{3}$$

$$|\mathbf{V}_l| = |\mathbf{V}_j| \tag{4}$$

Two new elements are added to the mesh: the first defined by nodes N_i, N_{i-1}, N_{j-1} and N_j, and the second defined by nodes N_i, N_j, N_k and N_l.

As alluded to in the overview in Section 2.2, after the first element has been added at a row corner, the operations of smoothing, seaming and intersection checking are performed. This is necessary because in some cases a seam of the completed portion of the row may be appropriate. Such a seam may place node N_i interior to the mesh, preventing continuation of the row. Likewise, an intersection of the row portion may be found which forms new boundaries and may either place N_i interior to the mesh or on a newly formed boundary. Again this would pre-empt continuation of the row. In such a situation, control transfers back to the row choice procedure described in Section 2.4 for determination of a new row to be paved.

As can be seen in Figure 12, the projection from a row corner node (N_i) not only propagates the row end around the corner, but introduces another new row corner node (N_k) into the paving boundary.

Row reversal node projection. The projection from a row reversal node is shown in Figure 13. Let the existing corner node again be N_i. Five new nodes, N_j, N_k, N_l, N_m and N_n, are placed at the tips of vectors $\mathbf{V}_j$, $\mathbf{V}_k$, $\mathbf{V}_l$, $\mathbf{V}_m$ and $\mathbf{V}_n$ respectively. All five vectors originate at N_i. Here $\mathbf{V}_j$, $\mathbf{V}_k$, $\mathbf{V}_l$, $\mathbf{V}_m$ and $\mathbf{V}_n$ are oriented at respectively the clockwise 1/4, 3/8, 1/2, 5/8 and 3/4 bisections of the interior angle of N_i. As before, let d_1 be the distance from N_i to N_{i-1} and d_2 the distance from N_i

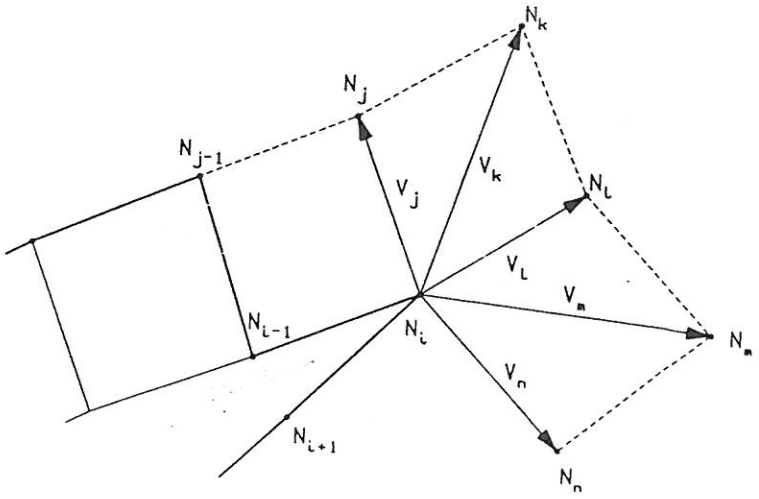

Figure 13. Projection of new nodes from a row reversal node

to N_{i+1} respectively. The magnitudes of the vectors are given by

$$|\mathbf{V}_j| = \frac{(d_1 + d_2)/2}{\sin(\alpha/4)} \tag{5}$$

$$|\mathbf{V}_k| = \sqrt{2}|\mathbf{V}_j| \tag{6}$$

$$|\mathbf{V}_l| = |\mathbf{V}_j| \tag{7}$$

$$|\mathbf{V}_m| = |\mathbf{V}_k| \tag{8}$$

$$|\mathbf{V}_n| = |\mathbf{V}_j| \tag{9}$$

Three new elements are added to the mesh: the first defined by nodes N_i, N_{i-1}, N_{j-1} and N_j, the second defined by nodes N_i, N_j, N_k and N_l, and the third element defined by nodes N_i, N_l, N_m and N_n.

Again, as mentioned for a row corner node projection, after the first and again after the second element has been added a smooth, seam and intersection check is performed to insure that continuation of the row is beneficial.

As can be seen in Figure 13, a row reversal node projection usually produces two new row corner nodes. This means that a row reversal node can occur only on the permanent boundary as the first row immediately buries the very large interior angle.

All side node row projection. Of the special cases mentioned in Section 2.4, only one, the case of a row with only row side nodes, requires an additional projection scheme. The projection to begin a row with only side nodes is shown in Figure 14. Let the beginning (and ending) node by N_i. Two new nodes, N_j and N_k, are placed at the tips of vectors $\mathbf{V}_j$ and $\mathbf{V}_k$ respectively. $\mathbf{V}_j$ originates at N_i and $\mathbf{V}_k$ originates at N_{i+1}. $\mathbf{V}_j$ is oriented to bisect the interior angle, α, at N_i. $\mathbf{V}_k$ is oriented to bisect the interior angle, β, at N_{i+1}. Let d_1 be the distance from N_i to N_{i-1}, d_2 be the distance from N_i to N_{i+1}, and d_3 be the distance from N_{i+1} to N_{i+1}. The magnitudes of the vectors are given by

$$|\mathbf{V}_j| = \frac{(d_1 + d_2)/2}{\sin(\alpha/2)} \tag{10}$$

$$|\mathbf{V}_k| = \frac{(d_2 + d_3)/2}{\sin(\beta/2)} \tag{11}$$

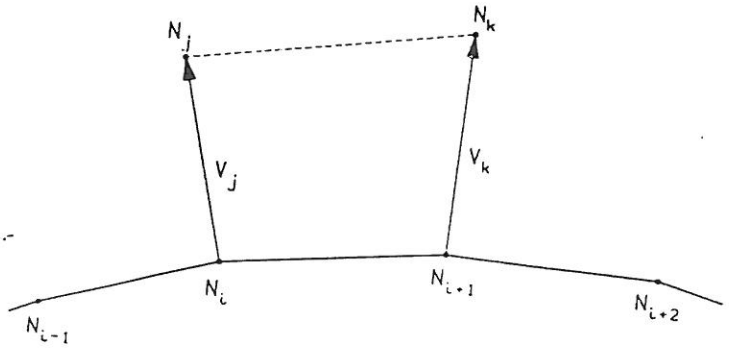

Figure 14. Projection of new nodes from an all side node row

One new element, defined by nodes N_i, N_j, N_k and N_{i+1}, is added to the mesh.

As can be seen in Figure 14, the projection from an all side node row generates two row end nodes which serve to propagate and eventually terminate the forming row.

Row termination. With each of the projections described above, the node projected from them becomes, in effect, a new row end node. Thus when the row has progressed such that the next node, N_i, in the row is the row's predetermined ending node, the row is terminated by simply closing the final element in the row. The process of row termination is shown in Figure 15. The final element in the row is defined by nodes N_{i-1}, N_j, N_{i+1} and N_i.

2.7. *Smoothing*

Smoothing is the most used tool during the paving process. It is used not only after rows are added and intersections completed, as shown in the control sequence in Figure 4, but essentially after every modification to the mesh. Local operations involving only a few nodes such as projection during row generation, seaming, intersections, etc., tend to distort the neighbouring mesh and paving boundaries. The smoother attempts to restore and maintain element size, perpendicularity, and overall paving boundary and mesh smoothness; a set of criteria that are often conflicting.

The smoothing requirements for paving differ from other mesh generation applications in that a *free* boundary must be maintained as the paving boundary moves from the *fixed* permanent boundary toward the interior of the region. Thus, for paving we developed a customized three-step smoother. First, all nodes not on the paving boundary are fixed and a smooth is applied to any floating nodes on the paving boundary. This is called the paving boundary smooth. Second, following the boundary smooth, all paving boundary nodes are fixed and all floating nodes, which are not paving boundary nodes, are smoothed. This is called the interior smooth. Finally, the boundary smooth is repeated. In this section we first describe the paving boundary smooth, next the interior smooth, and finally a localization mechanism to reduce the cost of smoothing.

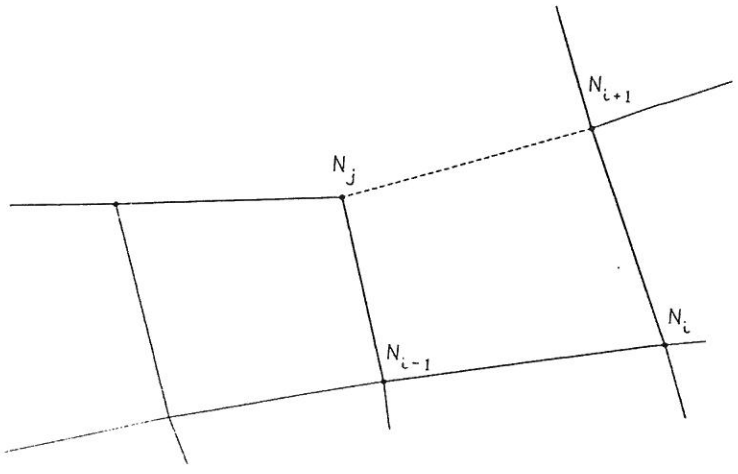

Figure 15. Row termination

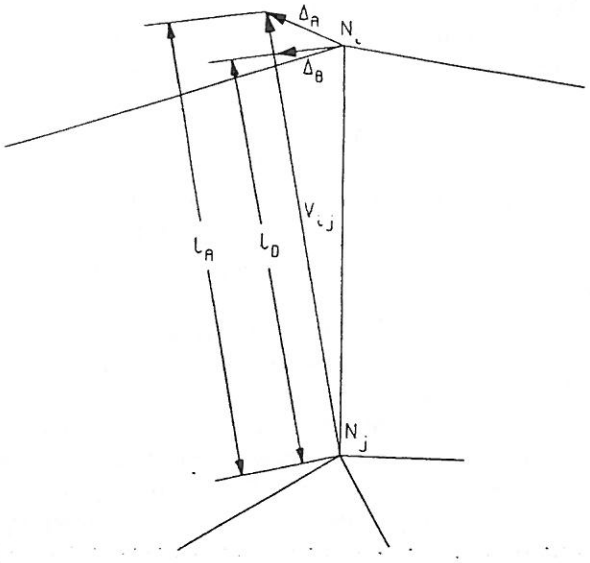

Figure 16. Length adjustment to row nodes

Boundary node smoothing. As mentioned above, the boundary smooth is limited to nodes on the current paving boundary that are not part of the original exterior boundary.[‡] The boundary smoother is a modified isoparametric smooth.[14] To begin the smoothing process, the node movement based on a true isoparametric smooth is calculated. Let $\mathbf{V}_i$ be the vector from the origin to node N_i. Assuming that N_i is attached to n elements, let $\mathbf{V}_{mj}$, $\mathbf{V}_{mk}$ and $\mathbf{V}_{ml}$ be vectors from the origin to respectively nodes N_j, N_k and N_l of the mth element. The nodes must be in a clockwise or counterclockwise order around the element. A new vector from the origin to the proposed new location of node N_i, vector $\mathbf{V}'_i$, is given as

$$\mathbf{V}'_i = \frac{1}{n}\sum_{m=1}^{n} \mathbf{V}_{mj} + \mathbf{V}_{ml} - \mathbf{V}_{mk} \qquad (12)$$

