Adaptive Remeshing Method for Finite-Element Thermal Analysis

Earl A. Thornton* and Gururaja R. Vemaganti†
Old Dominion University, Norfolk, Virginia

A finite-element remeshing approach that makes use of quadrilateral and triangular elements is described. The approach uses the solution on a previous mesh to create a new mesh. Meshes are completely unstructured with highly refined elements in regions of steep gradients and larger elements where gradients are smaller. Studies of convergence rates for heat conduction problems with exact solutions show that for problems with highly localized solution variations, the remeshing approach gives smaller solution errors with fewer unknowns than refinement of uniform, structured meshes.

Nomenclature

a = dimension of solution domain
A = element area, Eq. (B3)
A_i = area of ith element
A,B = constants in exact solutions, Eqs. (10–11)
e = solution error at a point x,y
$\|e\|$ = global solution error, Eq. (7)
h = element dimension
h_1,h_2 = dimensions for element creation
k = thermal conductivity
L_i = length of ith boundary segment
M = number of elements, Eq. (6)
N = number of boundary points, Eq. (5)
$[N]$ = finite-element interpolation functions
Q = heat generation rate per unit volume
$\boldsymbol{R}$ = position vector of a node
T = exact temperature
T_h = finite-element temperature
x,y = coordinate directions
α = orientation vector for element creation
λ_1,λ_2 = eigenvalues, principal value of derivatives
Φ = general dependent variable
ψ = function in exact solution, Eqs. (10–11)

Superscript

e = element quantity

Introduction

DEFORMATION and stresses induced by aerodynamic heating in high-speed flight vehicles are important concerns in vehicle structural design. Aerodynamic heating may have a significant effect on the performance of the structure, and effective techniques for predicting the heating and the thermal-structural responses are required.

Research is underway at the NASA Langley Research Center and other research centers to improve the capabilities and efficiency of the finite-element method for high-speed compressible flows and to develop efficient integration of finite-element fluid, thermal, and structural analyses. A focus of this research is to develop adaptive finite-element methodology for the accurate prediction of aerothermal loads and the thermal-structural response of complex three-dimensional bodies.

Adaptive finite-element methods have been the focus of research efforts for the last few years. A survey of the literature on adaptive finite elements was compiled by Oden and Demkowicz,[1] and adaptive grid generation in the finite-difference computational fluid dynamics (CFD) context is described by Eiseman.[2] An objective of adaptive refinement is to obtain the best resolution of physical phenomena for a given computational effort. One approach used in the finite-element community is first to obtain a solution on an original crude mesh. Based on this solution and the concept of error indicators, a new mesh is created based upon the original mesh. Two basic approaches being followed in this approach: 1) adaptive refinement/derefinement and 2) adaptive remeshing. There are three basic methods for mesh refinement/derefinement: 1) the h method; 2) the p method, and 3) the r method. In the h method the elements of the initial mesh are refined into smaller elements or derefined into larger elements. In the p method the order of the polynomial used for the element interpolation function is increased (or decreased) while keeping the geometry of the element constant. The r method keeps the number of elements and their connectivity the same but moves the nodes. There are also methods of mesh refinement that deal with combinations of the above three methods.

Adaptive refinement/derefinement schemes are being applied to a wide range of finite-element applications in fluids, structures, and heat transfer.[3–7] Each of the methods have limitations. In the neighborhood of strictly one-dimensional features like shocks and thin boundary layers, mesh enrichment is not very efficient, and, from one refinement to another the number of elements increases significantly. Another disadvantage of mesh enrichment methods is that the original location of the nodal points does not alter through successive refinements. Though new points are added, and old ones are deleted at each stage of refinement, the initial orientation of the elements does not change. Though the r method does not increase the number of elements, it may give rise to highly distorted elements. Implementation of the p method is more complicated than the h method, because extensive modification of analysis programs is required. However, convergence rates for p methods are higher than for h refinements. An area of current investigation is a combination of h and p methods.

These considerations led Peraire et al.[8] to develop an approach for adaptive remeshing. The approach generates an

Received July 5, 1988; revision received Jan. 27, 1989.

*Professor; currently Visiting Scholar, Texas Institute of Computational Mechanics, The University of Texas at Austin. Associate Fellow AIAA.

†Graduate Research Assistant. Student Member AIAA.

entirely new mesh based on the information provided by the solution on an earlier mesh.

In the remeshing approach, nodal spacing and the orientation of the elements are represented by parameters that are computed at the nodal points of an earlier mesh. In many instances, the remeshing approach improves the quality of the solution without significantly increasing the total number of unknowns. In the Peraire remeshing method only triangular elements are used, and the remeshing can provide stretching of the elements locally. However, a global directionality cannot be achieved because of the geometry of the elements. For the prediction of aerodynamic heating, it was found that quadrilateral elements are better suited for boundary layers, and remeshing with triangles in this region does not give the best results. Additionally there is the motivation that remeshing based on quadrilaterals when extended to three dimensions will offer important advantages. For example, hexahedral elements take considerably less computer storage compared to their tetrahedral counterparts in three-dimensional applications.

The purpose of this paper is to present a new remeshing approach that uses predominantly quadrilateral elements. The approach uses quadrilateral elements where possible, and triangles are introduced as needed. The remeshing approach at each stage uses a solution based on a previous mesh to generate a new mesh. In this paper the remeshing approach is applied to heat conduction problems, although, of course, remeshing may be applied to other applications as well. The basic concepts of adaptive remeshing are discussed first. Then, the remeshing approach based on quadrilateral and triangular elements is presented. Finally heat conduction examples are analyzed using the new remeshing approach. Comparison of solution errors based on exact solutions for both uniform h refinement and adaptive remeshing are made to assess the effectiveness of the approach.

Adaptive Remeshing

Concept of Remeshing

The main idea of remeshing lies in generating a completely new mesh based on solution information available on a previous mesh. This information takes the form of mesh generation parameters computed on the previous mesh at the nodal points. The method becomes adaptive when these parameters are computed from a numerical solution on the previous mesh. The mesh generation process produces proper sizing of the elements where refinement is required and provides smooth transition from a high-resolution region to a low-resolution region. This smooth transition in the mesh is desirable for certain algorithms.[9] Proper clustering of the elements near regions of high gradients is achieved in the method without refining existing elements. With remeshing, the nodal locations change from mesh to mesh unlike the h refinement/derefinement methods described previously. Numerical examples will show with remeshing the quality of the solution may sometimes be improved significantly without increasing the total number of unknowns. In some instances, solution quality increases with a *reduction* of the number of unknowns.

