Journal of Computational Physics **172,** 99–118 (2001)
doi:10.1006/jcph.2001.6816, available online at http://www.idealibrary.com on IDEAL®

Volume Conserving Smoothing for Piecewise Linear Curves, Surfaces, and Triple Lines

Andrew Kuprat,*[,1] Ahmed Khamayseh,† Denise George,‡ and Levi Larkey§

*Theoretical Division, Group T-1, Mailstop B-221, Los Alamos National Laboratory, Los Alamos, New Mexico 87545; †Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831; ‡Theoretical Divison, Los Alamos National Laboratory, Los Alamos, New Mexico 87545; and §Texas Institute for Computational and Applied Mathematics, University of Texas, Austin, Texas 78712

E-mail: *kuprat@lanl.gov, †khamaysehak@ornl.gov, ‡dcg@lanl.gov, and §levi@ticam.utexas.edu

Received October 31, 2000; revised April 4, 2001

We present smoothing algorithms for piecewise linear curves, surfaces, and triple lines of intersection of surfaces that are based on the the idea of sequentially relaxing either individual nodes or edges in the mesh. Each relaxation is designed both to smooth the mesh and to conserve down to round-off error the area or volume enclosed by the curve or surface. For the case of smoothing surfaces and lines of intersection of surfaces, each relaxation consists of a pure smoothing component and a volume conserving correction which is chosen to be of minimum norm. Since surfaces and triple intersection lines can be conservatively smoothed, the algorithms are suitable for improving multimaterial grids used by physics simulations where exactly conserving the volume of each individual material may be a requirement or at least highly desirable. The algorithms are also suitable for smoothing piecewise linear functions of one or two variables while simultaneously preserving their integrals. We show examples of the application of the more powerful edge-based algorithms to curve, surface, and multimaterial volume grids and to a thin film simulation.

1. INTRODUCTION

Curves or surfaces obtained from physics-based simulations are frequently "jagged" or "non-smooth" and as such may be unsuitable as input for subsequent simulations. For example, Potts model simulations of metallic grain growth describe the interface between differing grains as a series of "stair-steps." The jagged stair-step interface is an artifact of the simulation and might produce incorrect results in subsequent simulations unless the interface is smoothed. Another example would be Lagrangian surface motion under

[1] This research is supported by the Department of Energy under contract W-7405-ENG-36.

0021-9991/01 $35.00

a computational fluid dynamics flow which could leave surfaces highly convoluted after several time-steps and unsuitable for further time-stepping unless they are smoothed.

By "smoothing" a surface grid, we mean (1) adjacent facets of the surface grid have normals adjusted to vary more gradually, (2) node densities are equidistributed on the surface, and (3) the aspect ratios of facets are improved.

A popular approach to surface grid smoothing has been to rely on a mapping from a parametric space to the surface and to smooth the grid in the parametric space [5, 10]. There are drawbacks to this approach. First, a mapping to a parametric space must exist, and often surfaces generated by physical simulations are unstructured and are not easily parameterizable. Second, smoothing of the surface grid in the parametric domain—while preserving the *shape* of the surface—does not necessarily preserve the volume that the surface grid encloses, due to the discreteness of the grid. This can be unacceptable in physical simulations. Also, exact preservation of surface shape is undesirable for "stair-step" surfaces.

Another approach to surface grid smoothing is evolution of the surface grid by mean curvature [2, 6, 8]. This approach will easily erase jaggedness of the surface, but does not conserve volume and requires a sophisticated PDE solver.

In this paper, we present a non-parametric, volume conserving approach to the smoothing of piecewise linear surface grids. Our approach will, for example, rapidly deform a nonsmooth closed surface into a smoothed surface which encloses the same volume down to round-off error. The surface grid can be unstructured and the resulting grid will satisfy the three conditions for smoothness presented above. To accomplish this, our volume conserving approach allows small deformations in the *shape* of the surface geometry. However, the degree of surface deformation can be limited by controlling the number of smoothing iterations performed. Because the algorithms are locally conservative, they are applicable to open as well as closed curves and surfaces. They could also be used to smooth piecewise linear functions in one or two variables with compact support while preserving their integral. (However, see note in Section 6.)