Let the vector Δ_A define the change in location of node N_i for a true isoparametric smooth. This vector is then simply

$$\Delta_A = \mathbf{V}'_i - \mathbf{V}_i \qquad (13)$$

The isoparametric smoother adjusts nodes such that the elements are more nearly parallelpipeds. However, it does not necessarily maintain the desired element size or the desired *squareness* of an element. This becomes a problem for paving nodes that are attached to two elements.[§] For such paving nodes the change in node location is modified to reflect a desired length and a desired angular smoothness.

First we will examine the desired length criteria. Let node N_j be the node opposite node N_i on the common boundary between the two elements attached to node N_i, as shown in Figure 16. Again, let $\mathbf{V}_i$ and $\mathbf{V}_j$ be the vectors from the origin to node N_i and N_j respectively. Let l_D be node

[‡] Original exterior boundary nodes are never free to move

[§] These are usually boundary side nodes. As such, they constitute a majority of most boundary nodes during paving

N_i's desired length, which is the length of the projection vector used to initially generate the node as defined by equations (1), (2) and (5). Let l_A be the length of a vector $\mathbf{V}_{ij}$ from N_j to the new location of N_i determined from the isoparametric smooth (i.e. to the tip of vector $\mathbf{V}_i'$ in equation (12)). A modified change in position vector, Δ_B, is now defined as

$$\Delta_B = \mathbf{V}_j - \mathbf{V}_i + (\Delta_A + \mathbf{V}_i - \mathbf{V}_j)\frac{l_D}{l_A} \tag{14}$$

This adjustment can be understood as simply a length adjustment of the vector $\mathbf{V}_{ij}$ without adjusting its orientation, as shown in Figure 16. The length adjustment helps protect the desired size of the element being generated as well as the aspect ratio. However, the perpendicularity of element corners is not guaranteed.

The final adjustment during a paving row smooth is the desired angular smoothness. Again let N_j be the node opposite N_i on the common boundary, as shown in Figure 17. Let N_{i-1} and N_{i+1} be respectively the preceding and following nodes to N_i on the boundary. Let the vectors $\mathbf{P}_i$, $\mathbf{P}_{i-1}$ and $\mathbf{P}_{i+1}$ be vectors from node N_j to nodes N_i, N_{i-1} and N_{i+1} respectively. A new vector, $\mathbf{P}_{B1}$, is defined which bisects the angle between $\mathbf{P}_{i-1}$ and $\mathbf{P}_{i+1}$. Let a new vector, $\mathbf{P}_{B2}$, now be found. This vector will determine the adjusted location of N_i. The tail of this vector is placed at N_j. The direction of the vector $\mathbf{P}_{B2}$ is defined along the angle bisector of the angle between $\mathbf{P}_{B1}$ and $\mathbf{P}_i$.

Next the length of vector $\mathbf{P}_{B2}$ must be determined. First, the location of a new point, Q, is calculated as the intersection between $\mathbf{P}_{B2}$ extended, and a line connecting nodes N_{i-1} and N_{i+1}. Let the length from Q to N_j be l_Q and the original projection length of N_i be l_D as before. The length of vector $\mathbf{P}_{B2}$, $|\mathbf{P}_{B2}|$, is then defined based on the relative lengths of l_D and l_Q:

$$|\mathbf{P}_{B2}| = \begin{cases} \dfrac{l_Q + l_D}{2} & \text{if } l_D \geq l_Q \\ l_D & \text{otherwise} \end{cases} \tag{15}$$

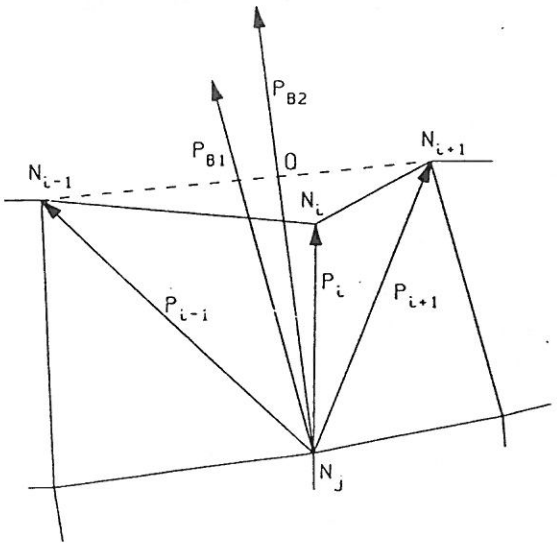

Figure 17. Angular smoothness adjustment to row nodes

The new location for N_i based on angular smoothness has now been defined. The change in position vector for angular smoothness, Δ_C, is thus defined as

$$\Delta_C = \mathbf{P}_{B2} - \mathbf{P}_i \tag{16}$$

This angle smoothness adjustment functions to keep elements perpendicular at corners and helps to maintain a *smooth* paving boundary.

The final change in location vector, Δ_i, for node N_i attached to only two elements on the boundary can now be defined as

$$\Delta_i = \frac{\Delta_B + \Delta_C}{2} \tag{17}$$

For nodes on the boundary which are attached to more or less than two elements, the change in location is based solely on the isoparametric smoother vector change, Δ_A, as defined in equation (13). Although somewhat complex to implement, the row smooth is critical to the correct formation of rows. It also converges quickly, usually in three or four iterations.

Interior node smoothing. Following the initial paving row smooth, all boundary nodes are fixed and the remaining floating nodes are smoothed. A modified length-weighted Laplacian[15] smoother is used. This technique uses a set of vectors from an interior node to all its attached nodes. Let a contributing vector, $\mathbf{C}_j$, be the vector contribution of an attached node, N_j, when smoothing the interior node, N_i. In defining $\mathbf{C}_j$, let $\mathbf{V}_j$ be the vector from node N_i to node N_j, as shown in Figure 18. If node N_j is not on a permanent boundary, then the contributing vector $\mathbf{C}_j$ is equivalent to $\mathbf{V}_j$. If node N_j is on a permanent boundary then $\mathbf{C}_j$ is defined as

$$\mathbf{C}_j = \mathbf{V}_j + \Delta_{C_j} \tag{18}$$

where the vector Δ_{C_j} is defined using the angular smoothness criteria defined in equation (16).

Now, let the vector defining the change in location of N_i be Δ_i. This vector, Δ_i, is defined as a sum of all the contributing vectors from attached nodes weighted by the length of the

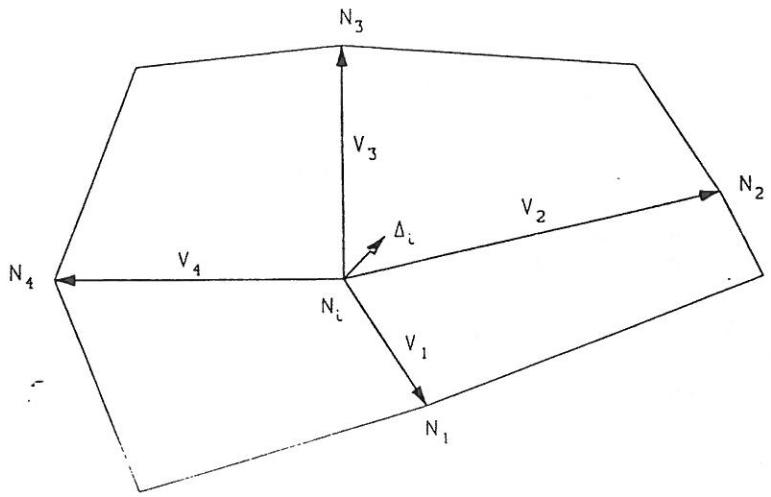

Figure 18. Modified length-weighted Laplacian smoothing applied to interior nodes

contributing vectors. If there are n nodes attached to N_i as shown in Figure 18 then

$$\Delta_i = \frac{\sum_{j=1}^{n} |\mathbf{C}_j| \mathbf{C}_j}{\sum_{j=1}^{n} |\mathbf{C}_j|} \tag{19}$$

This modification to the length-weighted Laplacian tends to keep elements along the exterior of the region perpendicular to the boundary. This smoother also tends to move nodes an equal distance from its attached neighbours and converges rapidly.

Localization of smoothing. Since smoothing is an often repeated and fairly expensive procedure, a localized smooth is used to minimize this cost. If a smooth mesh is changed through some local operation it is reasonable to assume that, during the subsequent smooth, nodes closer to the disturbance will require more adjustment than those more distant. In fact, nodes buried more than three element layers from the disturbance rarely move significantly. Thus, in order to save computer time, smoothing has been restricted to nodes connected within three element layers to any node involved in a disturbance. For instance, when a row is added, only those nodes three rows deep into the existing mesh are smoothed. This simple localization has improved run times significantly.