The crude earlier mesh in this method is referred to as the background mesh. Initially a hand-made mesh with a few elements can be used as a background mesh. Since a solution may not be available on this mesh, mesh generation parameters may be specified at the nodal points of the mesh. Intuitive solution knowledge may be useful in this specification, but a regular and uniform distribution of the parameters is sufficient for many cases. Once this information is available, a new mesh is created within the domain that is to be discretized. The domain boundaries are predefined by the user. The mesh generation parameters needed for the new mesh at various locations within the domain are linearly interpolated from the background mesh. A search algorithm (Appendix A) based on the nodal coordinates of an element is used for this purpose. The initial background mesh need not coincide with the domain of discretization. Then a finite-element solution is obtained on the new mesh, and the mesh generation parameters are computed based on the numerical solution. Continuing, the current mesh becomes a background mesh for the next mesh to be generated. The process of generating sequential meshes continues until the desired convergence for the solution is achieved.

Mesh Generation Parameters

The mesh generation parameters used for the construction of a new mesh are shown in Fig. 1. They are as follows: 1) two components of a vector α along which an element is to be stretched, 2) a spacing h_1 normal to this vector, and 3) a spacing h_2 tangential to the vector. Thus a new element has a dimension h_1 in the direction of the vector and a dimension h_2 in a direction normal to the vector.

The key step between a solution on a mesh and its adaptive remesh is the computation of the mesh generation parameters. The mesh generation parameters are computed from an error indicator based on a dependent variable. In most cases the actual solution error cannot be calculated exactly, but relative errors can be estimated by indicators based on the finite element solution. Studies of mesh optimization[10] have shown that an optimum mesh is achieved when the local error is distributed equally throughout the mesh. Equal distribution of the estimated error is the basis for mesh refinement/derefinement schemes as well as the adaptive remeshing scheme used herein.

As an illustration, in one-dimension the solution error for an element is estimated by

$$h^2 \left| \frac{\partial^2 \Phi}{\partial x^2} \right|$$

where h is the length of the element, and Φ is the dependent variable. An optimum mesh is obtained when this error is equally distributed by requiring

$$h^2 \left| \frac{\partial^2 \Phi}{\partial x^2} \right| = \text{const} \tag{1}$$

Notice that Φ is the dependent variable in the finite-element solution, and that second derivatives of the solution must be computed. In the refinement/derefinement schemes, the adaptive strategy is based on equal distribution of solution error

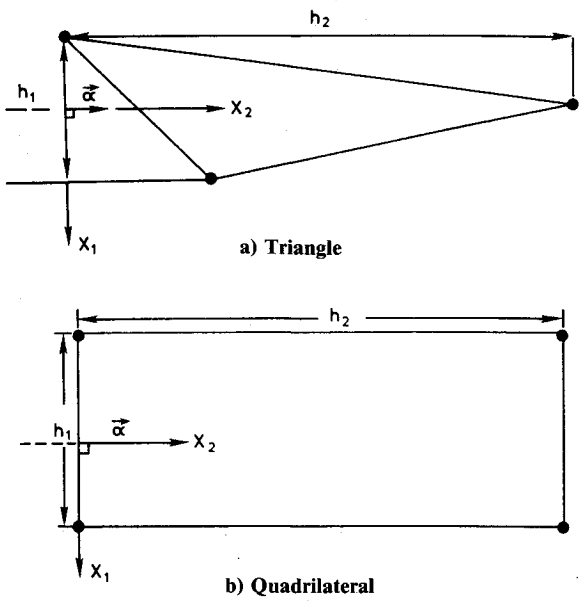


Fig. 1 Mesh generation parameters for adaptive remeshing.

for elements. In remeshing, the adaptive strategy is based on an interpolation of the solution error at nodes.

In application of remeshing to the two-dimensional case, the second derivatives at a point are given by the matrix

$$\begin{bmatrix} \frac{\partial^2\Phi}{\partial x^2} & \frac{\partial^2\Phi}{\partial x\partial y} \\ \frac{\partial^2\Phi}{\partial y\partial x} & \frac{\partial^2\Phi}{\partial y^2} \end{bmatrix} \tag{2}$$

The method used for computing the second derivatives needed in Eq. (2) is described in Appendix B. For remeshing, the principal directions are computed along which the cross derivatives vanish. Principal eigenvalues λ_1 and λ_2, and principal directions X_1 and X_2 are computed. Equal distribution of solution error at nodes is achieved by requiring

$$h_1^2\ |\lambda_1| = h_2^2\ |\lambda_2| \tag{3}$$

where h_1 refers to an element dimension in the X_1 direction, and h_2 refers to the element dimension in the X_2 direction (see Fig. 1). Thus,

$$\frac{h_2}{h_1} = \sqrt{\left|\frac{\lambda_1}{\lambda_2}\right|} \tag{4}$$

The eigenvalue problem is solved at each nodal point of the background mesh including the boundaries. Using the maximum eigenvalue and a user-specified minimum h, the constant in Eq. (3) is determined. Then at each point, h_1 and h_2 can be computed. These dimensions determine the size of an element to be created. The shape of the element is constrained by limits on the internal angles so that a distorted element is not created. Usually these limits are between 45 and 135 deg for a quad element and 30 to 150 deg for a triangle.[11]

Remeshing Example

To illustrate the basic steps in remeshing, an example is presented in Fig. 2. The figure illustrates a new mesh being created from a solution obtained on a previous mesh. In Fig. 2a, the background mesh is shown. Using this mesh, a finite-element solution has been obtained. Figures 2b–2d show the evolution of the new mesh. Figure 2b shows the boundary points that are created first; Figs. 2c and 2d show the mesh at two stages of development, and Fig. 2e shows the final mesh. Notice the remeshing proceeds inward from the boundary, and that quads as well as triangles are generated as the remeshing proceeds.