Many iterations of volume conserving smoothing will cause much change in the shape of the curve or surface. For example, multiple application of the smoother will turn a cube into a sphere which encloses the same volume. For most physical simulations, we wish to change the curve or surface shape only enough to allow the simulation to proceed. Thus it is most appropriate for the application designer who will utilize these volume conserving smoothing algorithms to make decisions on appropriate termination criteria. These criteria might include iterating until some mesh quality improvement measure is met or a maximum allowable surface deformation is reached. For example, one could measure the decrease in the ℓ_2 norm of the vector consisting of all the angles between the normals of adjacent surface facets (a vector of length equal to the number of edges in the surface) [3].

In Section 2 we develop area conservative smoothing of piecewise linear curves. In Section 3 we develop volume conservative smoothing of piecewise linear surfaces. In Section 4 we exhibit algorithms for smoothing of triple intersection lines of surfaces which conserve individually all volumes incident on the lines. In Section 5 we apply our algorithms to a thin film simulation. Finally in Section 6 we conclude by tying up some loose ends.

2. AREA CONSERVING SMOOTHING OF CURVE GRIDS

Consider a closed non-self-intersecting curve $\Gamma = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}, \mathbf{x}_n = \mathbf{x}_0)$ consisting of n line segments in $\mathbb{R}^2$. Say Γ encloses a region R (Fig. 1). We seek a smoothing operation

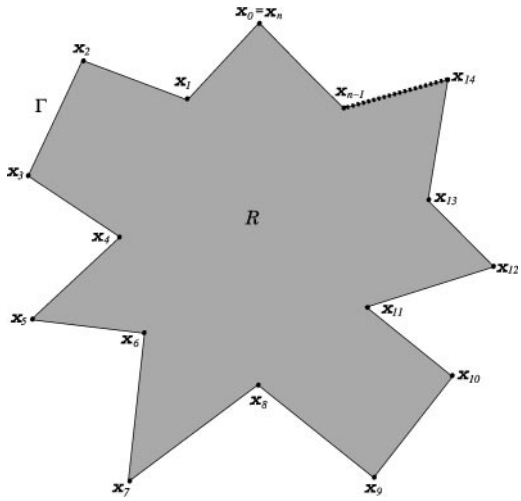

FIG. 1. Closed curve Γ enclosing region R.

on this curve that can be performed locally at each $\mathbf{x}_i$ that involves slightly altering the position of $\mathbf{x}_i$ based on nearby or adjacent data points (say $\{\mathbf{x}_{i-m}, \mathbf{x}_{i-m+1}, \ldots, \mathbf{x}_{i+m}\}$, m small). More generally, the smoothing operation could depend on points in $\{\mathbf{x}_{i-m}, \mathbf{x}_{i-m+1}, \ldots, \mathbf{x}_{i+m}\}$ and involve moving one or more points in this neighborhood. The smoothing operation should be chosen to not alter the area of R. If we perform the local smoothing operation in each local neighborhood in the curve in some order, this is called a *sweep*. We desire that only a small number of sweeps through the curve need be performed to smooth the overall appearance of the curve.

If now Γ is allowed to intersect itself, it is the *signed* area of R (i.e., with respect to the counter-clockwise orientation) we wish to conserve. Moreover, we can extend our ideas to an open curve $\Gamma = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}, \mathbf{x}_n \neq \mathbf{x}_0)$, by requiring that the sought-after smoothing operations conserve area in the closed curve $\Gamma = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{x}_0)$.