2.8. Seaming

The process of seaming may be understood as simply a process to close *cracks* in the paving boundary which often form during the paving process. *Crack tips* are easily recognized as boundary nodes with very small interior angles.¶

The criterion to seam a crack tip is based on the interior angle of the node, α, the number of attached element edges, N_E, and whether the node is a fixed node (i.e. on a permanent boundary). There are three types of seams which must be considered: (i) a seam of a floating paving boundary node, (ii) a seam of a fixed node, and (iii) a seam between two element sides of unequal length (a transition seam).

Interior node seams. The seaming of floating nodes on a paving boundary is the most common type of seam. Such nodes may be seamed if

$$\left.\begin{cases} \alpha < \varepsilon_1 & \text{for } N_E \geqslant 5 \\ \alpha < \varepsilon_2 & \text{otherwise} \end{cases}\right\} \quad \text{where } \varepsilon_1 < \varepsilon_2 \tag{20}$$

As seen in equation (20) above, the seaming tolerance decreases as the number of attached elements increase. This tolerance shift favours seams which will form a regular node or reduce the irregularity of the node at the *crack tip*. For example, the node, N_i, in Figure 19(a) is attached to four element edges ($N_E = 4$). Seaming this node, as shown in Figure 19(b), changes N_i to an irregular node which is attached only to three elements. Thus, in general, the interior angle of a node attached to four element edges must be relatively small for this type of seam to be beneficial. In this case the angle at node N_i was negative and seaming was necessary. On the other hand, a seam of node N_i, shown in Figure 19(c), which is attached to five element edges, changes N_i into a regular interior node attached to four elements as shown in Figure 19(d). This type of seam is thus allowable for much larger interior angles.

¶ It is not unusual for rows to cross at their ends, thus forming nodes with negative interior angles

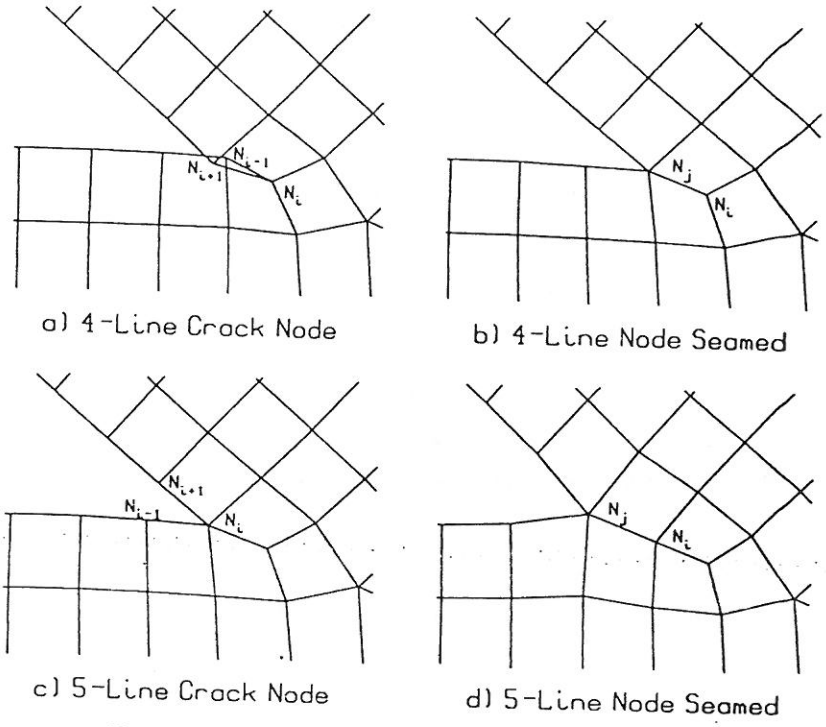

Figure 19. Seaming floating node cracks in the paving boundary

As can be seen in Figure 19, each seam is performed by simply combining the node preceding N_i, N_{i-1}, and the node following N_i, N_{i+1}, into a single node, N_j. This also reduces the number of element edges, N_E, attached to node N_i by one. The example seam shown in Figure 19 also shows that once a crack begins to close (19(b)) the seam tends to continue until a fairly square corner is formed (19(d)).

Fixed node seams. The seaming of cracks involving fixed nodes on the permanent boundary must be performed more judiciously. Since exterior nodes are not free to move, smoothing cannot improve elements which have two or more sides on the exterior boundary. Thus, care must be taken that a seam will not create elements with bad interior angles which cannot be improved. This is primarily a concern when the exterior boundary forms small angles. As shown below, the problem is overcome by simply delaying the closure of cracks where this becomes a problem.

Figure 20(a) shows an example of the fixed node seam problem. The first row placed into the small angle portion of the permanent boundary forms a very small interior angle at node N_i. However, seaming node N_i would create a degenerate quadrilateral element with an interior angle of $\approx \pi$. As shown in Figure 20(b), the seam of this crack is thus ignored and the next row inserted as if the small crack did not exist. This, of course, causes the new row to seriously overlap. But now the nodes to be seamed are all floating nodes and can be smoothed as needed. The mesh resulting from this delayed seam is shown in Figure 20(c). For geometries such as this which contain small interior angle portions, the quadrilateral mesh generated by the paving algorithm is almost always similar to the mesh shown in Figure 20(c).

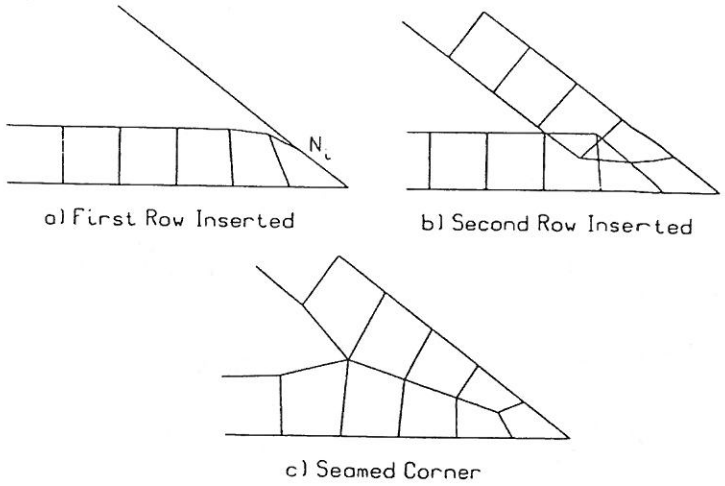

Figure 20. Seaming fixed node cracks in the paving boundary

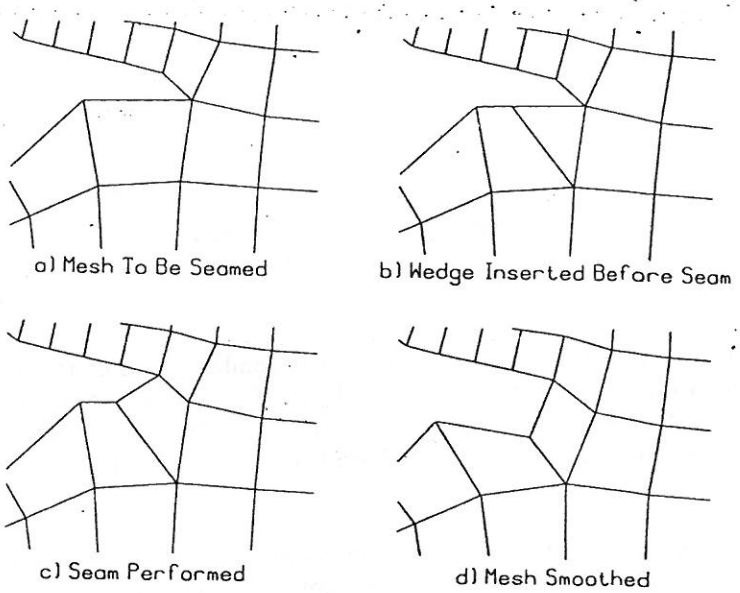

Figure 21. Seaming disproportionate elements in the paving boundary

Transition seams. Often, for paving boundaries with transitioning requirements, a seam may be required between two element sides of disproportionate sizes, as shown in Figure 21(a). Thus, before any seam is performed, the relative lengths of the two element sides to be seamed are compared. If the ratio of the longer side to the shorter side is greater than a threshold value, currently set at 2·5, then a *wedge* element, as will be explained in Section 2.9, is added to the mesh. The wedge is positioned so as to effectively cut the longer element side into thirds, as shown in Figure 21(b).[‖] This new arrangement of elements is then seamed, as shown in Figure 21(c), and the

[‖]The wedge or triangular looking element in Figure 21(b) is actually a quadrilateral with one of the nodes at the midpoint of the triangle's top side

result of such a seam after smoothing is shown in Figure 21(d). As can be seen in this example, insertion of a wedge during the seaming operation allows graceful transitions in otherwise difficult boundary closure situations.

2.9. *Row adjustment*

As rows are added to the mesh, the geometry, as well as the topology of the row, affect the elements being formed. For row portions that traverse a concave segment of the boundary, the elements in each successive row tend to become smaller. Likewise, if the row portions traverse a convex segment of the boundary, elements tend to become larger with each additional row. This undesired expansion or contraction of element sizes, depicted in Figure 22, can be corrected with appropriate row adjustments. This section describes the techniques used to recognize and correct these phenomena: first the insertion of *wedges* in expanding rows, and second the formation of *tucks* in contracting rows.

Insertion of wedges. Wedges are used to check the growth of the size of elements in portions of a boundary row. The wedge insertion process can be visualized as cracking the row at a floating paving boundary node and forcing an additional element into the crack. This new element essentially forms a wedge which separates two previously connected elements and forms a row corner node in the row.