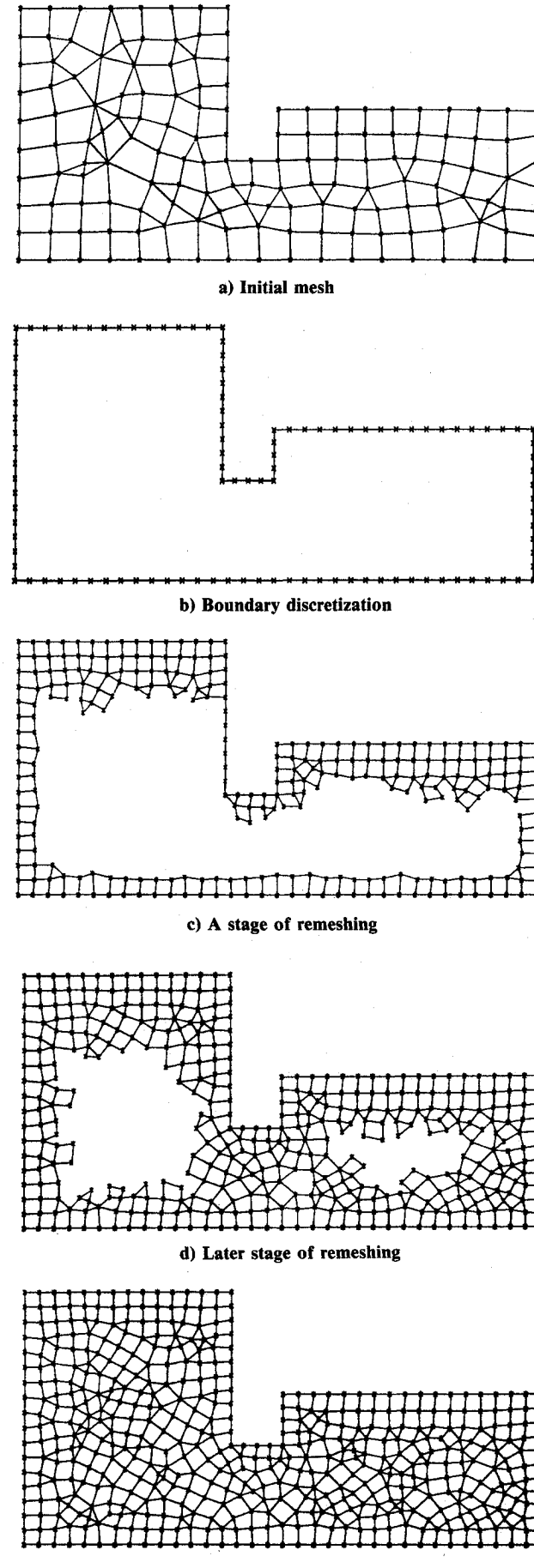
Remeshing with Quads and Triangles

Boundary Discretization

The mesh generation process starts with boundary discretization. Boundary segments joining fixed-boundary nodes are ordered in an anticlockwise manner for an external boundary and are ordered in a clockwise manner for an internal boundary defining a hole within domain. This way the domain to be discretized always exists on the left-hand side of the boundary.

Additional boundary nodes are included to satisfy the spacing requirements compatible with the background mesh. Each boundary segment is descretized in order until the entire boundary is covered. When a boundary segment is to be discretized in the new mesh, the number of intermediate points to be included is determined first. A number of sample points on the same segment in the background mesh are taken where the dimensions h_1 and h_2 are known. Then the number of intermediate points N to be included is determined from

$$N = \sum \frac{1}{2}\left[\frac{1}{(h_1)_i} + \frac{1}{(h_1)_{i+1}}\right] L_i \tag{5}$$



a) Initial mesh

b) Boundary discretization

c) A stage of remeshing

d) Later stage of remeshing

e) Final remesh

Fig. 2 An adaptive remeshing example.

where i denotes a sample point, and L_i is the distance between adjacent sample points. Note that the right-hand side of Eq. (5) is adjusted to the closest integer to yield N. Then the intermediate spacings between these points are computed based on the local h_1 values. The spacings are scaled appropriately to account for adjusting the value N. Once the intermediate spacings are established, the coordinates of the new points are computed.

Front Concept

The mesh generation process is based on an advancing front technique similar to the method proposed by Lo.[12] The front consists of adjacent nodes joined by line segments. The initial front consists of the boundary segments that link adjacent boundary nodes. As the mesh construction goes on, the front changes its shape. Whenever new nodes are created they are included in the front accompanying the nodes that are linked to them. When a node cannot be used for further element construction, it is eliminated from the front and becomes inactive. Thus the front can be defined as a chain of line segments that surround the domain that remains to be discretized. The front changes its shape constantly during the construction process and vanishes when the mesh is complete.

The evolution of the front is illustrated by the example introduced in Fig. 2. In Fig. 2b, the boundary of the mesh makes up the initial front as the remeshing begins. Figure 2c shows the front as the mesh generation proceeds inward from the boundary. In Fig. 2d, there are multiple fronts.

Element Creation

Element creation proceeds from the smallest frontal segment giving priority to regions which require refinement. Three different options are used for creating elements. At the smallest segment of the front, these options are attempted in sequence until an element is created. The first two options create quadrilateral elements, and the last option creates triangular elements. Thus the final mesh consists of predominantly quadrilateral elements. Once the mesh is complete, a post-process combines two triangles at a time, wherever possible, to make as many well-shaped quads as can be created. The different options for creating elements are illustrated in Fig. 3 and are discussed below.

Option 1—This option attempts to make quadrilaterals on corners of the front (see Fig. 3). Corner points where frontal segments can be used as sides of a quad element are identified first. An advantage of this option is three frontal nodes are available for a quad, and only one node needs to be created. Based on the mesh generation parameters at the end nodes of the smallest frontal segment A-B, the midpoint parameters are computed by linearly interpolating endpoint values. The location of the fourth point D is determined from the α vector and the dimension h_2. Before this point is taken as a fourth point, other points that already exist nearby in the front are considered. Hence before a new point is created, existing points that can be used are checked to keep the number of nodes to a minimum. Finally, the front is updated deleting the inactive node B and including the new active node D. If the method cannot generate an element at this corner point it goes to the next corner point in an order of increasing frontal length. If all the corner points fail to generate an element the next option for the element creation is tried.

Option 2—This point attempts to make a quadrilateral on the smallest frontal segment by locating two points within the domain (see Fig. 3). Based on the mesh generation parameters at the end nodes of the segment A-B, the midpoint parameters are computed as in Option 1. The location of the points C and D are determined from the α vector and the dimension h_2. Before these points are created, points that already exist nearby are examined. Construction takes place by joining A,B,C, and D. Again the front is updated including the points C and D. If an element cannot be created with this option, the method attempts the last option.

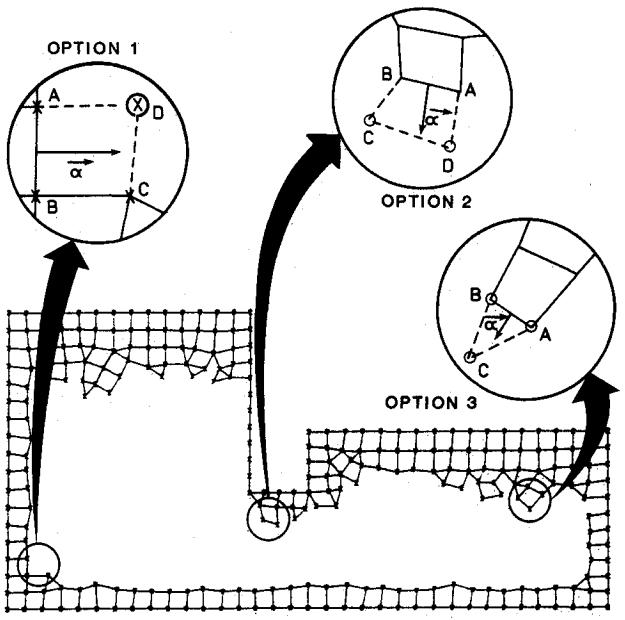


Fig. 3 Options considered in creating a new element.

