JOURNAL OF MATERIALS SCIENCE 38 (2003) 4175–4182

Adaptive modification for time evolving meshes

T. J. BAKER
MAE Department, Princeton University, Princeton, NJ 08540, USA
E-mail: baker@tornado.princeton.edu

A mesh adaptation scheme is presented that employs edge collapse to coarsen the mesh and selective point placement combined with an incremental Delaunay algorithm for mesh enrichment. The coarsening and enrichment procedures can be used in isolation to achieve a solution adapted mesh for time varying computations whose boundaries remain fixed. These procedures can also be combined with a mesh movement technique to form an adaptation algorithm that will modify the mesh of a domain whose shape is evolving in time. The adaptation techniques are described and examples are presented to illustrate the method for time dependent problems arising from crack nucleation studies and vortex shedding of incompressible flow over a cylinder.

1. Introduction

Many computational problems of engineering interest are essentially unsteady in nature and can only be realistically solved if their time dependence is fully recognized as part of the computation. For computations that involve a finite element analysis, this requires a mesh adaptive scheme that is able to follow the evolution of the computed solution and, if necessary, adjust to changes in the shape of the computational domain. For static adaptation (i.e., adaptation to the time dependent changes in the computed solution on a fixed computational domain) mesh coarsening and mesh refinement are the main tools used to achieve the required goal. For dynamic adaptation (i.e., meshes that evolve to follow changes in the shape of the computational domain) it is necessary to augment the mesh coarsening and refinement by a mesh movement algorithm. This paper presents a method to carry out both static and dynamic adaptation on unstructured meshes consisting of triangles in 2D and tetrahedra in 3D.

Static adaptation is illustrated by an example from fluid dynamics that shows an adaptive computation of vortex shedding for incompressible flow over a cylinder. Dynamic adaptation, in which the domain changes shape, is illustrated by an example of a finite element computation of crack nucleation.

2. Mesh coarsening

The method of coarsening that is currently implemented uses edge collapse. Given a triangle that is a candidate for removal, take its shortest edge and allow the two endpoints of this edge to collapse to a single point. As a result, this edge together with the two incident triangles is removed from the triangulation and the number of points in the mesh decreases by one. The effect of this operation, when repeatedly applied to a region of the triangulation, will result in a much coarser distribution of points. Unfortunately, the quality of the triangulation is also severely degraded. The mesh enrichment algorithm described in the next section dramatically improves the mesh quality, but even better results are obtained if the coarsened mesh is optimized prior to mesh enrichment. In the planar case, the coarsened mesh is optimized by swapping diagonals in order to maximize the minimum of the six angles of any pair of triangles with a common edge which form a convex quadrilateral. This operation will lead to the Delaunay triangulation of the coarsened point set for any planar triangulation [1].

A similar collapse procedure can be applied in three dimensions although the implementation is not so straightforward as in the planar case. It is necessary to identify the ring of tetrahedra incident to the candidate edge. The two end points of the edge are collapsed into a single point, the edge is removed and the ring of tetrahedra disappears from the mesh. It is necessary to check whether the modified tetrahedra are still valid (i.e., have positive volumes). In three dimensions, as in the planar case, it is advantageous to optimize the coarsened mesh prior to the enrichment stage. This is particularly important since slivers, which may have been created in the coarsened tetrahedral mesh, are not necessarily removed during mesh enrichment. Slivers are zero volume, or near zero volume, tetrahedra for which the ratio of circum-radius R to average edge length L is $O(1)$ [2].

Edge/face swapping procedures are used to remove slivers [3]. The tetrahedra are sorted into a list, ordered according to the ratio of circum-radius to in-radius, with the tetrahedron having the largest ratio appearing at the front. The dihedral angle of each tetrahedron is computed and any tetrahedron with a maximum dihedral angle greater than 120° is considered to be a candidate for removal by edge/face swapping. Only tetrahedra with large ratios of circum-radius to in-radius are likely to be slivers. These tetrahedra typically have two opposite edges whose dihedral angles are very large with the remaining four dihedral angles assuming very small values. An effective strategy is to consider in turn each of the two edges whose dihedral angles are largest and examine the possibility of removing the edge, and thus eliminating the sliver, by an edge/face swap.

It is possible that non-convexity of the tetrahedral ensemble prevents the removal of a sliver. In this case, the placement of a new point inside the tetrahedral ensemble followed by further iterations of the edge/face swapping procedure will usually eliminate all remaining slivers [3].

3. Mesh enrichment

The decision whether or not to enrich a particular region of the mesh is based on a comparison between the actual local length scale h (e.g., element width, circum-radius) and the desired length scale specified by a scalar variable ρ called the length density function. Suppose, for example, that the goal is to convert an existing volume mesh into one with a smooth gradation in mesh density throughout the domain and such that the density of the volume mesh near the boundary surface matches the mesh size of the boundary triangulation. The value of the mesh density function at each point on the boundary is computed as the average length of the incident boundary edges [4]. Solving Laplace's equation on the current volume mesh, using the values of the length density function at the boundary as Dirichlet data, will yield appropriate values of ρ at each mesh point. If the value of ρ at any position in the mesh is less than the actual local length scale h then the mesh is refined by the insertion of an extra point followed by a local mesh reconstruction using an incremental Delaunay algorithm. Several possibilities have been considered for selecting the position of point placement (e.g., at element barycenters [4], along edges [5], at element circum-centers [6] or along Voronoï segments [7]). The Voronoï segment method [7, 8] works extremely well in 2D but does not extend readily to 3D. The circum-center point insertion (CPI) method generates provably good quality meshes in 2D [8]. In 3D the CPI method can generate meshes with good element quality provided care is taken to remove all slivers [3].