The simplest possible area conserving smoothing operation is depicted in Fig. 2. Here we consider the three points $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$ along Γ. By moving the central point $\mathbf{x}_1$ *parallel* to the line segment $\overline{\mathbf{x}_0\mathbf{x}_2}$, we are assured conservation of area. Further, by moving $\mathbf{x}_1$ such that the projection of $\mathbf{x}_1$ onto $\overline{\mathbf{x}_0\mathbf{x}_2}$ occurs midway between $\mathbf{x}_0$ and $\mathbf{x}_2$, we have achieved equal spacing of the segments $\overline{\mathbf{x}_0\mathbf{x}_1}$ and $\overline{\mathbf{x}_1\mathbf{x}_2}$ when projected onto the segment $\overline{\mathbf{x}_0\mathbf{x}_2}$.

We now formally state the algorithm based on this one-point smoothing operation. For a vector $\mathbf{v} = (x, y)$ in 2-D, we define $\mathbf{v}^\perp \equiv (-y, x)$. Let $\boldsymbol{A}_{021} = \frac{1}{2}(\mathbf{x}_2 - \mathbf{x}_0)^\perp \cdot (\mathbf{x}_1 - \mathbf{x}_0)$ be

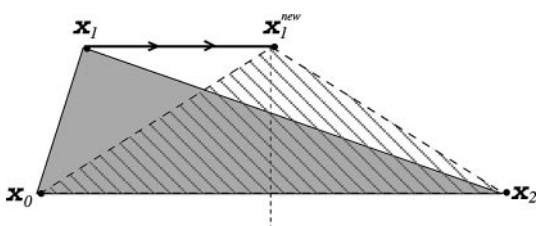

FIG. 2. One-point smoothing operation: Movement of $\mathbf{x}_1$ parallel to $\overline{\mathbf{x}_0\mathbf{x}_2}$ assures conservation of area under curve $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$.

the (signed) area of triangle $\Delta \mathbf{x}_0\mathbf{x}_2\mathbf{x}_1$. Then

$$h = \frac{2\boldsymbol{A}_{021}}{\|\mathbf{x}_2 - \mathbf{x}_0\|} \tag{1}$$

is the height of $\mathbf{x}_1$ above the baseline segment $\overline{\mathbf{x}_0\mathbf{x}_2}$. $\hat{\mathbf{n}} = \frac{(\mathbf{x}_2-\mathbf{x}_0)^\perp}{\|(\mathbf{x}_2-\mathbf{x}_0)^\perp\|}$ is the unit normal to the baseline $\overline{\mathbf{x}_0\mathbf{x}_2}$. Our smoothing operation thus involves repositioning $\mathbf{x}_1$ from its original position to

$$\mathbf{x}_1^{\text{new}} = \frac{1}{2}(\mathbf{x}_0 + \mathbf{x}_2) + h\hat{\mathbf{n}}. \tag{2}$$

Sweeping through the nodes in sequential order, we obtain the following algorithm:

ALGORITHM 1. Area conserving smoothing of a closed plane curve using single-node relaxations.

Repeat (sweep) until "done"

Do $i = 0, \ldots, n-1$

[Perform smoothing operation on neighborhood $\{\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}\}$
(i.e., relax node $\mathbf{x}_{i+1}$)]

$\hat{\mathbf{n}} \leftarrow \frac{(\mathbf{x}_{i+2}-\mathbf{x}_i)^\perp}{\|(\mathbf{x}_{i+2}-\mathbf{x}_i)^\perp\|}$

$A_{i,i+2,i+1} \leftarrow \frac{1}{2}(\mathbf{x}_{i+2} - \mathbf{x}_i)^\perp \cdot (\mathbf{x}_{i+1} - \mathbf{x}_i)$

$h \leftarrow 2\frac{\boldsymbol{A}_{i,i+2,i+1}}{\|(\mathbf{x}_{i+2}-\mathbf{x}_i)^\perp\|}$

$\mathbf{x}_{i+1} \leftarrow \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_{i+2}) + h\hat{\mathbf{n}}$.