Option 3—In the last option tried in the cycle, a triangle is created instead of a quadrilateral. Unlike in Option 2 where two points are created within the domain, only one point is created in this case. The principle of element creation in this option is shown in Fig. 3. Point C is created in such a way that a well-shaped triangle is obtained. Inclusion of this option gives the method flexibility to create an element on almost any arbitrary front. Once a triangular element is created the front is updated.

After an element is created by one of the options, the next element is attempted on the updated front starting with Option 1.

Mesh Smoothing

The mesh that is generated may have some distorted elements because of maximum angle tolerances and/or aspect ratios permitted. To minimize element distortions, the nodes are repositioned. This operation is referred to by some researchers as mesh smoothing. In the smoothing process every interior node is moved to the centroid of the polygon formed by the nodes that are connected to it. Boundary nodes are constrained to move along the boundary edge on which they lie.

Let C denote the node that is to be repositioned, and let M denote the number of elements that share C as a common node. Then the new position vector of node C is computed from

$$\left(\sum_{i=1}^{M} A_i\right)\boldsymbol{R}_c = \sum_{i=1}^{M} A_i \boldsymbol{R}_i \tag{6}$$

where A_i denotes the area of the ith connected element, and $\boldsymbol{R}_i$ is the position vector of the centroid of the ith element. The new position of all nodes are computed in this fashion. Then the process is repeated for several iterations until the nodes reach their final positions. Typically, ten iterations are sufficient to bring the nodes to their final positions.

More sophisticated versions[2] of this type of repositioning may be used to improve the quality of the solution. These procedures use Eq. (6) with a weighting factor that is determined by the solution on the smoothed mesh.

Applications

These applications are presented to illustrate the remeshing approach. Heat conduction examples with exact analytical

solutions are used to assess the convergence of the global error as the meshes are adapted successively to the numerical solutions. For heat conduction problems, let $T(x,y)$ be the exact solution, and $T_h(x,y)$ be the finite-element solution. The error in the approximation at any point is the function

$$e(x,y) = T(x,y) - T_h(x,y)$$

In the examples that follow, the global error in the approximate solution is measured in the L^2 norm defined by

$$\| e \| = \left[\frac{\int_A (T - T_h)^2 \, dx dy}{A} \right]^{1/2} \tag{7}$$

where A is the area of the solution domain. The integral is evaluated on an element basis and summed over all elements in the domain. Four-point Gauss integration is used to evaluate the integrals; a check with nine-point integration for a solution with steep gradients verified the accuracy of the four-point approach.

Example 1

The first example (Fig. 4) consists of the classical problem of heat conduction in a square where $T(x,y)$ is a solution of the steady equation,

$$k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + Q = 0 \tag{8a}$$

subject to the boundary conditions,

$$\frac{\partial T}{\partial x}(o,y) = 0; \qquad T(a,y) = 0$$

$$\frac{\partial T}{\partial y}(x,o) = 0; \qquad T(x,a) = 0 \tag{8b}$$

for constant thermal conductivity k and heat generation Q. The exact solution given by Carslaw and Jaeger[13] is

$$T(x,y) = \frac{Q(a^2 - x^2)}{2k} - \frac{16Qa^2}{k\pi^3} \sum_{n=0}^{\infty} \times \frac{(-1)^n \cos(2n+1)\frac{\pi x}{2a} \cosh(2n+1)\frac{\pi y}{2a}}{(2n+1)^3 \cosh(2n+1)\frac{\pi}{2}} \tag{9}$$

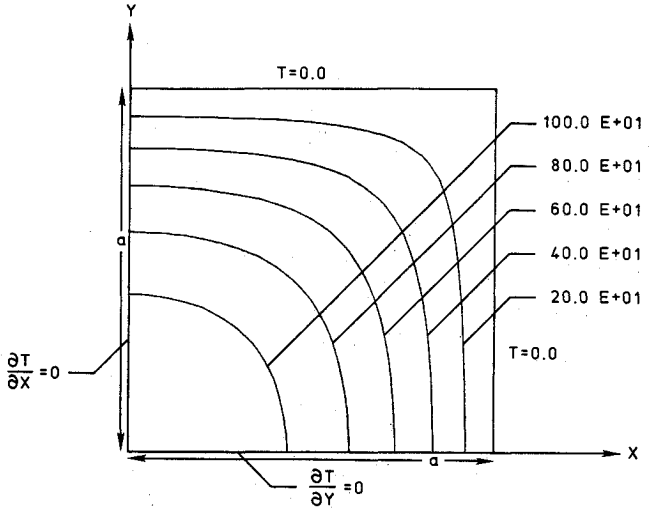
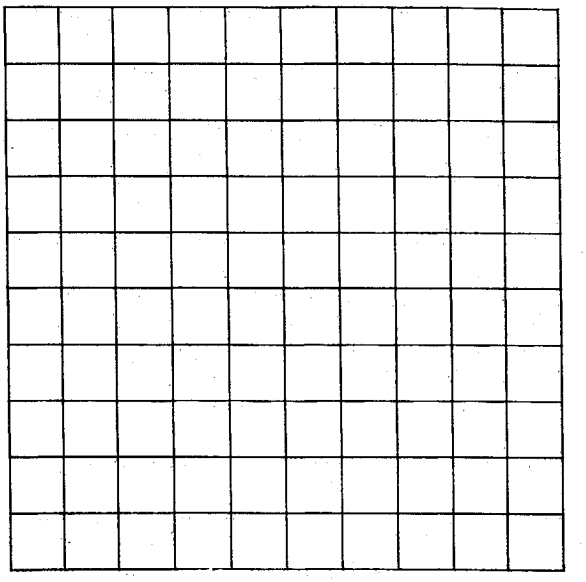
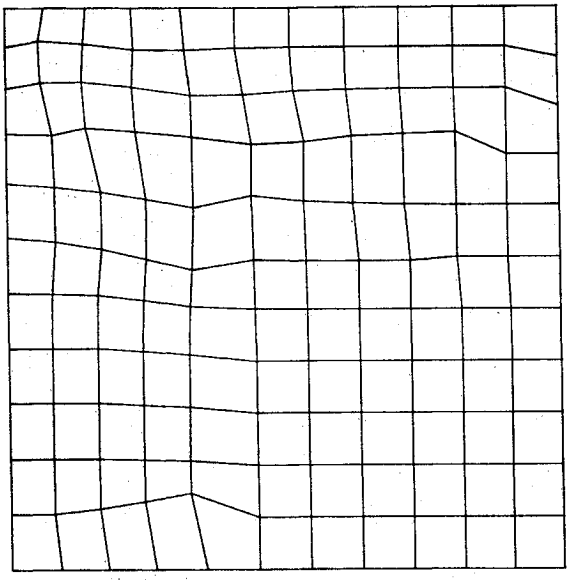