For all cases shown in this paper, the actual local length scale h is proportional to the element circumradius and point insertion is based on the CPI strategy. A element is thus marked for refinement if the circumradius is too large when compared with the length density function ρ. The local mesh reconstruction exploits a constrained Delaunay algorithm that can be applied to any valid tetrahedral or triangular mesh. Elements to be removed are those whose circumspheres contain the new point subject to the constraint that no boundary face should be removed and that the new point is always visible from the exposed faces of the remaining elements. This particular procedure for inserting a new point is an integral component of the constrained Delaunay method that has been successfully exploited in a number of tetrahedral mesh generators [3–5].

4. Static adaptation

Extra refinement at any mesh position can be achieved by locally reducing the length density function ρ to a value lower than the actual length scale h. Mesh coarsening can be achieved in a similar way by increasing the length density function ρ to a value larger than the actual mesh length scale h. Let E be a scalar function, defined over the computational domain, that provides an estimate of the solution error. This may be some form of *a posteriori* error estimate based on the computed solution, or alternatively, it may be constructed from a specific physical property [9]. An adaptation trigger e is then constructed to effect enrichment in regions of rapid change in the computed solution. The average value $\bar{E}$ of the error estimator E is first computed and the standard deviation σ is formed where $\sigma^2 = \overline{E^2} - \bar{E}^2$. Let

$$e = \frac{E - \bar{E} - \lambda\sigma}{\sigma} \tag{1}$$

where λ is a user defined constant. Then

$$\rho_{\text{new}} = \begin{cases} \dfrac{\rho}{1 + \alpha \min(e, 1)}, & E > \bar{E} + \lambda\sigma \\ \rho, & E \leq \bar{E} + \lambda\sigma \end{cases} \tag{2}$$

In other words, when the error estimator E exceeds the average plus λ times σ, the length density function ρ is reduced in size. ρ is reduced by at most a factor $\frac{1}{1+\alpha}$ where α is a $O(1)$ user defined constant. Larger values of α create more rapid changes in mesh density while larger values of λ increase the threshold at which adaptation kicks in.

In a similar way, let μ be a constant defined by the user so that the mesh will be coarsened if E falls below the average $\bar{E}$ less μ times the standard deviation σ. Set,

$$e = \frac{\bar{E} - \mu\sigma - E}{\sigma} \tag{3}$$

and define the new length density function to be

$$\rho_{\text{new}} = \begin{cases} \rho(1 + \alpha \min(e, 1)), & E < \bar{E} - \mu\sigma \\ \rho, & E \geq \bar{E} - \mu\sigma \end{cases} \tag{4}$$

An example in which static adaptation is applied can be found in the work of Lin [10]. He presents a computation of incompressible, laminar flow over a cylinder for Reynolds numbers between 50 and 180. Vortices shed from the cylinder are convected downstream so that the flow in the wake region exhibits a periodicity related to the vortex shedding frequency. For this problem the estimator E was taken to be the total pressure. Fig. 1 shows a view of the unadapted mesh around the cylinder while Fig. 2 shows the corresponding view of the mesh a given instant of time t after adaptation to the vorticity field. Figs 3 and 4 show the near wake region of the adapted mesh and the associated flow solution at time t. Note that the mesh density matches the intensity of the vorticity and varies in a reasonably smooth manner between regions of high and low mesh density. Figs 5 and 6 present similar plots of the near field vorticity distribution and adapted mesh at a time that is one half of a period T later. It is evident that the mesh adaptation has successfully tracked the moving vortices, refining the mesh in the new vortex positions and coarsening the part of the mesh that has been

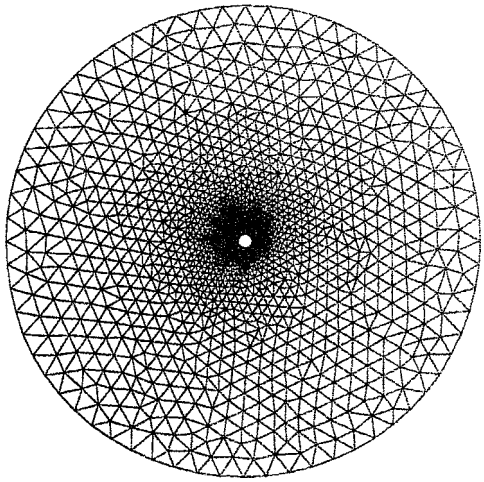

Figure 1 View of complete unadapted mesh at time instant t.

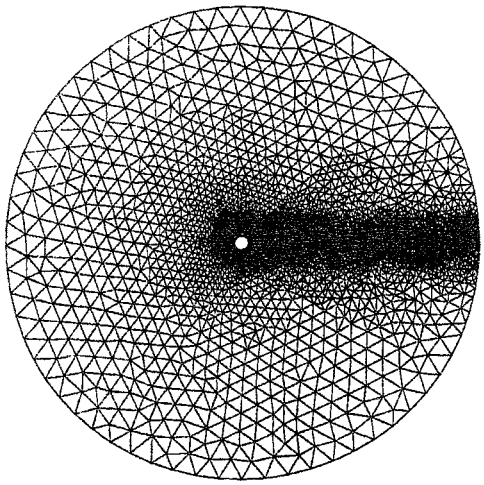

Figure 2 View of complete adapted mesh at time instant t.