(If Γ is not closed, Algorithm 1 is modified to not relax the endpoint nodes.) Algorithm 1, although simple, suffers from the following serious deficiency. Referring to Fig. 2, and calling the direction $\overrightarrow{\mathbf{x}_0\mathbf{x}_2}$ the direction "tangential" to Γ and the direction orthogonal to $\overrightarrow{\mathbf{x}_0\mathbf{x}_2}$ the "normal" direction, we see that the one-point smoothing operation smooths only in the the tangential direction. Any smoothing in the normal direction is forbidden by the conservation of area requirement. Because of this lack of normal smoothing, some star-shaped regions (Fig. 3) will not be affected by the operation. We conclude that it is necessary to design a local smoothing operation that includes normal smoothing.

Now consider four sequential points $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ along Γ. We take $\overrightarrow{\mathbf{x}_0\mathbf{x}_3}$ to be the direction tangential to the curve and the direction orthogonal to $\overrightarrow{\mathbf{x}_0\mathbf{x}_3}$ to be normal to the curve. If we simultaneously solved for the positions of $\mathbf{x}_1, \mathbf{x}_2$ subject to the constraint of area conservation, normal smoothing is possible. This is because conservation of area represents a single constraint in the normal direction, but there are two degrees of freedom available (the normal components of $\mathbf{x}_1$ and $\mathbf{x}_2$).

Thus, consider the following smoothing operation: Move $\mathbf{x}_1$, $\mathbf{x}_2$ so that the projection of $\mathbf{x}_1$ onto $\overline{\mathbf{x}_0\mathbf{x}_3}$ is one-third of the way between $\mathbf{x}_0$ and $\mathbf{x}_3$ and the projection of $\mathbf{x}_2$ is two-thirds of the way between $\mathbf{x}_0$ and $\mathbf{x}_3$. Furthermore, the distances of $\mathbf{x}_1$ and $\mathbf{x}_2$ away from $\overline{\mathbf{x}_0\mathbf{x}_3}$ are set to be equal and this distance (h in Fig. 4) is taken to conserve area. If this is done, smoothing occurs in the normal direction, as well as in the tangential direction.

The calculation of h is straightforward: The (signed) area A_{0321} of the quadrilateral $(\mathbf{x}_0, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1)$ cannot be altered. Repositioning of the points $\mathbf{x}_1, \mathbf{x}_2$ so that their projections

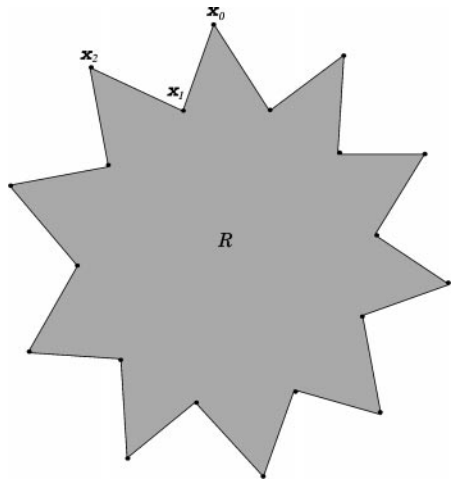

FIG. 3. Star-shaped region is invariant under (and hence not smoothed by) Algorithm 1.

are equally spaced implies that the area of the quadrilateral $(\mathbf{x}_0, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1)$ will be $\frac{2}{3}hl$, where l is the length of $\overline{\mathbf{x}_0\mathbf{x}_3}$. Thus we require

$$h = \frac{3}{2}\frac{A_{0123}}{l}.$$

The above smoothing operation can be interpreted as being a smoothing operation acting on the *edge* $\overline{\mathbf{x}_1\mathbf{x}_2}$. Thus, to perform a smoothing sweep through Γ using the above smoothing operation, we perform the operation on all the edges of Γ in some order. For example, if we use *sequential order*, we would perform the smoothing operation on the edge $\overline{\mathbf{x}_1\mathbf{x}_2}$, then perform it on the edge $\overline{\mathbf{x}_2\mathbf{x}_3}$ and continue until we have smoothed the last edge $\overline{\mathbf{x}_n\mathbf{x}_1}$.