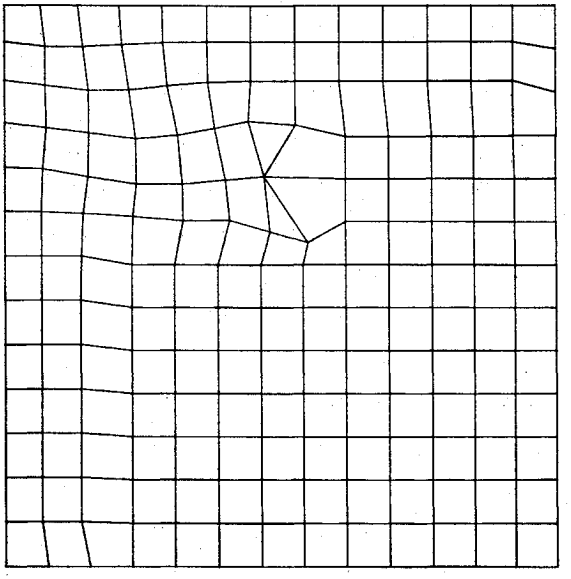
Fig. 4 Temperature contours for two-dimensional heat conduction with internal heat generation, Example 1.



a) Mesh 1



b) Mesh 2



c) Mesh 3

Fig. 5 Example 1: initial mesh and adaptive remeshes.

Table 1 Example 1: Comparative solution errors

Uniform h refinement			
Mesh	Unknowns	h	Error, $\|\|e\|\|$
1	36	4	4.3480
2	121	2	1.0870
3	441	1	0.2717
Remeshing			
Mesh	Unknowns	h_a^*	Error, $\|\|e\|\|$
1	121	2.000	1.0870
2	144	1.818	0.8816
3	196	1.539	0.6381

*h_a denotes an average element size defined as the square root of the area A divided by the number of elements in the remesh.

Finite-element solutions were obtained for the following: 1) three meshes with uniform h refinement of square elements, and 2) two adaptive remeshes where the remeshing started from the solution on uniform mesh 2. The meshes used in the remeshing approach are shown in Fig. 5, and solution errors are tabulated in Table 1.

The remeshes (meshes 2 and 3) shown in Fig. 5, do not differ greatly from those obtained in a uniform h refinement. The exception is the two triangles that appear in mesh 3, Fig. 5c. The triangles are generated spuriously by the program, possibly due to tolerance for element distortions. Some studies of mesh movement (not shown) indicated that a nearly optimum mesh for this problem consists of square elements slightly distorted into quads but with very smooth element interface curves. The remeshes shown in Fig. 5 appear to have some features of the optimum meshes, but fall short of producing smooth optimum meshes. Nevertheless, when solution errors are considered, the remeshes of Fig. 5 give acceptable results.

Considering Table 1, note that uniform h refinement reduces the global solution error according to h^2 as predicted by finite-element theory. Using the average element size h_a, the solutions based on the remeshes follow the same trend; that is, the solution error is $0(h_a^2)$.

Thus in this problem, remeshing is improving the solution quality at the same rate as a uniform h refinement. The reason that remeshing shows no advantage over uniform h refinement is that the problem solution is smooth with no steep gradients.

Example 2

This model problem[4] assumes a solution that satisfies Eq. (8a) and homogenous Direchlet boundary conditions. The solution takes the form

$$T(x,y) = \psi(x,x_o,c_x)\ \psi(y,y_o,c_y) \tag{10a}$$

$$\psi(x,x_o,c) = (x + x_o)^c + Ax + B \tag{10b}$$

where A and B are selected so that the homogeneous boundary conditions are satisfied. In the example, $x_o = y_o = 0.03$, and $C_x = C_y = -0.25$. The solution is formulated to give steep gradients near the coordinate axes. For the finite-element computations, the solution, Eq. (10), is substituted into the governing equation (8a) which is solved for $Q(x,y)$. The heat generation Q is then integrated numerically over each element, and nodal temperatures are computed. The temperature contours from the exact solution, Eq. (10), are presented in Fig. 6.

Finite-element solutions were obtained for the following: 1) three meshes with uniform h refinement of square elements and 2) three adaptive remeshes where the remeshes started from the solution on uniform mesh 2. The meshes used in the remeshing approach are shown in Fig. 7, and solution errors are tabulated in Table 2.

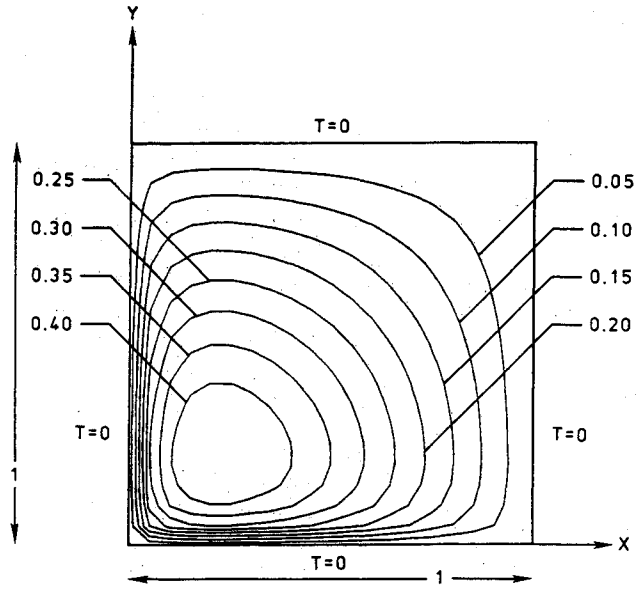


Fig. 6 Temperature contours for two dimensional heat conduction problem, Example 2.