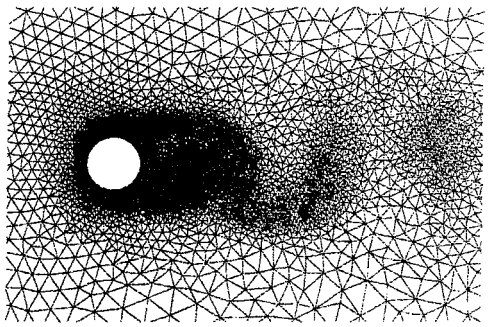

Figure 3 Close up view of the near wake region of the adapted mesh at time instant t.

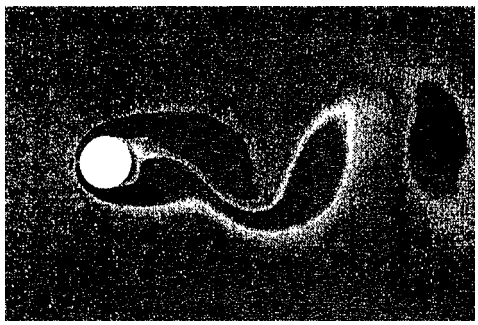

Figure 4 Vorticity distribution in the near wake region at time instant t.

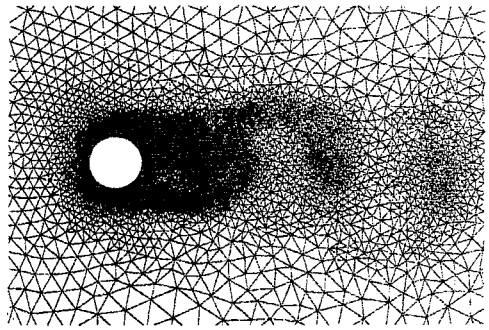

Figure 5 Close up view of the near wake region of the adapted mesh at time $t + \frac{1}{2}T$.

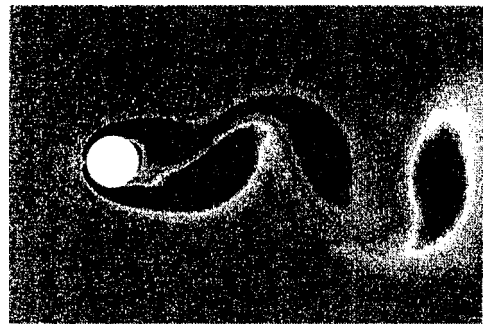

Figure 6 Vorticity distribution in the near wake region at time $t + \frac{1}{2}T$.

vacated by the vortex. The computational cost of mesh adaptation is small compared with the cost of one time step of the flow solver. Lin was therefore able to apply mesh adaptation repeatedly during each iterative computation that advanced the flow field one time step with the result that there is no time lag between the adapted mesh and the computed solution.

Other types of simulation will require a different type of error estimator E. In reference [11], for example, a generalized solution sensor was developed that could be applied to both inviscid and viscous flowfields. The crack nucleation problem shown later computes the stress field in a solid under lateral tension [12, 13]. This simulation exploits both static and dynamic adaptation; static adapation is driven by an error estimator formed by the strain energy while dynamic adaptation is needed to follow the development of the crack.

5. Dynamic adaptation

Mesh modification for time evolving domains can be carried out by a three stage combination of mesh movement, mesh coarsening and mesh enrichment. One application of this three stage procedure forms one cycle of dynamic adaptation. Fig. 7 shows the initial mesh for a circular disk lying at the left hand side of a rectangular domain. Suppose that it is required to move the disk to the right hand side of the computational domain. Even if a mesh movement scheme could be devised to achieve this goal, the resulting mesh would exhibit very poor quality. For a given mesh movement scheme, the extent of domain deformation that can be accommodated during one cycle depends on how far the mesh can be stretched without creating negative

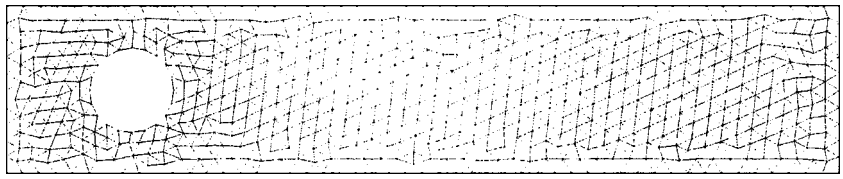

Figure 7 Original mesh for disk inside duct.

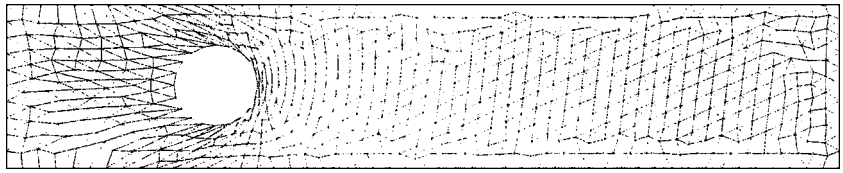

Figure 8 Mesh after translating disk 1 diameter to the right.