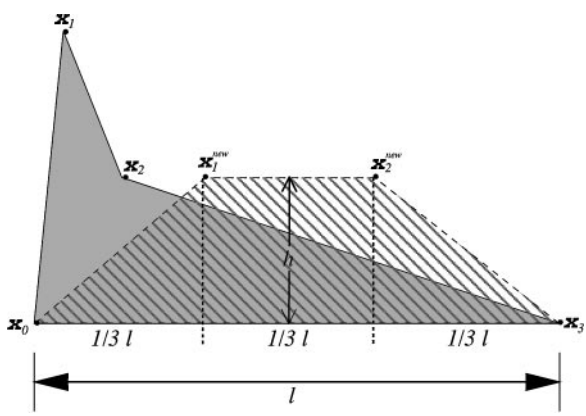

FIG. 4. Two-point (edge) smoothing. $\overline{\mathbf{x}_1\mathbf{x}_2}$ moved to be parallel to $\overline{\mathbf{x}_0\mathbf{x}_3}$ with projected endpoints at $\frac{1}{3}l$ and $\frac{2}{3}l$. Choosing $h = \frac{3}{2}\frac{A_{0123}}{l}$ conserves area A_{0123}.

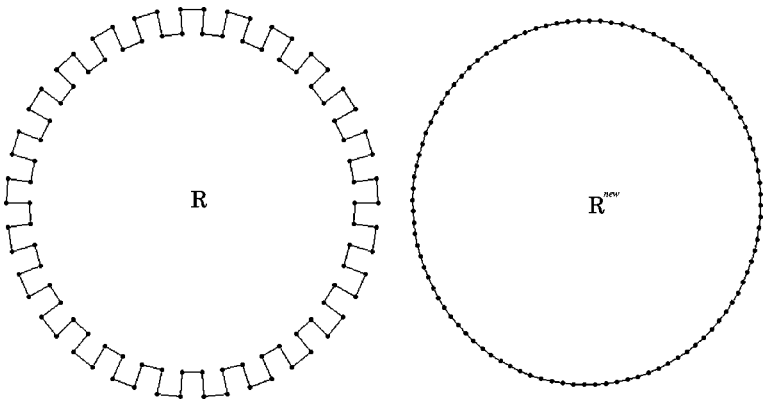

FIG. 5. Before and after smoothing of a closed stair-step curve using Algorithm 2 with 20 sweeps. Area of region R conserved down to round-off.

ALGORITHM 2. Area conserving smoothing of a closed plane curve using edge relaxations.

Repeat (sweep) until "done"

Do $i = 0, \ldots, n-1$

[Perform smoothing operation on neighborhood $\{\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}, \mathbf{x}_{i+3}\}$
(i.e., relax edge $\overline{\mathbf{x}_{i+1}, \mathbf{x}_{i+2}}$)]

$\hat{\mathbf{n}} \leftarrow \frac{(\mathbf{x}_{i+3} - \mathbf{x}_i)^\perp}{\|(\mathbf{x}_{i+3} - \mathbf{x}_i)^\perp\|}$

$A_{i,i+3,i+2,i+1} \leftarrow \frac{1}{2}(\mathbf{x}_{i+3} - \mathbf{x}_i)^\perp \cdot (\mathbf{x}_{i+2} - \mathbf{x}_i) + \frac{1}{2}(\mathbf{x}_{i+2} - \mathbf{x}_i)^\perp \cdot (\mathbf{x}_{i+1} - \mathbf{x}_i)$

[signed area of quad $(\mathbf{x}_i, \mathbf{x}_{i+3}, \mathbf{x}_{i+2}, \mathbf{x}_{i+1})$]

$h \leftarrow \frac{3}{2}\frac{A_{i,i+3,i+2,i+1}}{\|(\mathbf{x}_{i+3} - \mathbf{x}_i)^\perp\|}$

$\mathbf{x}_{i+1} \leftarrow \frac{2}{3}\mathbf{x}_i + \frac{1}{3}\mathbf{x}_{i+3} + h\hat{\mathbf{n}}$

$\mathbf{x}_{i+2} \leftarrow \frac{1}{3}\mathbf{x}_i + \frac{2}{3}\mathbf{x}_{i+3} + h\hat{\mathbf{n}}$.