In practice wedge insertion entails the generation of two new nodes and one new element. Referring to Figure 23(a), for the insertion of a wedge at node N_i, a new node, N_i', is generated and used to replace N_i in the element definition of the element on the right of N_i. Node N_i is repositioned on an imaginary line joining node N_{i-1} and the original position of N_i at a distance of 1/3 the original distance from N_i to N_{i-1}. Likewise, node N_i' is positioned on an imaginary line joining the original position of N_i and node N_{i+1} at a distance of 1/3 the original distance from N_i to N_{i+1}. This forms the crack referred to earlier. The second new node, N_k, is then placed at a

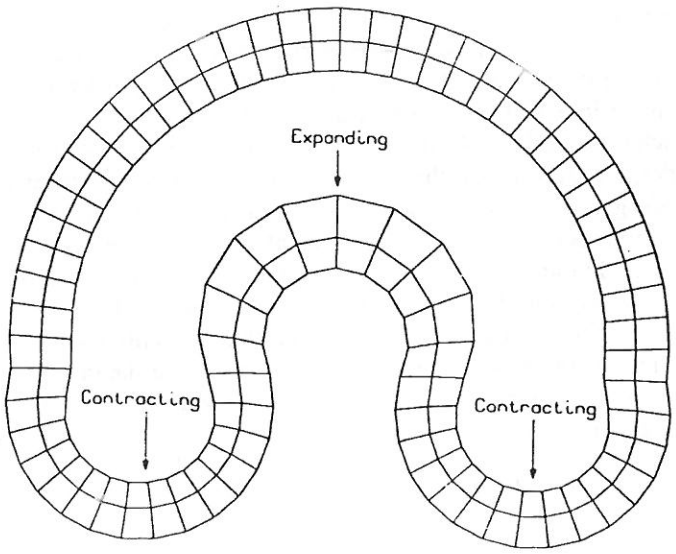

Figure 22. An example of a row with portions of contracting and expanding element sizes

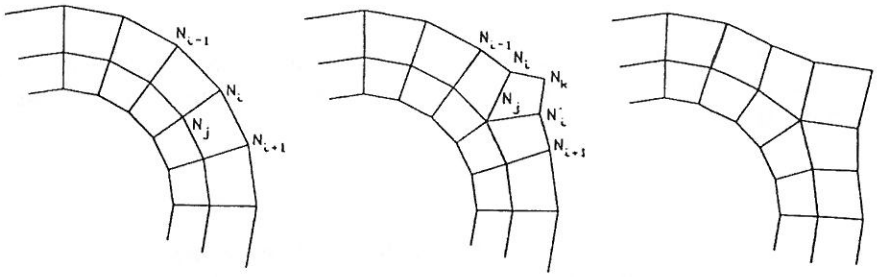

Figure 23. The insertion of a wedge to correct element size expansion

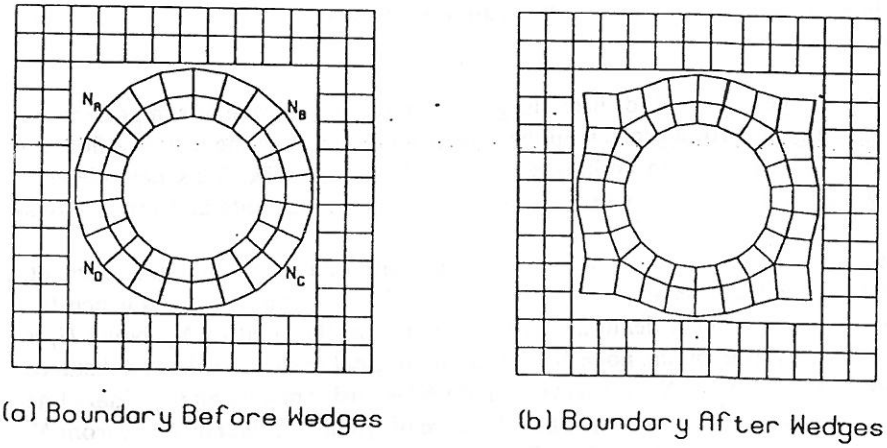

Figure 24. Example wedge insertion in a boundary row portion

location defined using the isoparametric smoothing equation, equation (12), using nodes N_i, N_j and N_i' in the equation. The new element is then defined by nodes N_i, N_j, N_i' and N_k, as shown in Figure 23(b). Smoothing is then performed and the results are shown in Figure 23(c).

Row portions where wedge insertions would be beneficial are detected based on two criteria: first, on an expansion ratio which is the ratio of the desired length* to the actual distance between adjacent paving boundary nodes, and second, on the interior angle of the node. If the expansion ratio and interior angle are both greater than threshold values,† then the node *qualifies* for the insertion of a wedge. The actual number of wedges and their location is based on the number of sequential paving boundary nodes that *qualify* for wedge insertion, and the total angle traversed by the boundary for this sequential set of nodes. The optimum wedge placement would be at the midpoint of every $\pi/2$ portion of the total angle traversed by this sequential set of paving boundary nodes. For example, in Figure 24(a), all paving boundary nodes in the interior paving boundary (around the interior hole) *qualify* for wedge insertion. Since the total angle traversed by this sequence of eligible nodes is 2π, and since there are more than four nodes in the sequence, four

*The desired length is usually propagated from the boundary inward and originates as the distance between nodes along the permanent boundary

†The expansion ratio threshold is currently set at 1·25 while the interior angle threshold is 183°

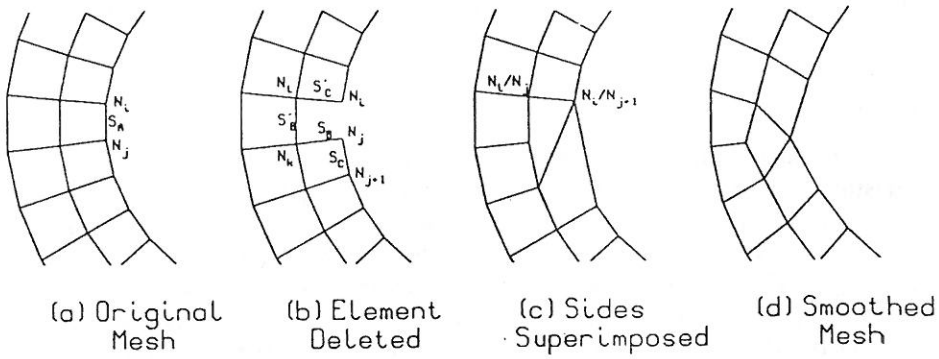

Figure 25. The formation of a tuck to correct element size reduction

wedges are inserted at nodes N_A, N_B, N_C and N_D, as shown in Figure 23(b). The nodes where wedges are placed are located at the clockwise 1/8, 3/8, 5/8 and 7/8 points (midpoints of each $\pi/2$ portion) of the traversed angle. No attempt is made to insure alignment of these wedges with other existing boundaries, although in this example alignment does occur. After all of the wedges have been inserted into the row portion, a localized smooth is performed.

Formation of tucks. Tucks are used to check the reduction of the size of elements in portions of a boundary row. The tuck formation process can be visualized as combining two adjacent row elements into one. This combination process usually creates a row end node where three row side nodes previously existed.

A tuck formation eliminates two boundary nodes and one element. Referring to Figure 25, for the formation of a tuck at node N_i, an element must first be deleted. Let element side S_A be the element side connecting nodes N_i and N_j. The element attached to side S_A is the one deleted, as shown in Figure 25(b). Let element side S_B connect nodes N_k and N_j, element side S'_B connect nodes N_l and N_k, element side S_C connect nodes N_j and N_{j+1}, and element side S'_C connect N_i and N_l. The tuck is completed by superimposing sides S_B and S_C on sides S'_B and S'_C respectively, as shown in Figure 25(c). This ... replaces node N_j with node N_l and node N_{j+1} with node N_i within all associated elements. A smooth completes the tuck and produces the mesh shown in Figure 25(d).

As with wedges, row portions where tuck formations would be beneficial are detected based on two criteria: first, on a reduction ratio which is the ratio of the desired length to the actual distance between adjacent boundary nodes, and second, on the interior angle of the node. If the reduction ratio and interior angle are both less than threshold values,[‡] then the node *qualifies* for the formation of a tuck. Again, as with wedges, the actual number of tucks and their location is based on the number of sequential boundary nodes that *qualify* for tuck formation and the total angle traversed by the boundary for this sequential set of nodes. The optimum tuck placement would be at the midpoint of every $\pi/2$ portion of the total angle traversed. For example, in Figure 26(a), three sequential sets of nodes exist which *qualify* for tuck formation. The first set is the top right boundary node N_1, the second set is the left loop counterclockwise from node N_2 to N_3, and the third set is the lower right boundary node N_4. Of course, since the first and third set are composed of only one node, only one tuck is taken at the node in each set. The angle traversed by the second sequential set is $\approx 3\pi/2$ and there are more than three nodes in the sequence. Thus

[‡] The reduction ratio threshold is currently set at 0·8 while the interior angle threshold is 177°

three tucks are formed at nodes N_A, N_B and N_C, as shown in Figure 23(b). Nodes N_A, N_B and N_C are at the counterclockwise 1/6, 1/2 and 5/6 points (midpoints of each $\approx \pi/2$ portion) of the traversed angle, respectively.

2.10. *Perimeter intersection handling*

As paving boundaries propagate into the region, it is common to have these boundaries intersect or overlap existing portions of the mesh. These intersections may be associated with a negative interior angle and thus be completely eliminated during the seaming operation, as previously shown in Figure 19. However, many times the overlap is simply due to the original geometry. As shown in Figure 27, the overlap may be of two types: (i) a single paving boundary crossing itself, and (ii) a paving boundary crossing another paving boundary (e.g. holes within a region). This section describes the mechanism for detecting and correcting these two types of intersections.