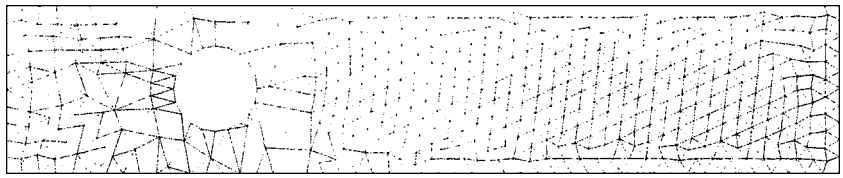

Figure 9 Mesh after disk translation of 1 diameter and mesh coarsening.

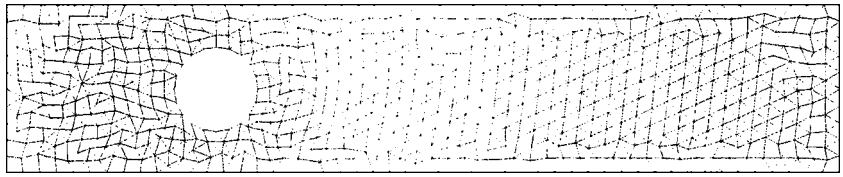

Figure 10 Mesh after 1 complete cycle of movement and modification.

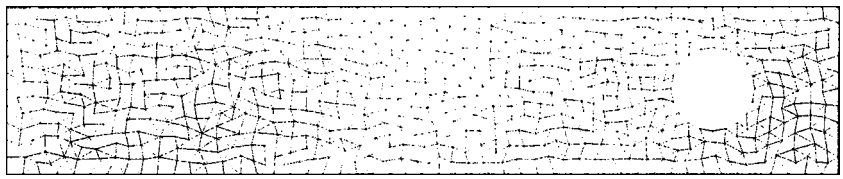

Figure 11 Mesh after 7 complete cycles of movement and modification.

element volumes (see Fig. 8). Mesh coarsening is then carried out to remove points associated with elements that have become badly shaped during the mesh movement stage (see Fig. 9). Finally, mesh enrichment serves to re-create a mesh whose element quality is comparable to that of the original mesh (see Fig. 10). This modification cycle can be repeated any number of times to obtain a good quality mesh for any homotopic deformation. This is illustrated in Fig. 11 which shows the result of seven consecutive cycles of dynamic adaptation. Although the final mesh, shown in Fig. 11, has a different appearance than the initial mesh of Fig. 7, measures of mesh quality (e.g., minimum and maximum angle) show that the meshes are comparable. At each stage one is operating on a valid (i.e., conforming, space filling and non-overlapping) mesh which thus avoids the difficulties that are associated with opening up pockets and remeshing. In other words, dynamic mesh adaptation is achieved by transforming the mesh through a series of intermediate states, each state being obtained from its predecessor by either mesh movement, an edge collapse operation or enrichment by point insertion.

6. Mesh movement

Suppose that the domain D has boundary B at time t and that during time step Δt domain D deforms into D' with boundary B'. One seeks a homeomorphism from D to D' so that a mesh on D is mapped to a valid mesh on D'. Suppose that the point $(x, y, z)\varepsilon D$ is mapped to $(x', y', z')\varepsilon D'$ and define $\mathbf{u} = (u_1, u_2, u_3)$ where $u_1 = x' - x, u_2 = y' - y$ and $u_3 = z' - z$. One may use the mesh on D to solve a suitable elliptic partial differential equation and thus compute the displacement vector $\mathbf{u}$ at all mesh points on D. Adding the computed displacement u_1 to x, u_2 to y and u_3 to z one obtains the new mesh points (x', y', z') on D'.

For example, one could solve the Laplace equations $\nabla^2 u_1 = 0, \nabla^2 u_2 = 0, \nabla^2 u_3 = 0$ subject to the Dirichlet boundary conditions u_1, u_2 and u_3 given on B.

In the absence of any further mesh modification, it is clearly desirable to move the mesh as far as possible before inversion of one or more elements causes mesh breakdown. Elements most prone to inversion are typically those with a small volume and/or aspect ratio. Mesh breakdown can be delayed by keeping the displacement nearly constant in regions occupied by these particular elements. Let V denote the volume of an element, with $V_{\max}$ and $V_{\min}$ respectively the maximum and minimum element volumes. Define a variable diffusivity by

$$\tau = 1 + \frac{V_{\max} - V_{\min}}{V} \tag{5}$$

Clearly, $\tau = 1$ for a mesh whose elements all have the same size. For a mesh with varying element size, $\tau_{\max} = \frac{V_{\max}}{V_{\min}}$ occurs for elements whose volume is equal to $V_{\min}$. Alternatively, the diffusivity can be made to depend on the aspect ratio of the element. Solving the equations $\nabla \cdot (\tau \nabla u_i) = 0, i = 1, 2, 3$ provides a mesh movement scheme [14] that allows for greater deformation in regions where the elements have large volume while maintaining a greater degree of mesh rigidity where element volumes are small. By iterating this procedure it is possible to enhance further the degree of mesh deformation that can be achieved.