In Fig. 5, we show the results of performing 20 sweeps on a closed curve. Area is conserved to within round-off error, and the curve is very smooth. Clearly, further iterations will not affect the appearance of the smoothed curve. In Fig. 6, we show the results of performing Algorithm 2 with 10 sweeps on an open curve Γ, holding the first and last points fixed. If Γ

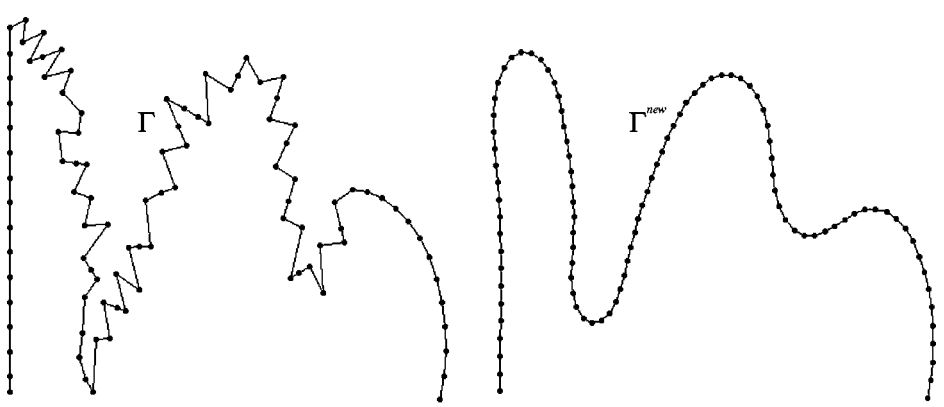

FIG. 6. Before and after smoothing of an open curve using Algorithm 2 with 10 sweeps.

were closed by addition of a segment between the first and last points, the area enclosed by Γ would be conserved down to round-off error. Further iterations will continue to deform the curve.

3. VOLUME CONSERVING SMOOTHING OF SURFACE GRIDS

Now consider a closed surface $S = \bigcup T_i$, where the T_i are planar triangular facets $T_i = T_{\mathbf{x}_{i_1}\mathbf{x}_{i_2}\mathbf{x}_{i_3}}$. We wish to perform a local smoothing operation in sweeps over small neighborhoods throughout the surface which will have the net effect of smoothing the surface without changing the amount of volume enclosed by the faceted surface. (If the surface is subdivided by other types of geometric facets—such as quadrilaterals—they can be subdivided into triangular facets for purposes of the following smoothing schemes.) More generally, if S is not closed we seek a local smoothing operation that does not alter the enclosed volume when S has been closed by some choice of additional triangles.

Similar to the previous section, we first consider the simple operation of altering the position of a single node at $\mathbf{x}$ based on data from its immediate neighbors. Consider Fig. 7, here the node at $\mathbf{x}$ is surrounded by the points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}$, which form a counterclockwise cycle when viewed from "outside" the surface. We define

$$\mathbf{e}^{(j)} = \mathbf{x}^{(j)} - \mathbf{x}.$$

Suppose we move the node at $\mathbf{x}$ to $\mathbf{x}^{\mathrm{s}} \equiv \mathbf{x} + d\mathbf{x}^{\mathrm{s}}$ through the action of a smoothing operation only depending on data in the immediately surrounding neighborhood. Since this motion will in general alter the volume enclosed by the surface, we restore the correct volume by further repositioning the central node by $h\hat{\mathbf{n}}$. That is, to ensure conservation of volume, we further move the central node by some multiple h of a prudently chosen direction. Thus, the node will undergo a total displacement from $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}$, where

$$d\mathbf{x} = d\mathbf{x}^{\mathrm{s}} + h\hat{\mathbf{n}}. \tag{3}$$

In fact, we can easily solve for the direction $\hat{\mathbf{n}}$ that minimizes the norm of the volume