Single paving boundary intersection. When a paving boundary overlaps itself, as shown in Figure 27(a), there are at least two pairs of lines (connections between boundary nodes) that

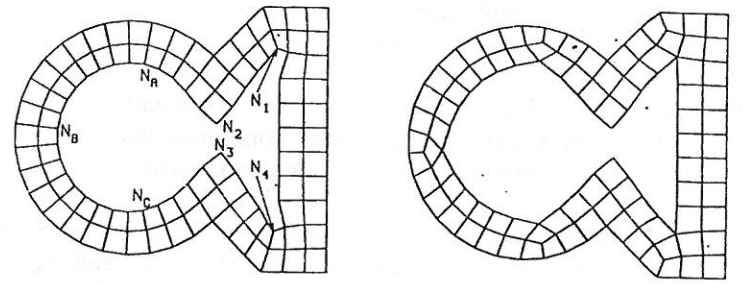

Figure 26. Example tuck formation in a boundary row portion

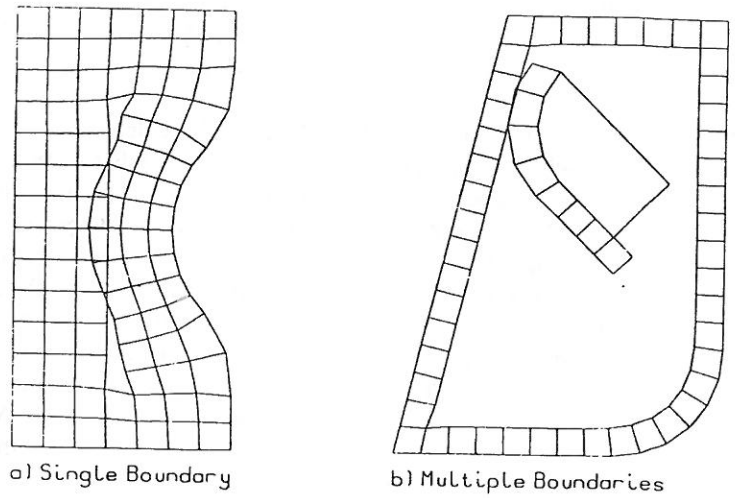

Figure 27. Examples of perimeter intersections requiring adjustment

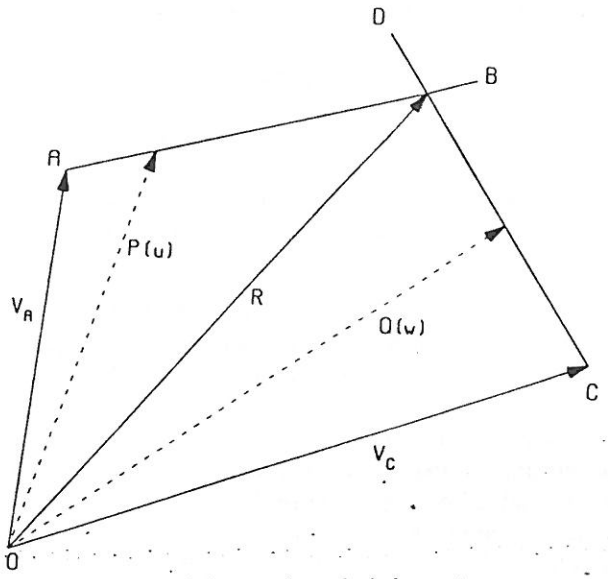

Figure 28. Intersection calculation vectors

intersect each other. The intersection is detected by calculating the intersection of two straight line segments (Reference 16, pages 319, 320). The intersection calculation formula used is fast and informative, providing not only the verification of an intersection but also the location of the intersection and the proportional lengths of the four line segments that would be formed if the lines were divided at the intersection. As shown in Figure 28, let the vector $\mathbf{V_A}$ be the position vector from the origin to end point A of line segment $\overline{AB}$. Let vector $\mathbf{V_B}$ be a vector from end point A to end point B of line segment $\overline{AB}$. Likewise, let $\mathbf{V_C}$ be the position vector of end point C of line segment $\overline{CD}$ and $\mathbf{V_D}$ be a vector from C to D of line segment $\overline{CD}$. A position vector P to any point along line segment $\overline{AB}$ may then be described as $\mathbf{P}(u) = \mathbf{A} + u\mathbf{B}$, where $u \in [0, 1]$. Similarly, a position vector Q to any point along second line $\overline{CD}$ may be described as $\mathbf{Q}(w) = \mathbf{C} + w\mathbf{D}$, where $w \in [0, 1]$. Let R be the position vector of the point of intersection. If an intersection exists then

$$\mathbf{R} = \mathbf{P}(u) = \mathbf{Q}(w) \tag{21}$$

or

$$\mathbf{A} + u\mathbf{B} = \mathbf{C} + w\mathbf{D} \tag{22}$$

In two dimensions, this leads to two linear equations (onc for each co-ordinate direction) and two unknowns, u and w:

$$A_x + uB_x = C_x + wD_x \tag{23}$$

$$A_y + uB_y = C_y + wD_y \tag{24}$$

In the solution of these equations, if $0 \leqslant u \leqslant 1$, then the lines intersect[8] and the relative location of the intersection is indicated by the values of u and w. For instance, if $u = 0{\cdot}25$, then the

[8] If $0 \leqslant u \leqslant 1$ then the relation $0 \leqslant w \leqslant 1$ must also hold

intersection occurs at the quarter point along line $\overline{AB}$ closest to point A. A similar relation holds for the value of w and line $\overline{CD}$. This type of information is essential in properly correcting the overlap, as explained below.

Once an intersection has been detected, it is corrected by connecting the overlapping portions in the best possible manner. For this case of a single boundary intersecting itself, the connection will result in two new paving boundaries being formed. In many cases the best connection will be between the two lines that intersect. However, it is often more advantageous to combine lines adjacent to the intersection lines on either side and in either direction. The following considerations are followed to insure that the connection formed will be valid and will avoid the formation of irregular nodes where possible:

- *Evenness constraint.* The two new paving boundaries must each contain an even number of elements, as explained in Section 2.1. If connecting the intersecting lines will produce paving boundaries with an odd number of nodes, the connection location is adjusted based on the relative values of u and w.
- *Forward/backward search.* In some cases, connecting the intersecting lines would produce a more irregular mesh than connecting the two lines either forward or backward one node of the actual intersection. Thus connecting lines one step in either direction are compared based on nearness and how close the lines are to being parallel.
- *Size differences.* If the length of the respective intersecting lines differ significantly, a wedge is inserted similar to the process described in Section 2.8 to enable an easier transition.
- *Explore all intersections.* There will always be an even number of intersections of a closed paving boundary upon itself. By connecting one intersection and then seaming, as explained below, at least one more intersection location is eliminated. Thus, all intersections are found and compared and the *best* one (again based on the nearness and parallelness of the two intersecting lines) connected first.

Once the most appropriate connection has been found and the lines combined, a seam is attempted for both of the newly formed boundaries. Since any boundary portions which remain overlapped will form negative internal angles at the point of connection, a seam will pull all of these back into the existing mesh. Since multiple intersections may occur with the insertion of any row, each of the newly formed paving boundaries are themselves checked for remaining intersections before proceeding.

Figure 29 illustrates the connection of an intersecting boundary. In Figure 29(a), a new row has been added which overlaps portions of the existing paving boundary. In this case there are two intersection locations found, as highlighted by the dashed lines in Figure 29(a). The bottom intersection pair could be connected to form two loops with an even number of boundary nodes. However, the top intersection pair would form two loops with odd numbers of boundary nodes, and so steps in either direction are considered. The match highlighted in Figure 29(b) is the best connection for the top intersection. This connection is also preferable to the bottom connection based on nearness to being parallel and is thus connected as shown in Figure 29(c). In this example the seam operation following the connection closes both of the new paving boundaries formed and the completed mesh is shown in Figure 29(d).

Multiple boundary intersections. When a paving region contains interior paving boundaries (holes), they must all be eventually connected before closure is possible. Intersections are found similar to the intersection checking performed for a single boundary intersecting itself. However, since each boundary must contain an even number of intervals, it is not possible to generate an odd boundary loop from the connection of any two paving boundaries. Thus, connection is based

only on the goodness of the connection. This type of intersection may occur between the external and an internal paving boundary or between any two internal paving boundaries.

Figure 30 illustrates the intersection and connection of multiple boundaries. As shown in Figure 30(a), the exterior paving boundary intersects three of the interior paving boundaries (circular holes). The intersections with these three boundaries are evaluated, the most appropriate

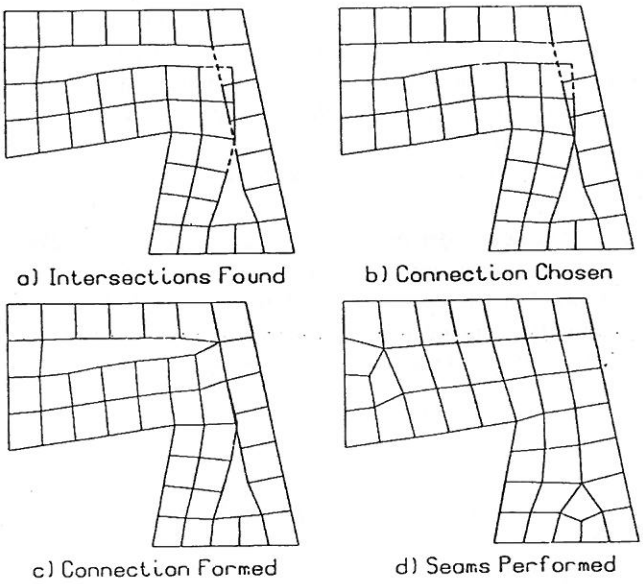

Figure 29. Example single boundary intersection formation

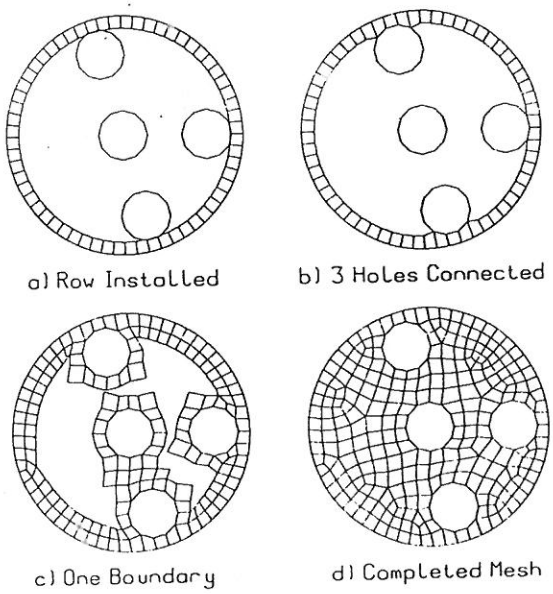

Figure 30. Example multiple boundary intersection formation

connection chosen, connected and seamed as shown in Figure 30(b). At this point there are only two paving boundaries left, the exterior paving boundary and the interior boundary around the centre hole. These two boundaries are then paved until they overlap and are connected as shown in Figure 30(c), forming a single exterior boundary. The completed mesh is shown in Figure 30(d).