A disadvantage of solving Laplace's equation, or its variant, is the lack of any dependence between the components of displacement. If the mesh is sheared in a direction parallel to the x-axis, for example, the effect of this shear is not felt by the other components of displacement. A more robust mesh movement scheme that overcomes this limitation can be constructed by modeling the domain as an elastic solid and solving the equilibrium equations for the stress field [15, 16]. In terms of the displacement vector **u** the strain tensor can be written as

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad i, j = 1, 2, 3 \tag{6}$$

For an isotropically elastic solid the stress tensor is defined as

$$\sigma_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij} \quad i, j = 1, 2, 3 \tag{7}$$

where λ and μ are the Lamé constants, δ_{ij} is the Kronecker delta and the summation convention (viz. $\epsilon_{kk} = \epsilon_{11} + \epsilon_{22} + \epsilon_{33}$) has been invoked. If there is no distributed body force the stress field satisfies the equation

$$\frac{\partial \sigma_{ij}}{\partial x_i} = 0 \tag{8}$$

Dividing by the shear modulus μ leads to an equation that depends only on the parameter λ/μ. Alternatively, one can introduce Poisson's ratio

$$\nu = \frac{\lambda}{2(\lambda + \mu)} \tag{9}$$

and consider this to be the user defined parameter. It is again possible to increase the rigidity of the mesh in regions of small element size, and/or bad element aspect ratio, by modifying the coefficients λ and μ [15, 16] in much the same manner that variable diffusivity was introduced above into Laplace's equation.

A linearly varying displacement field is an exact solution of Laplace's equation and also of the equation for elastic equilibrium (assuming constant coefficients of elasticity). If the discrete equations, used to approximate either equation, fail to preserve this property then inaccuracies in the numerical solution can lead to premature mesh breakdown. A linearly varying displacement field will be an exact solution of the discrete equations if one uses a finite element scheme with linear elements. An essentially equivalent solution procedure is a vertex based finite volume scheme which assumes a linear variation of the displacement components across each element and uses a trapezoidal integration of the flux terms. For a linearly varying displacement field, the derivative of each component, and hence the flux terms, will be constant over the entire domain. It then follows that the trapezoidal approximation to the surface integration sums to zero for each mesh point.

When mesh coarsening and enrichment are introduced as part of a dynamic adaptation cycle, it is no longer necessary to move the mesh as far as possible during each mesh movement stage. In fact, considerations of computational efficiency favor a mesh movement scheme that permits a smaller degree of mesh deformation if this can be accomplished at a much smaller computational cost than more robust mesh movement methods. In practice, the combined mesh coarsening and enrichment stages account for no more than 20% of the total computation time if the simplest mesh movement scheme is used. For more robust mesh movement schemes the computational cost of mesh coarsening and enrichment can be less than 5% of the total time.

A particularly simple and cost effective mesh movement scheme is obtained if one approximates Laplace's equation by summing differences of the dependent variable along each edge incident to a mesh point. Introducing the variable diffusivity adds only a negligible computational overhead. Let ϕ be the dependent variable and let $\nabla \cdot (\tau \nabla \phi) = 0$ be the equation to be solved. If ϕ_0, respectively ϕ_k, represents the discrete approximation to ϕ at the mesh point P_0, respectively P_k, then the residual at P_0 is given by

$$\nabla \cdot (\tau \nabla \phi)|_0 \approx \frac{1}{S_0} \sum_{k=1}^{m} \sigma_{0k} (\phi_k - \phi_0) \tag{10}$$

where the k summation is over the m edges $\{\overline{P_0 P_k} \mid k = 1, \ldots, m\}$ incident to the point P_0, the coefficient $\sigma_{0k} = \sum_j \tau_j$ where the j summation is over all the elements incident to the edge $\overline{P_0 P_k}$ and the term $S_0 = \sum_{k=1}^{m} \sigma_{0k}$.

Using the superscript n to denote the nth iteration, one can write a point Jacobi scheme as

$$\phi_0^{n+1} = \phi_0^n + \frac{\epsilon}{S_0} \sum_{k=1}^{m} \sigma_{0k} \left(\phi_k^n - \phi_0^n\right) \tag{11}$$

where $0 < \epsilon \leq 1$ is a relaxation factor. Even if there is no variable diffusivity, this simple mesh movement scheme is not exact for a linearly varying displacement field and will permit only a limited deformation of the mesh before element inversion occurs. It is, however, computationally far less expensive than solving the stress equilibrium equations by a finite element method, and the above scheme is therefore the preferred mesh movement method for use in the dynamic adaptation procedure.

7. Boundary enrichment

If the boundary of the computational domain changes shape during the computation, it will be necessary to modify the triangulation of the boundary surface. Coarsening of the boundary triangulation occurs automatically as part of the volume coarsening procedure. It is, however, necessary to enrich the surface prior to enrichment of the volume mesh. A boundary face is currently marked for refinement if the maximum edge length is more than twice the minimum edge length. Future developments will seek to enrich the boundary surface based on an assessment of the local curvature.

Each boundary face that has been selected in this way is split by bisecting its longest edge. Inserting a mesh point on a boundary edge will cause both incident boundary faces to be split (see Fig. 12). It follows that the state of the boundary triangulation after refinement will depend on the order in which the boundary edges are taken. A list of boundary faces that have been marked for refinement is therefore created and this list is ordered according to triangle aspect ratio (defined as the ratio of circum-radius to in-radius). Starting with the face whose aspect ratio is largest, the longest edge is bisected and the two incident boundary faces are split. Each tetrahedron that is incident to the edge is also split into two new tetrahedra. This procedure is applied in turn to the longest edge of each boundary face that has been marked for refinement. The length density function is recalculated on the boundary and a Laplacian solver is used to distribute the length density function throughout the volume mesh as described in the previous section. Delaunay based terahedral refinement is then applied to the volume mesh to complete the enrichment procedure.