The remeshes shown in Fig. 7 consist of a mixture of quad and triangular elements generally of good proportions. The elements do not follow any clear pattern that might be intuitively suggested, but as the meshes are adaptively created there is clear refinement of elements at the sharp gradients near the coordinate axes. At the same time larger elements are being created away from the boundaries in regions of smaller gradients. Although the remeshes are not aesthetically satisfying, they are effective in producing high-quality solution convergence as demonstrated in Table 2.

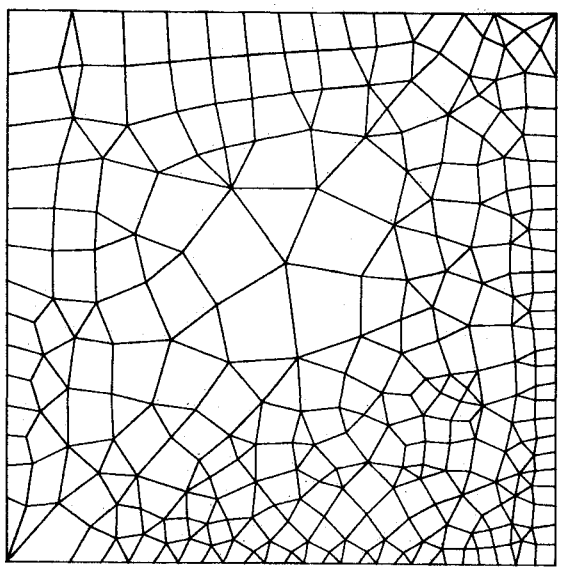
From Table 2, note that uniform h refinement reduces the global solution error at a rate less than the optimum rate, $0(h^2)$. This appears to be because of the steep gradients present in the solution. Additionally, the optimum convergence rate is $0(h^2)$ only as an asymptotic limit which apparently has not been reached.

The errors based on the remeshes are reduced at a faster rate than for uniform refinement. The first mesh, which produced mesh 2 with 299 unknowns, gives an error $\|\ e\ \| = 0.01969$, which is smaller than the error produced on uniform mesh 3, which has 441 unknowns.

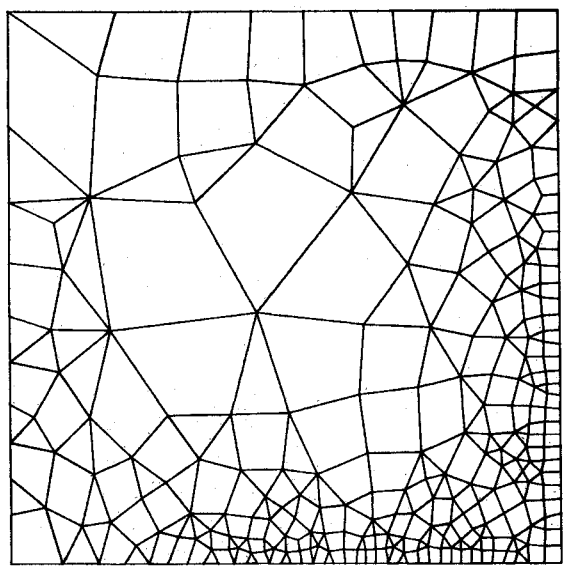
In addition, the second remesh which produces mesh 3 has reduced the number of unknowns from 299 to 288 and simultaneously has reduced the error. Finally, if one more uniform h refinement were made assuming an optimal convergence rate of four, the solution error would reduce from 0.03305 to 0.00826 for over 1600 unknowns. Yet, the third mesh that produced mesh 4 has an error of 0.00550 for only 572 unknowns. Clearly for this problem with steep gradients, the

Table 2 Example 2: Comparative solution errors

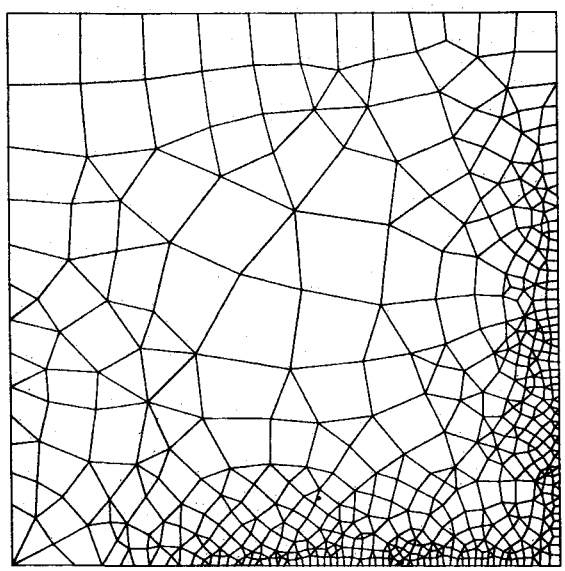
Uniform h refinement			
Mesh	Unknowns	h	Error, $\|\|e\|\|$
1	36	1.00	0.12720
2	121	0.50	0.07982
3	441	0.25	0.03305
Remeshing			
Mesh	Unknowns		Error, $\|\|e\|\|$
1	121		0.12720
2	299		0.01969
3	288		0.01270
4	572		0.00550



a) Mesh 2



b) Mesh 3



c) Mesh 4

Fig. 7 Example 2: adaptive remeshes.