2.11. *Cleanup of completed mesh*

After the region has been completely paved, local adjustments are attempted to improve the overall quality of the finished mesh. These adjustments are essentially an insertion or deletion of an element to improve the local aspect ratios, reduce the number of irregular nodes, or eliminate elements with bad interior angles. Factors controlling the adjustments include the number of element edges attached to a node, the interior angle of all elements attached to the node and the anticipated probability of improving the mesh locally.

Figure 31 shows an example of an element deletion improving the mesh. In Figure 31(a), the element formed by nodes N_i, N_j, N_k and N_l has been targeted for deletion. Of the nodes which form this element, only N_i is a regular node. Nodes N_j and N_l are attached to three elements and node N_k is attached to five elements. The deletion is performed by collapsing the element as shown by the arrows in Figure 31(a). Nodes N_j and N_l and the attached element edges are combined during the deletion. As a result of the deletion one regular node, N_i, is transformed into an irregular node, as shown in Figure 31(b). However, two irregular nodes, N_j and N_l, are

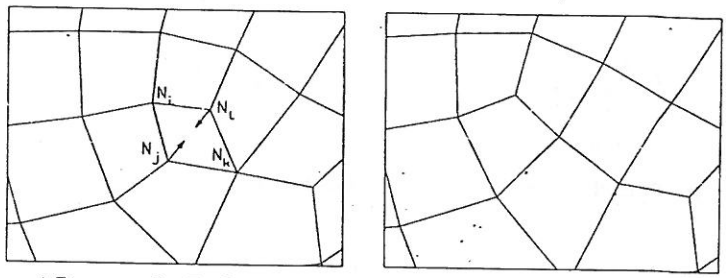

Figure 31. Example of an element deletion adjustment

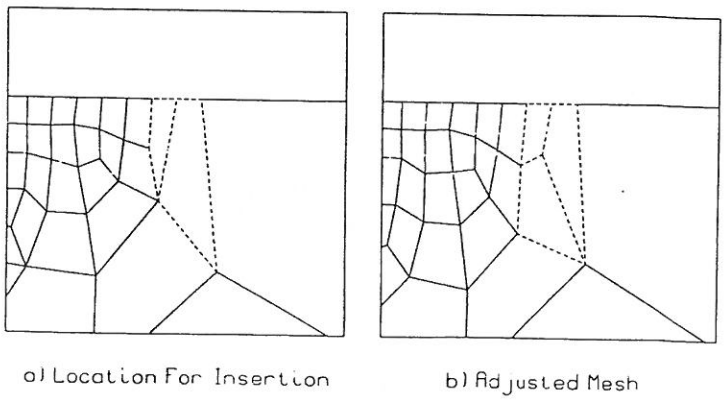

Figure 32. Example of an element insertion adjustment

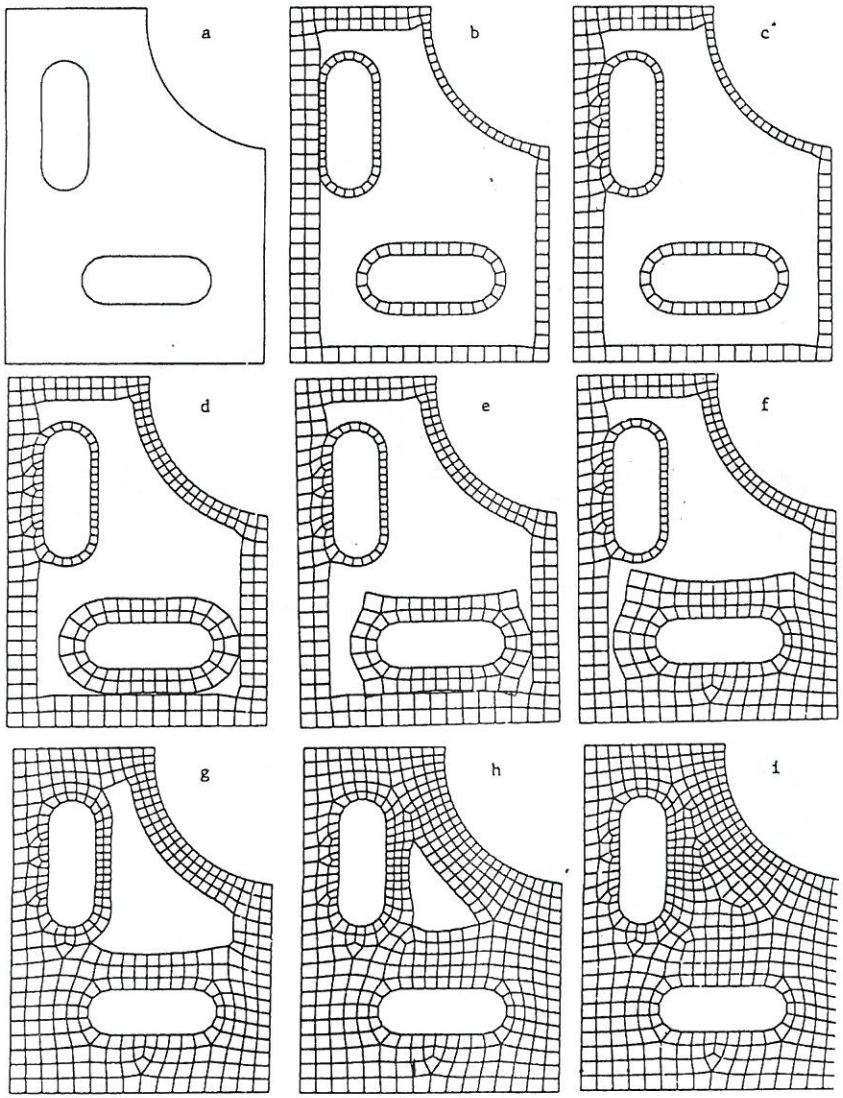

Figure 34. Paving of Example B

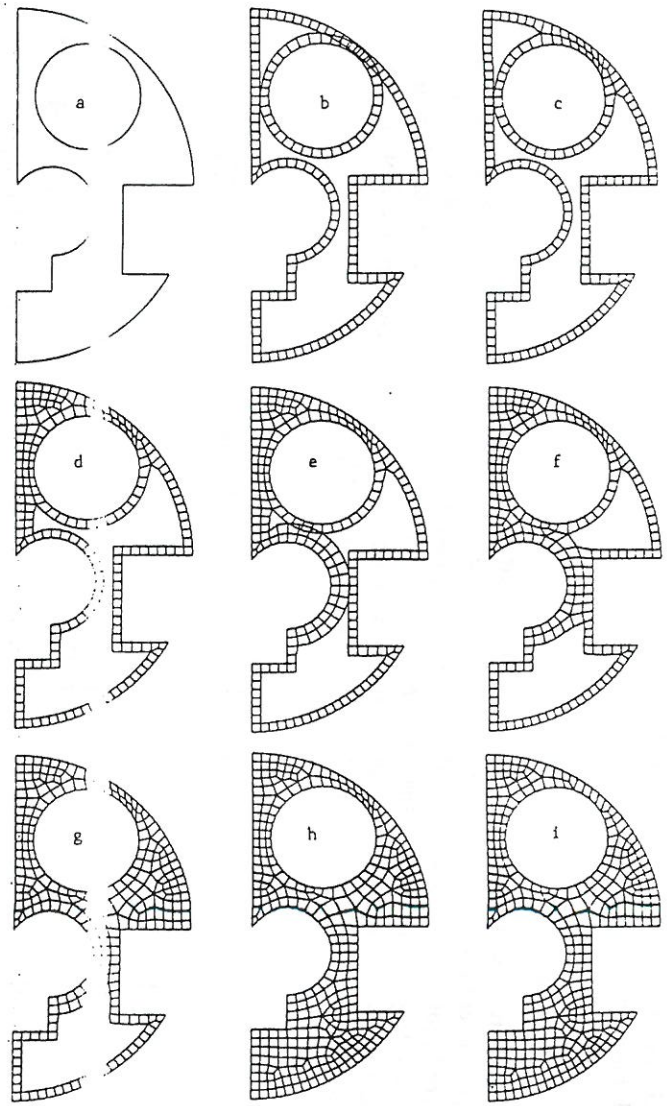

Figure 35. Paving of Example C

cleanup technique is able to eliminate most of the distorted elements, as shown in the final mesh in Figure 35(i).

3.4. *Example D*

Figure 36 depicts the paving process for Example D. Example D consists of a simple exterior permanent boundary with no interior boundaries. The exterior boundary does have an extreme gradation in the density of nodes along the boundary. As can be seen in Figure 36, nodes on the boundary of the notch at the upper right of the geometry are on the order of twenty times more dense than the nodes along the left and bottom boundary segments. The top and right boundary segments have an increasing density of nodes as they approach the notch portion.