It is important to ensure that each new boundary point lies on, or at least close to, the true boundary surface. If this information is not readily accessible then the position can be approximated reasonably well by means of Hermite interpolation. A similar approach has been suggested by Weatherill *et al.* [17] who interpolate new points on boundary faces. The surface enrichment used here only requires the interpolation of new points at the midpoints of boundary edges. The resulting edge and face splitting is applied recursively in a manner that maintains a reasonable level of surface mesh quality. Inserting new boundary points at edge midpoints in a recursive fashion leads to a considerable simplification of the boundary enrichment procedure. There is no need to allow for the multiple splitting patterns that arise when a boundary face is split along one, two or three edges at a time. In addition, Hermite interpolation at the midpoint of an edge assumes a particularly simple form that would not be the case if the new point were placed on the interior of a boundary face.

Let the two endpoints of the boundary edge be P_0 with position vector $\mathbf{x}_0$ and P_1 with position vector $\mathbf{x}_1$, and let the unit normals at these points be $\mathbf{n}_0$ and $\mathbf{n}_1$ respectively. The surface normal at each boundary point is approximated by averaging the normals of the incident boundary faces.

Let $\mathbf{n}$ be the unit normal to the plane Γ in which the new point Q should lie. This plane also contains points P_0 and P_1 and thus the edge joining P_0 to P_1. Fit a cubic polynomial lying in the plane Γ through the points P_0 and P_1 whose tangents at P_0 and P_1 are parallel to the corresponding surface tangent planes (see Fig. 13). Let $\mathbf{c}$ be the vector from the edge midpoint to the point Q which lies halfway along the interpolating cubic curve. The correction $\mathbf{c}$ thus represents a displacement in the plane Γ that should be added to the coordinates of the edge midpoint in order to obtain an approximation to the true boundary surface.

Let $\mathbf{t}_0$ and $\mathbf{t}_1$ be the tangent vectors lying in the plane Γ. Thus,

$$\mathbf{t}_0 = \frac{\mathbf{n}_0 \times \mathbf{n}}{|\mathbf{n}_0 \times \mathbf{n}|}, \quad \mathbf{t}_1 = \frac{\mathbf{n}_1 \times \mathbf{n}}{|\mathbf{n}_1 \times \mathbf{n}|} \tag{12}$$

Let α_0 be the angle between $\mathbf{t_0}$ and the edge vector $\mathbf{x}_1 - \mathbf{x}_0$. Similarly, define α_1 as the angle between $\mathbf{t_1}$ and $\mathbf{x}_1 - \mathbf{x}_0$ and let $m_0 = \tan\alpha_0$, $m_1 = \tan\alpha_1$, be the imposed slopes of the interpolating curve at each endpoint. For a point lying half way between P_0 and P_1, Hermite interpolation takes a particularly simple form.

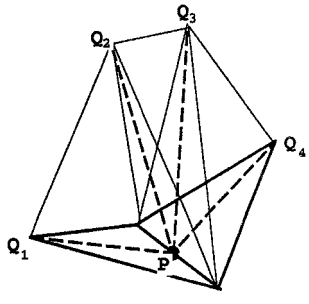

Figure 12 Splitting of boundary faces and tetrahedra.

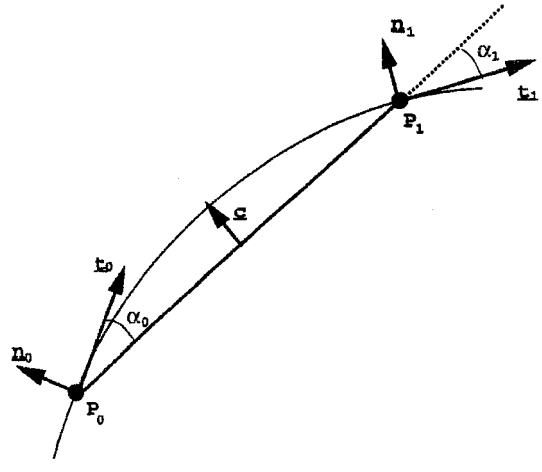

Figure 13 Interpolation of boundary point at edge midpoint.

The displacement is given by

$$\mathbf{c} = \frac{(m_0 - m_1)}{8}\,\mathbf{n} \times (\mathbf{x}_1 - \mathbf{x}_0) \tag{13}$$

In general, the plane Γ, in which the interpolated point should lie, is chosen to be the plane that bisects the dihedral angle between the two boundary faces incident to the edge joining P_0 and P_1. This choice is, however, undesirable at corners or salient edges where the angle between the normals of adjacent faces undergoes a large change. This situation typically occurs when two surfaces intersect. It is then preferable to place the new point on one of the two new surfaces so that Γ contains one of the two faces that are incident to the edge joining P_0 to P_1.

8. Mesh deformation and quality measures

A key feature in the successful implementation of the dynamic adaptation procedure is the use of mesh deformation and quality measures to monitor the time evolution of the mesh. The extent of boundary deformation that can be tolerated during the mesh movement stage is not known a priori. It is therefore necessary to allow the option of automatically repeating the mesh movement stage with a smaller boundary deformation if any of the deformed elements has a negative volume.

More critical, however, is the need to identify those elements that should be removed during the coarsening stage. On the one hand, it is possible to classify elements according to a deformation measure that assesses the relative change in element shape as a result of mesh movement. On the other hand, one can exploit a quality measure to assess the instantaneous element shape or aspect ratio. It turns out that the most effective strategy is to coarsen the mesh based on a combined assessment of both element deformation and element quality.