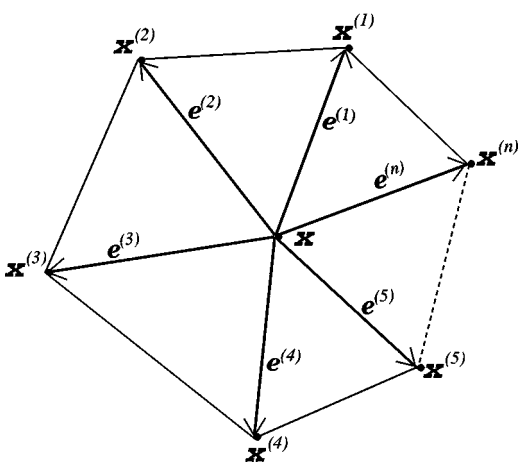

FIG. 7. Node at $\mathbf{x}$ on triangular faceted surface surrounded by n neighbors at $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$.

correction movement $\|h\hat{\mathbf{n}}\|$ and it will be a direction that could be reasonably considered to be "normal" to the undisturbed surface at $\mathbf{x}$.

Indeed, when the node is moved from $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}^s + h\hat{\mathbf{n}}$, the change of volume is given by

$$6\,dV = \sum_{j=1}^{n} (d\mathbf{x}^s + h\hat{\mathbf{n}}) \cdot \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)} = (d\mathbf{x}^s + h\hat{\mathbf{n}}) \cdot \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}. \tag{4}$$

(The "6" arises from use of the volume formula for tetrahedra employed for the terms in the sum in (4).) Thus $dV = 0$ implies

$$h = \frac{-d\mathbf{x}^s \cdot \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}}{\hat{\mathbf{n}} \cdot \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}}.$$

Thus if we choose

$$\hat{\mathbf{n}} = \frac{\sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}}{\left\| \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)} \right\|}, \tag{5}$$

then $\|h\hat{\mathbf{n}}\|$ will be minimized and our minimal volume corrective movement will be

$$h\hat{\mathbf{n}} = -(d\mathbf{x}^s \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}. \tag{6}$$

Note that $\hat{\mathbf{n}}$ could reasonably be considered to be the "normal" to the undisturbed surface at $\mathbf{x}$, since it is the normalized sum of area vectors of all triangles incident on $\mathbf{x}$.

It remains to specify the smoothing scheme that yields $\mathbf{x}^s$ based on nearest neighbor information. We choose Laplacian smoothing, defined by

$$\mathbf{x}^s = \mathbf{x} + d\mathbf{x}^s \equiv \frac{\sum_{j=1}^{n} \mathbf{x}^{(j)}}{n}.$$

With this choice, our smoothing scheme (3), (6) is entirely analogous to our simple smoothing scheme for curves (2). However, the correction (6) can be used with any smoothing scheme $\mathbf{x} \to \mathbf{x}^s$, and indeed smoothing schemes more sophisticated than Laplacian smoothing are available [4].

ALGORITHM 3. Volume conserving smoothing of a surface using single-node relaxations.

Repeat (sweep) until "done"

For each node $\mathbf{x}$ surrounded by neighbors $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$

$$\hat{\mathbf{n}} \leftarrow \frac{\sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}}{\left\| \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)} \right\|}$$

$$d\mathbf{x}^s \leftarrow \frac{\sum_{j=1}^{n} \mathbf{x}^{(j)}}{n} - \mathbf{x}$$

$$\mathbf{x} \leftarrow \mathbf{x} + d\mathbf{x}^s - (d\mathbf{x}^s \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}.$$

We give our simple volume corrected smoothing scheme with Laplacian smoothing in Algorithm 3. The weakness of Algorithm 3 is identical to that of the analogous scheme