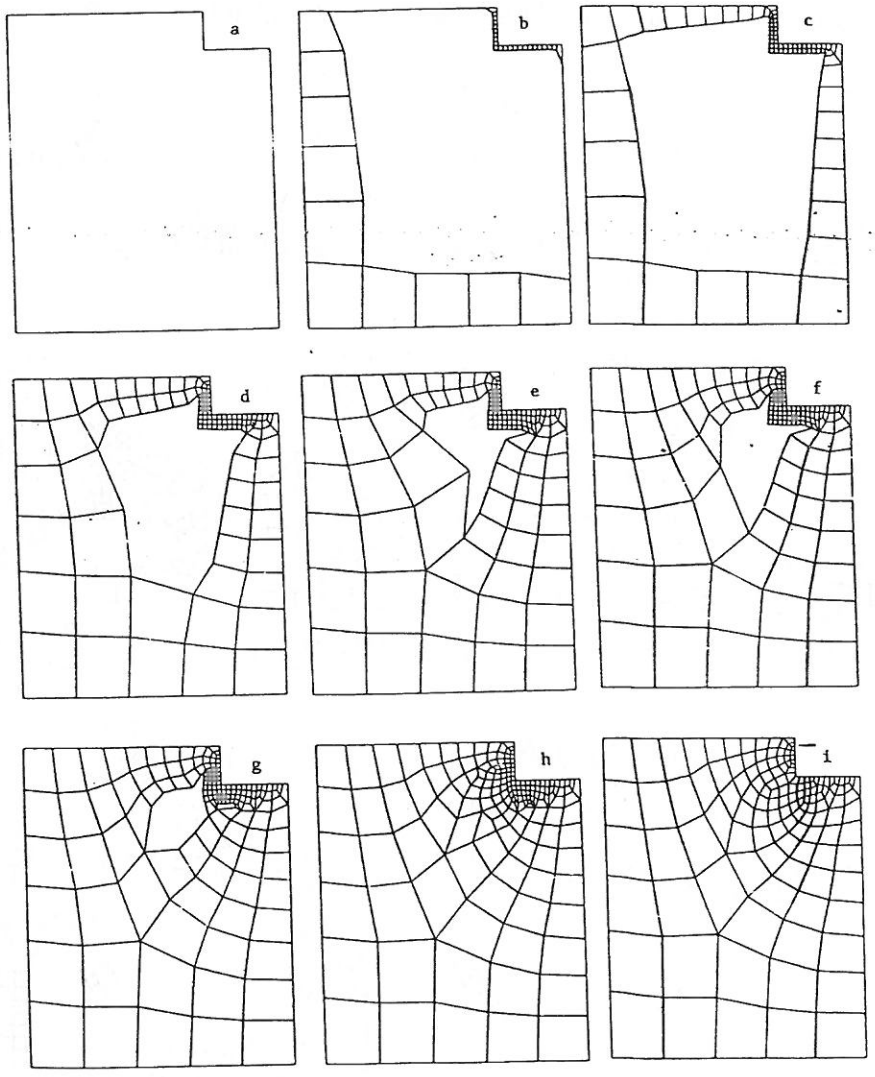

Figure 36. Paving of Example D

This example shows the versatility of the paving algorithm to handle large mesh gradation requirements. Paving rows retain the size dictated by the boundary segments from which they originate, as shown in Figure 36(d), (e) and (f). As disproportionate rows interact with each other adjustments are made to insure validity of the mesh. The mesh after final closure is shown in Figure 36(h). Cleanup dramatically improves the quality of the final mesh, as shown in Figure 36(i). As shown in this example, transitions tend to be *tight* or close to the areas of smaller mesh density. This is because a large element row, by definition, moves farther from the permanent boundary than a small element row. Thus when intersections occur between small and large rows, the intersection is physically much closer to the permanent boundary of the small rows.

4. PERFORMANCE EVALUATION

As mentioned earlier, the goal of this research is to generate an all-quadrilateral mesh in an arbitrary geometry. The criteria used to evaluate the performance of paving include (i) the quality of the generated mesh, (ii) the robustness of the algorithm, and (iii) the speed of generation. The example problems shown in Section 3 present a qualitative view of the paving algorithm. In this section we attempt to quantify these results.

4.1. Mesh quality

The qualitative objectives listed in Section 1 have been met by this technique. As can be seen in the examples, the meshes generated are boundary sensitive, orientation insensitive and minimize the placement of irregular nodes.

However, this type of qualitative assessment of the technique is of limited use in the very quantitative field of finite element analysis. In an attempt to quantify the quality of a mesh produced with the paving technique, an element distortion metric, proposed by Oddy *et al.*[17] for isoparametric elements, has been used. This distortion metric measures the degree of deviation of an element in the physical domain (x, y, z) from its assumed shape in the computational domain (ξ, η, ζ). It combines effects of shearing and stretching, is not affected by rigid body motions and is independent of element size. In calculating this metric we assumed a 4 node linear-order isoparametric element with single point integration.

As stated in Reference 17, a distortion metric can be used only as a relative indication of error since factors other than the mesh (e.g. type of analysis, size of elements and loading) affect analysis accuracy. However, elements with distortion metric values greater than 5 tend to be problematic. As an aid in understanding this distortion metric, a surface plot of the metric values for various parallelograms is shown in Figure 37. In this figure the aspect ratio is the ratio of lengths of the non-parallel sides of the parallelogram and θ is the acute angle between them.

Let D be the value of the distortion metric for an element. Average values of D, $\bar{D}$, for meshes generated using the paving technique range from 0·0 for *blocky* type geometries to 0·1 for relatively complex geometries without transitioning requirements. Transitioning, of course, requires the use of more distorted elements, but values of $\bar{D}$ still tend to be around 0·3. These results support the qualitative observation that paving tends to produce meshes with square elements.

As examples of the distortion metric evaluation, the meshes generated in Section 3 have been used. The element with the maximum D value has been indicated with the asterisk (*) symbol in Figures 38–41. Figure 38 shows the mesh generated for Example A. In this mesh the average element distortion metric, $\bar{D}$, is 0·02, with a maximum D value of 0·26. Figure 39 shows the mesh

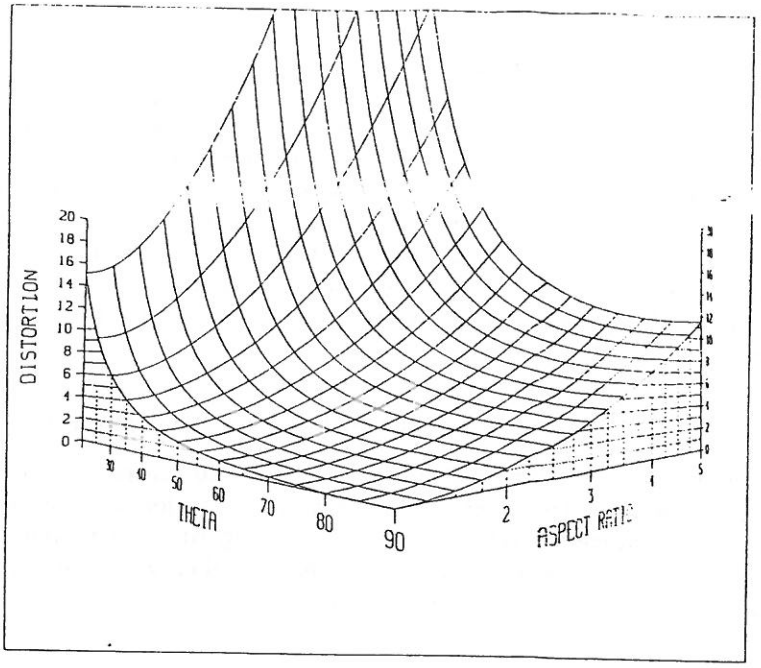

Figure 37. Surface plot of distortion metric for parallelograms

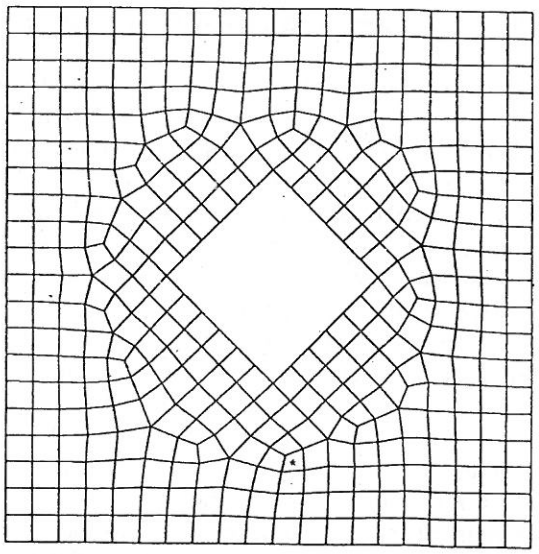

Figure 38. Example A: $\bar{D} = 0{\cdot}02$, $D_{max} = 0{\cdot}26$

for Example B. In this mesh $\bar{D}$ is 0·09, with a maximum value of D of 1·09. Figure 40 is the mesh for Example C. Here $\bar{D}$ is 0·11 and the maximum D value is 1·13. The final example, Example D, is shown in Figure 41. For this example $\bar{D}$ is slightly higher at 0·30 with the maximum D value of 1·96. This value is for the element in the upper left corner where the intervals on the exterior boundary force a 3:1 aspect ratio at the corner.