Consider a tetrahedron whose vertices are given by the points with position vectors $\mathbf{x}_n$ where $n = 0, 1, 2, 3$. The position vector $\mathbf{x}_n$ is a column vector of the three coordinates ($\mathbf{x}_n^T = (x_n, y_n, z_n)$). The edge matrix T_n is defined as

$$T_n = (-1)^n(\mathbf{x}_{n+1} - \mathbf{x}_n,\ \mathbf{x}_{n+2} - \mathbf{x}_n,\ \mathbf{x}_{n+3} - \mathbf{x}_n) \tag{14}$$

where addition in the indices is to be interpreted as modulo four (i.e., $\mathbf{x}_{n+4} \equiv \mathbf{x}_n$). Thus, T_n is formed by the three edge vectors joining the vertex at $\mathbf{x}_n$ to each of the three remaining vertices. The determinant $|T_n|$ of the matrix is equal to six times the volume of the tetrahedron (twice the area of the triangle in the 2D case). If a right handed rule is assumed for the edge ordering then $|T_n| > 0$ for elements with positive volume.

In the definition of the edge matrix T_n the vertex n plays a special role. Call this the corner vertex for the edge matrix T_n. Now let F be the elementary column exchange matrix

$$F = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \tag{15}$$

It follows that

$$T_{n+1} = T_n F = T_0 F^n \tag{16}$$

For any given element there is a sequence of $d+1$ edge matrices T_n, $n = 0, \ldots, d$ where $d = 2$ for a triangle and $d = 3$ for a tetrahedron. It is clear that any intrinsic property of the element, such as a deformation measure or shape measure, should not depend on the choice of the corner vertex.

Suppose that the element defined by the edge matrix T_n is mapped, under the action of a mesh movement procedure, to an element whose corresponding edge matrix is $\tilde{T}_n$. Define the deformation matrix associated with this mapping by

$$\tilde{T}_n = A_n T_n \quad \text{so that } A_n = \tilde{T}_n T_n^{-1} \tag{17}$$

Then,

$$A_{n+1} = \tilde{T}_{n+1} T_{n+1}^{-1} = \tilde{T}_n F F^{-1} T_n^{-1} = A_n \tag{18}$$

The matrix A_n is therefore independent of the choice of the corner vertex n and one may write A for the deformation matrix.

Now use the polar decomposition theorem to write A as $A = PU$ where U is a unitary matrix representing pure rotation and P is a positive definite matrix whose eigenvalues correspond to the modes of element distortion. An eigenvalue larger than 1 represents a stretching while a compression is indicated by an eigenvalue less than 1. The eigenvalues of the dilatation matrix P, also known as the singular values of A, are found by taking the square root of the eigenvalues of AA^T. Note that A^TA and AA^T have the same eigenvalues even though their eigenvectors will generally be different.

Let σ_{max} and σ_{min} be the maximum and minimum singular values of A (i.e., the maximum and minimum eigenvalues of the dilatation matrix P). If A corresponds to a pure rotation then $\sigma_{max} = \sigma_{min} = 1$ and a uniform scaling by a factor μ would result in $\sigma_{max} = \sigma_{min} = \mu$.

When distortion is so severe that an eigenvalue of the dilatation matrix P becomes zero, the determinant of A and hence also the determinant of the new edge matrix $\tilde{T}_n$ is zero. It follows that the volume of the element has shrunk to zero and any further deformation will create elements with negative signed volumes. In practice, it seems prudent to repeat the mesh movement with a smaller boundary deformation if any of singular values falls below 0.1 in size.

Several options are available for constructing a element quality measure to assess the instantaneous element shape or aspect ratio. In references [11, 18] the ratio of average edge length to in-radius was used. Alternatively, one can construct a shape measure based on a unitarily invariant matrix norm [19, 20] by considering the deformation matrix associated with the transformation of a reference equilateral element.

9. Crack nucleation study

This technique has been applied to a variety of problems, including the simulation of crack propagation which involves both static adaptation as well as dynamic adaptation to adjust the mesh for the deformation that occurs as the crack penetrates the solid. Fig. 14 shows the mesh for a rectangular shaped solid that is under lateral tension. An initial defect, modeled by a small depression, is evident on the upper surface. The external tension that is applied to the left and right sides of the domain induces a stress field in the solid. This stress field causes material to migrate along the upper surface and away from the region of highest strain energy at the tip of the defect. The resulting deformation in the shape of the upper boundary changes the stress field and increases the strain energy. The defect thus evolves into a crack that steadily penetrates further into the solid. Fig. 15 shows the resulting domain and adapted mesh at a particular instant after the crack has formed. The mesh has adapted automatically to the change in domain shape and has become highly enriched in the vicinity of the crack tip in order to resolve the large strain energy values in that region. Further details of this study and several computational results can be found in references [12, 13].

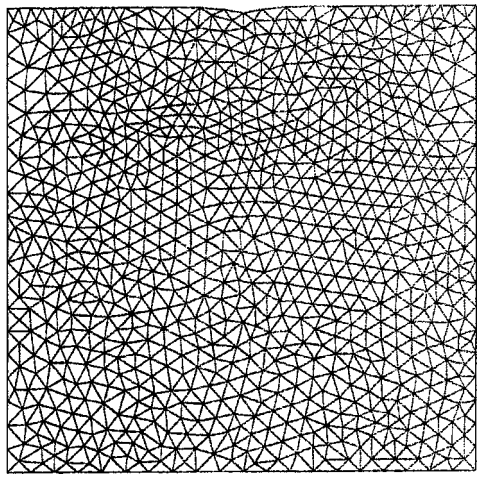

Figure 14 Initial domain and mesh for simulation of crack propagation.