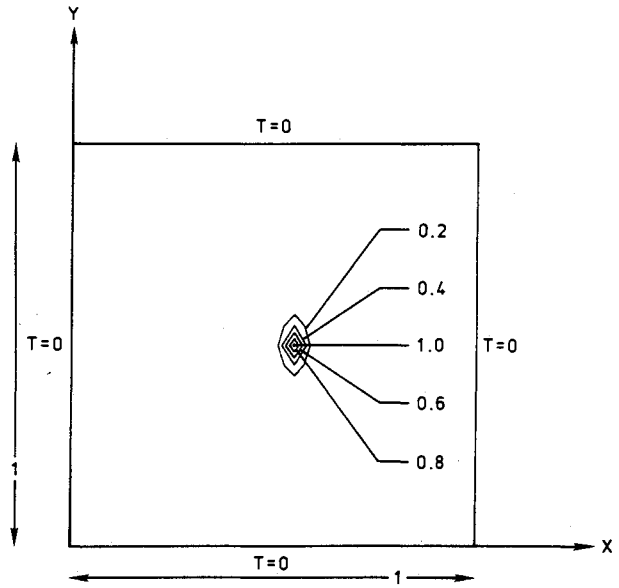
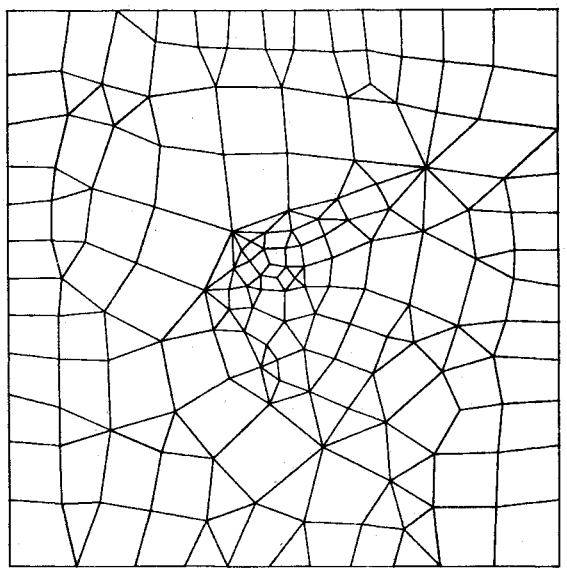
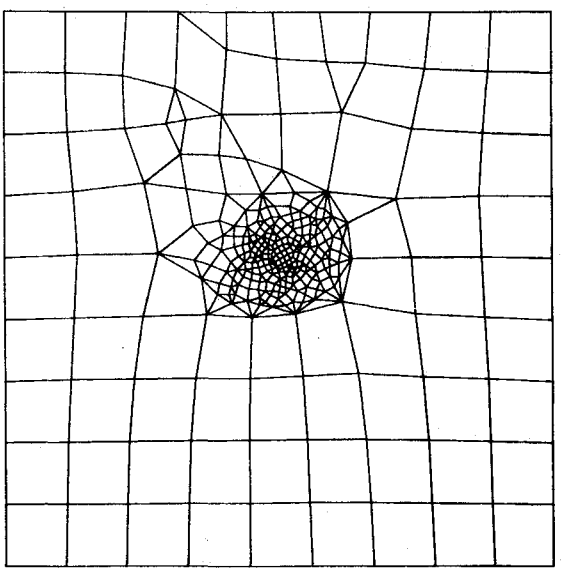


Fig. 8 Temperature contours for two-dimensional heat conduction problem, Example 3.



a) Mesh 2



b) Mesh 3

Fig. 9 Example 3: adaptive remeshes.

Table 3 Example 3: Remeshing solution errors

Type	Unknowns	Error, $\|e\|$
Uniform h, (21×21)	441	0.415000
Remesh 1	176	0.028700
Remesh 2	322	0.002868

remeshing approach produces higher solution quality for reduced computational effort than uniform h refinement.

Example 3

In Example 2, the gradients were roughly one dimensional with the solution varying steeply normal to the x and y axes. For this final example, a model problem[4] is solved where the gradients were more two dimensional. As before, the model problem assumes a solution that satifies Eq. (8a) and homogeneous boundary conditions. The solution takes the form

$$T(x,y) = \psi(x,x_o,\epsilon_x)\ \psi(y,y_o,\epsilon_y) \tag{11a}$$

$$\psi(x,x_o,\epsilon) = \frac{1}{\exp(\epsilon^{-1})}\exp\left(\frac{1}{(x-x_o)^2+\epsilon}\right) + Ax + B \tag{11b}$$

where again A and B are selected so that homogeneous boundary conditions are satisfied. In the example, $x_o = 0.55$, $y_o = 0.50$, and $\epsilon_x = 0.02$, $\epsilon_y = 0.05$. The solution gives very steep gradients near x_o, y_o. As in the previous example, the solution is substituted into the governing equation, Eq. (8a), to give a variable heating rate $Q(x,y)$ for the finite-element solution. The temperature contours from the exact solution, Eq. (11), are presented in Fig. 8.

Finite-element solutions were obtained for two remeshes starting from a solution on a uniform mesh. The meshes obtained from remeshing are shown in Fig. 9, and the solution errors are tabulated in Table 3.

Table 3 shows that for the first remesh there was an error reduction of about 15 even though the number of unknowns was reduced by more than one-half. For the second remesh, the number of unknowns are less than 75% of the unknowns in the original mesh, yet the solution error has been reduced by more than two orders of magnitude. The example shows the strong benefits that were obtained by adaptively refining a mesh in regions of steep gradients.

Concluding Remarks

A finite-element adaptive remeshing approach is described. The approach uses quadrilateral elements where possible, and triangles are introduced as needed. The remeshing approach uses a solution based on an old mesh to create a new mesh. Second derivatives of the previous solution are used to compute parameters that determine the size and orientation of elements on the new mesh. The new mesh is created one element at a time using a front concept. The front begins from the discretized boundary of the solution domain and advances into the domain as the mesh evolves. As the mesh construction continues, the front continuously changes shape and vanishes when the mesh is complete. In the present implementation, two options for creating quadrilaterals are tried first, and only when a well-shaped quad cannot be created is a triangle introduced. The resulting meshes are completely unstructured and are highly refined in regions of steep solution gradients; elements are relatively large in regions of small gradients.

The remeshing approach is applied to three pure conduction problems with exact analytical solutions. To assess the effectiveness of the approach, a global error measure was used to study solution convergence rates as meshes were adaptively refined. Comparison were also made with convergence rates for solutions obtained on successively refined uniform, structured meshes.

For a problem with a smooth solution with no steep gradients, the remeshing approach produced convergence rates of order h^2 the same as uniform refinement. Hence, for problems with few gradients the approach offers no advantage over uniform mesh refinement and probably would be a more expensive way to generate the meshes. However, for two examples with steep gradients, the approach consistently outperformed uniform refinement. In both examples, the remeshing approach gave much faster convergence rates. Moreover, in both cases there were instances where the error in the solution was reduced even though the number of unknowns was actually smaller than in the previous mesh. This desirable behavior has also been observed by Peraire and coauthors who had previously developed the method using all triangles. The present implementation of the method has some advantages over an all-triangle mesh including the fact that fewer elements are required with quadrilaterals. The approach offers potential for greater savings in three dimensions, where a single hexahedral element can replace at least five tetrahedrons.

This paper has demonstrated an approach for applying the remeshing concept to more general elements and demonstrated the benefits offered by adaptive remeshing for problems with steep gradients. The remeshing concept offers excellent potential for development as a general means of adaptive refinement for a wide variety of applications. The method will be very valuable to resolve highly localized phenomena with minimum computational effort.

Acknowledgments

The authors are pleased to acknowledge the support of this research by Grant NSG-1321 from the NASA Langley Research Center. We greatly appreciate the advice and encouragement of our technical monitor Allan R. Wieting, Head of Aerothermal Loads Branch. We are also grateful for the benefits of technical discussions with K. Morgan and J. Peraire of University College of Swansea, Wales. Their comments and the innovations they made in developing remeshing for triangles were very valuable and facilitated the development of the remeshing approach presented herein.