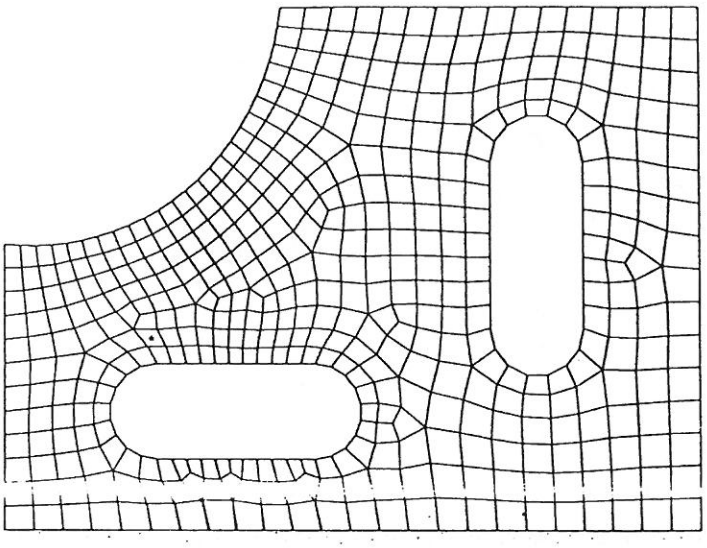

Figure 39. Example B: $\bar{D} = 0{\cdot}09$, $D_{max} = 1{\cdot}09$

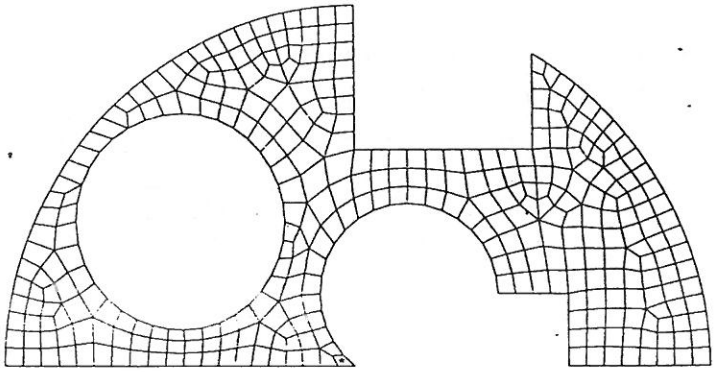

Figure 40. Example C: $\bar{D} = 0{\cdot}11$, $D_{max} = 1{\cdot}13$

4.2. Robustness

As mentioned earlier, paving has proved empirically to be robust for very complex geometries. Even drastic transitions have been handled well with paving. However, the technique breaks down when *impossible* transitions (e.g. 30:1) are required over a very restricted area. However, even this limitation can be removed if adjustment of the permanent boundary (i.e. insertion/deletion of nodes) is allowed.

4.3. Speed

The speed of the paving approach is difficult to tie directly to the size or number of nodes in a generated mesh. This is because paving's effectiveness is tied to the shape of the boundary as well as the number of nodes in the boundary. For instance, the paving boundary for a geometry which

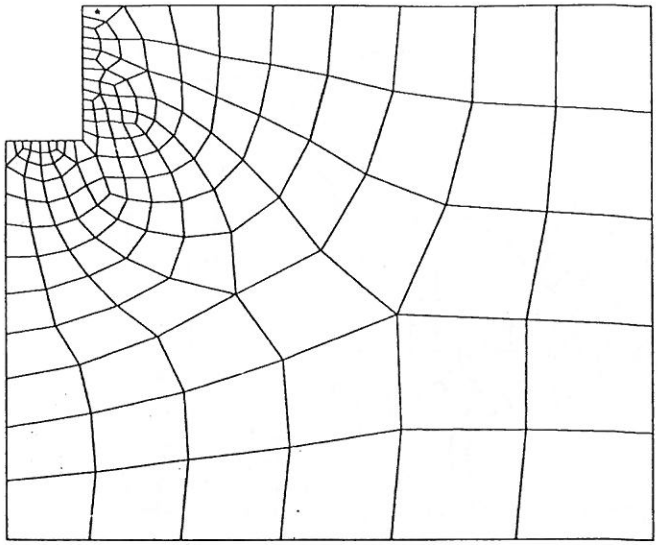

Figure 41. Example D: $\bar{D} = 0{\cdot}30$, $D_{max} = 1{\cdot}96$

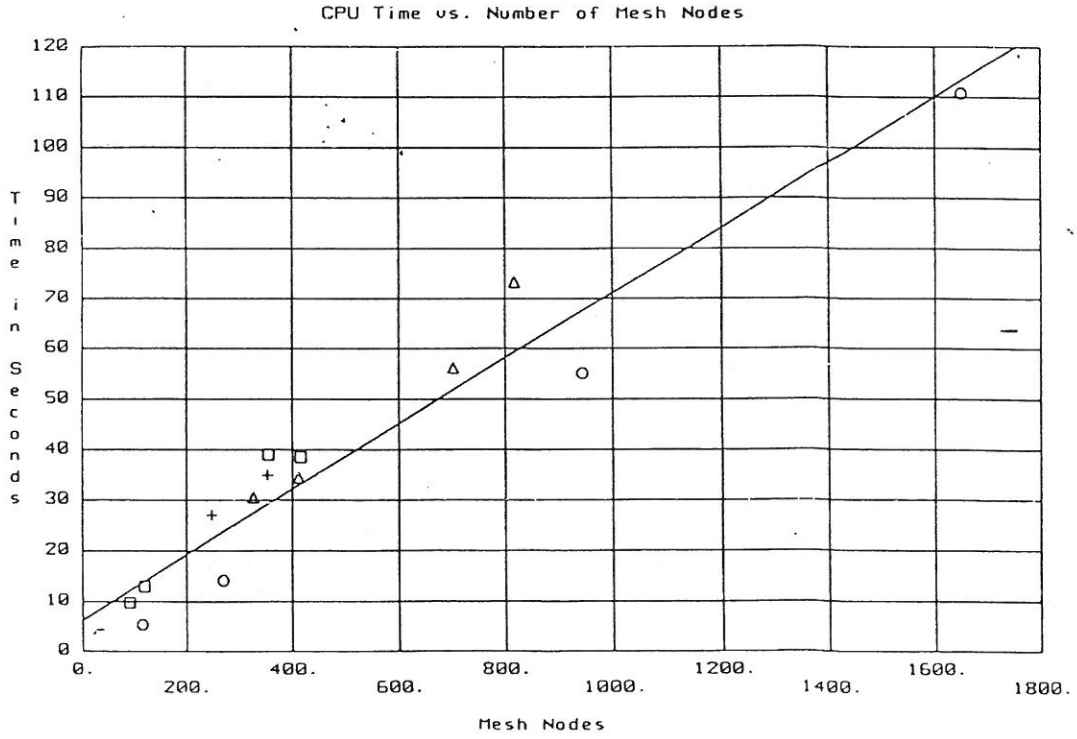

Figure 42. Required CPU for paving vs. number of nodes in the mesh

is relatively long and thin tends to overlap itself quickly, generating many smaller paving boundaries which can be processed rapidly. On the other hand, large open boundaries are processed somewhat more slowly. We have measured performance empirically and have found that the required CPU time is roughly linear with the number of nodes in the final mesh, as shown in Figure 42. The geometric dependencies account for the scatter in the data. For this study, a relatively unloaded VAX 11/780 computer was used to process various geometries with increasingly finer mesh requirements. In practice, most meshes are being generated quickly enough to allow interactive processing.

5. CONCLUSIONS

The technique of paving has satisfied the objectives of automatically generating a high quality mesh for an arbitrary geometry. The technique has been empirically shown to be robust and fast for complex geometries and extreme mesh gradations. The paving technique tends to place well formed elements along the boundary with irregular nodes in the interior of the geometry. The mesh quality measured using a distortion metric has been shown to be extremely good. This technique is thus suitable for general automated all-quadrilateral mesh generation.

This technique may prove very powerful when combined with an error estimator for iterative adaptive analysis. The use of a function defining the required element size over the entire geometry rather than only at the boundaries should improve the technique. The use of an approach similar to paving for 3-D all-hexahedral mesh generation is also being investigated.

REFERENCES

1. P. L. Baehmann, S. L. Wittchen, M. S. Shepard, K. R. Grice and M. A. Yerry, 'Robust geometrically based, automatic two-dimensional mesh generation', *Int. j. numer. methods eng.*, **24**, 1043–1078 (1987).
2. A. Kela, R. Perruchio and H. B. Voelker, 'Toward automatic finite element analysis', *Comp. Mech. Eng.*, **5**, 1 (1986).
3. M. S. Shepard and M. A. Yerry, 'Approaching the automatic generation of finite element meshes', *Comp. Mech. Eng.*, **1**, 49–56 (1983).
4. B. Joe and R. B. Simpson, 'Triangular meshes for regions of complicated shape', *Int. j. numer. methods eng.*, **23**, 751–778 (1986).
5. B. Wordenweber, 'Finite-element analysis for the naive user', in M. S. Pickett and J. W. Boyse (eds.), *Solid Modeling by Computers*, Plenum Press, New York, 1984, pp. 81–102.
6. A. J. G. Schoofs, L. H. T. M. Van Beukering and M. L. C. Sluiter, 'A general purpose two-dimensional mesh generator', *Advan. Eng. Software*, 1–3, 131–136 (1979).
7. W. A. Cook and W. R. Oakes, 'Mapping methods for generating three-dimensional meshes', *Comp. Mech. Eng.*, 1, 67–72 (1983).
8. W. J. Gordon and C. A. Hall, 'Construction of curvilinear co-ordinate systems and applications to mesh generation', *Int. j. numer. methods eng.*, 7, 461–477 (1973).
9. K. Ho-Le, 'Finite element mesh generation methods: A review and classification', *Comp. Aided Des.*, **20**, 27–38 (1988).
10. M. S. Shepard, 'Finite-element mesh generation for use with solid modeling and adaptive analysis', in M. S. Pickett and J. W. Boyse (eds.), *Solid Modeling by Computers*, Plenum Press, New York, 1984, pp. 53–80.
11. T. D. Blacker, M. B. Stephenson, J. L. Mitchiner, L. R. Phillips and Y. T. Lin, 'Automated quadrilateral mesh generation: A knowledge system approach', *ASME Paper No. 88-WA/CIE-4*, 1988.
12. T. D. Blacker, *FASTQ Users Manual Version 1.2*, SAND88-1326, Sandia National Laboratories, 1988.
13. M. B. Stephenson and T. D. Blacker, 'Using conjoint meshing primitives to generate quadrilateral and hexahedral elements in irregular regions', *Proc. ASME Computations in Engineering Conference*, 1989.
14. L. R. Herrmann, 'Laplacian-isoparametric grid generation scheme', *J. Eng. Mech. Div. ASCE*, **102**, EM5, 749–756 (1976).
15. R. E. Jones, *QMESH: A self-organizing mesh generation program*, SLA-73-1088, Sandia National Laboratories, 1979.
16. M. E. Mortenson, *Geometric Modelling*, Wiley, New York, 1985.
17. A. Oddy, J. Goldak, M. McDill and M. Bibby, 'A distortion metric for isoparametric finite elements', *Trans. CSME*, No. 38-CSME-32, Accession No. 2161 (1988).