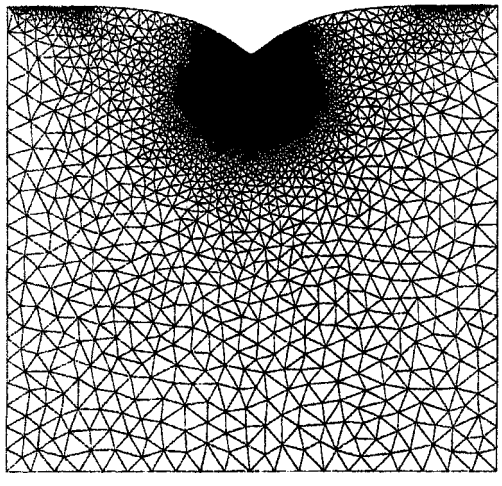

Figure 15 Deformed domain with adaptive refinement near crack tip.

Acknowledgments

Support for this work has been provided by NASA Ames under grant NAG-2-1327, by the DOE and NSF under subgrants 00-250 and 00-255 from the University of Illinois, and by the NSF through grant CMS-9988788 (Dr. J. Larsen-Basse, program director). The support of these sponsors is gratefully acknowledged.

References

1. C. L. LAWSON, "Mathematical Software III" (Academic Press, 1977) p. 161.
2. T. J. BAKER, "Element Quality in Tetrahedral Meshes," in Proc. 7th Int. Conf. on Finite Element Methods in Flow Problems, 1989, p. 1018.
3. T. J. BAKER and J. C. VASSBERG, "Tetrahedral Mesh Generation and Optimization," in Proc. 6th International Conference on Numerical Grid Generation, ISGG, 1998, p. 337.
4. D. L. MARCUM and N. R. WEATHERILL, "Unstructured Grid Generation using Iterative Point Insertion and Local Reconnection," AIAA Paper 94-1926 (1994).
5. P. GEORGE and H. BOROUCHAKI, Delaunay Triangulation and Meshing, Hermes, 1998
6. D. G. HOLMES and D. D. SNYDER, "The Generation of Unstructured Triangular Meshes using Delaunay Triangulation," in Proc. 2nd International Conference on Numerical Grid Generation, edited by Sengupta, *et al.* (Pineridge Press, 1988) p. 643.
7. S. REBAY, *J. Comp. Phys.* **106** (1993) 125.
8. T. J. BAKER, "Frontiers of Computational Fluid Dynamics," edited by D.A. Caughey and M.M. Hafez (J. Wiley and Sons, 1994) p. 101.
9. *Idem.*, *Finite Elem. Anal. Design* **25** (1997) 243.
10. P. LIN, L. MARTINELLI and T. J. BAKER, "Two-Dimensional Implicit Time Dependent Calculations for Incompressible Flows on Adaptive Unstructured Meshes," AIAA 15th Computational Fluid Dynamics Conference, AIAA Paper 2001-2655, Anaheim, CA, June 2001.
11. P. A. CAVALLO and T. J. BAKER, "Efficient Delaunay-Based Solution Adaptation for Three-Dimensional Unstructured Meshes," AIAA 38th Aerospace Sciences Meeting, AIAA Paper 2000-0809, Reno, NV (Jan. 2000).
12. Z. SUO, J. PREVOST and T. J. BAKER, "Delayed Fracture of Material Structures of Small Feature Sizes," 5th US National Congress of Computational Mechanics, U.C. Boulder, August 1999.
13. J. H. PREVOST, T. J. BAKER, J. LIANG and Z. SUO, *Int. J. Solids Struct.* **38** (2001) 5185.
14. A. MASUD and T. J. R. HUGHES, "A Space-time Finite Element Method for Fluid-structure Interaction," SUDAM Report No. 93-3, Stanford University, Stanford, CA (1993).
15. A. A. JOHNSON and T. E. TEZDUYAR, *Comp. Meth. Appl. Mech. Eng.* (1996) 351.
16. P. A. CAVALLO, R. A. LEE, A. HOSANGADI and S. M. DASH, "Dynamic Unstructured Grid Methodology with Application to Aero/Propulsive Flowfields," AIAA Paper 97-2310 (1997).
17. N. P. WEATHERILL, O. HASSAN, M. J. MARCHANT and D. L. MARCUM, "Adaptive Inviscid Solutions for Aerospace Geometries on Efficiently Generated Unstructured Tetrahedral Meshes," AIAA Paper 93-3390 (1993).
18. T. J. BAKER and P. A. CAVALLO, "Dynamic Adaptation for Deforming Tetrahedral Meshes," AIAA 14th Computational Fluid Dynamics Conference, AIAA Paper 99-3253, Norfolk, VA (1999).
19. L. A. FREITAG and P. A. KNUPP, "Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number," 8th Int. Meshing Roundtable, South Lake Tahoe, CA, USA, October 1999.
20. T. J. BAKER, "Deformation and Quality Measures for Tetrahedral Meshes," Eccomas00, Barcelona, Spain, Sept. 2000.