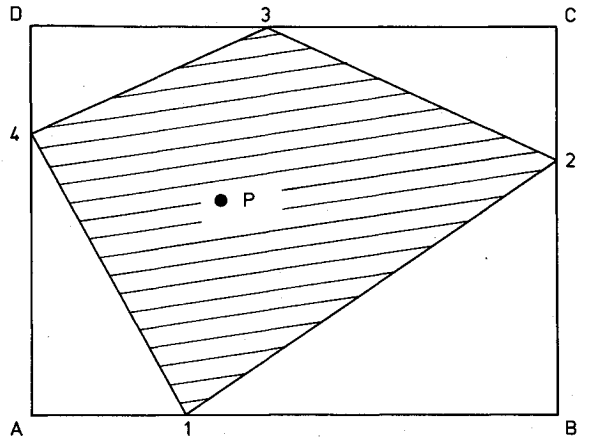
Appendix A: Search Algorithm

The interpolation of mesh generation parameters in the new mesh uses a search algorithm to identify the element on the background mesh where the point of interest P lies. This search is made on a global basis since the background mesh is unstructured.

An array is created in the program that gives

XMIN—minimum X coordinate
XMAX—maximum X coordinate
YMIN—minimum Y coordinate
YMAX—maximum Y coordinate

for all the elements on the background mesh. As the search begins, the coordinates of P are checked to determine if they lie within the limits for each element. This check is made one limit at a time so that a negative response avoids the remaining checks. When the response is positive for all four checks, the point lies in the neighborhood of an element shown as the rectangle $ABCD$ in the following sketch. A second set of checks are then made to determine if point P lies within the element (shaded region). The second set of checks are made by computing the areas of the triangles $12P$, $23P$, $34P$, and $41P$ where 1, 2, 3, 4, are the nodes numbered in an counterclockwise manner. If any of the areas are negative or zero (within a specified tolerance), the response is negative, and the search proceeds to the next element.



Appendix B: Computation of Derivatives

For the elements used in the remeshing method the value of the dependent variable within an element is interpolated from nodal values by

$$\Phi(x,y) = [N(x,y)]\ \{\Phi\} \tag{B1}$$

where, for a triangle, the interpolation functions $[N]$ are linear, and, for a quad, the interpolation functions are bilinear. From Eq. (B1), first derivatives can be computed, but it is not possible to compute the second derivatives that are needed for remeshing. As an alternate approach the following procedure[8] is used.

From Eq. (B1), element first derivatives are computed by direct differentation. For example,

$$\frac{\partial \Phi^e}{\partial x} = \left[\frac{\partial N}{\partial x}\right] \{\Phi\} \tag{B2}$$

Values of the first derivatives are computed at nodal points by assembling system equations from element contributions of the form,

$$\int_A \{N\}\ [N]\ dA \left\{\frac{\partial \Phi}{\partial x}\right\} = \int_A \{N\}\ dA\ \frac{\partial \Phi^e}{\partial x} \tag{B3}$$

The coefficient matrix on the left-hand side of Eq. (B3) is diagonalized to yield an explicit set of equations that are solved for the nodal value of the derivatives. The procedure given by Eq. (B3) may be interpreted as computing the nodal derivative as a weighted average of the derivatives from the elements surrounding the node. Element area factors serve as the weighting factors.

The computation of the second derivative follows the same steps. Element second derivatives are computed from nodal derivatives by

$$\frac{\partial^2 \Phi^e}{\partial x^2} = \left[\frac{\partial N}{\partial x}\right] \left\{\frac{\partial \Phi}{\partial x}\right\} \tag{B4}$$

and then nodal second derivatives are computed from system equations assembled from

$$\int_A \{N\}\ [N]\ dA \left\{\frac{\partial^2 \Phi}{\partial x^2}\right\} = \int \{N\}\ dA\ \frac{\partial^2 \Phi^e}{\partial x^2} \tag{B5}$$

The procedure described by Eqs. (B1–B5) lacks mathematical rigor, but it has been proven adequate for computing the second derivatives needed for remeshing. The second derivatives computed on the boundaries typically are less accurate than those computed at interior nodes.

References

[1]Oden, J. T. and Demkowicz, L., "Advances in Adaptive Improvements: A Survey of Adaptive Finite Element Methods in Computational Mechanics," *State of the Art Surveys in Computational Mechanics*, edited by A. K. Noor and J. T. Oden, American Society of Mechanical Engineers, New York, 1986.

[2]Eiseman, P., "Adaptive Grid Generation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, 1987, pp. 321–376.

[3]Oden, J. T., et al., "Adaptive Methods for Problems in Solid and Fluid Mechanics," *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, edited by I. Babuska, Wiley, New York, 1986, pp. 249–280.

[4]Demkowicz, L., Devloo, P., and Oden, J. T., "On an *h*-type Mesh-Refinement Strategy Based on Minimization of Interpolation Errors," *Computer Methods in Applied Mechanics and Engineering*, Vol. 53, 1985, pp. 67–89.

[5]Demkowicz, L. and Oden, J. T., "On a Mesh Optimization Method Based on Minimization of Interpolation Error," *International Journal of Engineering Science*, Vol. 24, No. 1, 1986, pp. 55–68.

[6]Carey, G. F., "Mesh Refinement and Redistribution," *Numerical Methods for Transient and Coupled Problems*, edited by R. W. Lewis, E. Hinton, P. Beltess, and B. A. Schrefler, Wiley, New York, 1987.

[7]Murman, E. M. and Shapiro, R. A., "Cartesian Grid Finite Element Solutions to the Euler Equations," AIAA Paper 87-0559, Jan. 1987.

[8]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466.

[9]Raymond, W. H. and Gardner, A., "Selective Damping in a Galerkin Method for Solving Wave Problems with Variable Grids," *Monthly Weather Review*, Vol. 104, No. 12, 1976, pp. 1583–1590.

[10]Devloo, P., Oden, J. T., and Strouboulis, T. , "Implementation of an Adaptive Refinement Technique for the SUPG Algorithm," *Computer Methods in Applied Mechanics and Engineering*, Vol. 61, No. 3, 1987, pp. 339–358.

[11]Baehmann, P. L., Wittchen, S. L., Shepard, M. S., Grice, K. R., and Yerry, M. A., "Robust, Geometrically Based Automatic Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 6, 1987, pp. 1043–1078.

[12]Lo, S. H., "A New Mesh Generation Scheme for Arbitrary Planar Domains," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 8, 1985, pp. 1403–1426.

[13]Carslaw, H. S. and Jaeger, J. C., *Conduction of Heat in Solids*, second edition, Oxford Univ. Press, Oxford, England, UK, 1959, pp. 170–171.