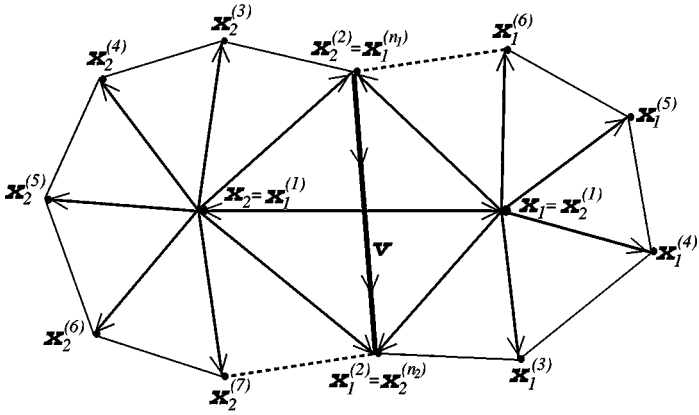

FIG. 8. Nomenclature for nodes surrounding edge $\overline{\mathbf{x}_1\mathbf{x}_2}$ on a triangular faceted surface.

(Algorithm 1) presented in the previous section. Both schemes are simple, but lack smoothing in the direction normal to the surface, since conservation of area or volume fully determines the normal distance of the relaxed node from the surface. As a consequence, Algorithm 3 will leave some star-shaped polyhedra (analogous to Fig. 3) unchanged.

Analogous to the development of the previous section, we develop a smoothing operation which exhibits smoothing normal to the surface, and which involves relaxing two adjacent neighbors—thus we relax *edges* on the surface. Consider Fig. 8; here we contemplate relaxing the edge $\overline{\mathbf{x}_1\mathbf{x}_2}$ based on data from the surrounding nodes.

Here $\mathbf{x}_1$ is surrounded by the nodes $\mathbf{x}_1^{(1)}, \mathbf{x}_1^{(2)}, \ldots, \mathbf{x}_1^{(n_1)}$, and $\mathbf{x}_2$ is surrounded by nodes $\mathbf{x}_2^{(1)}, \mathbf{x}_2^{(2)}, \ldots, \mathbf{x}_2^{(n_2)}$, such that

$$\mathbf{x}_2 = \mathbf{x}_1^{(1)} \quad \text{and} \quad \mathbf{x}_1 = \mathbf{x}_2^{(1)}.$$

We define $\mathbf{e}_i^{(j)} = \mathbf{x}_i^{(j)} - \mathbf{x}_i$. We also define

$$\mathbf{A}_i = \sum_{j=1}^{n_i} \mathbf{e}_i^{(j)} \times \mathbf{e}_i^{(j+1)}, \quad i = 1, 2. \tag{7}$$

We now contemplate moving $\mathbf{x}_i$ to $\mathbf{x}_i^{\mathrm{s}} \equiv \mathbf{x}_i + d\mathbf{x}_i^{\mathrm{s}}$ by some smoothing operation and then further shifting the two nodes by $h\hat{\mathbf{n}}$, which is a multiple h of an optimal direction $\hat{\mathbf{n}}$ chosen such that volume is conserved and $\|h\hat{\mathbf{n}}\|$ is minimal. We derive the optimal change $h\hat{\mathbf{n}}$ as follows. The two nodes undergo the total displacement

$$d\mathbf{x}_i \equiv d\mathbf{x}_i^{\mathrm{s}} + h\hat{\mathbf{n}}, \quad i = 1, 2. \tag{8}$$

The movement of the node at $\mathbf{x}_1$ to $\mathbf{x}_1 + d\mathbf{x}_1$ causes the triangles $\{(\mathbf{x}_1, \mathbf{x}_1^{(j)}, \mathbf{x}_1^{(j+1)}) \mid 1 \le j \le n_1\}$ to "sweep out" volume between their initial positions and their final positions at $\{(\mathbf{x}_1 + d\mathbf{x}_1, \mathbf{x}_1^{(j)}, \mathbf{x}_1^{(j+1)}) \mid 1 \le j \le n_1\}$. The volume change caused by motion of the node at $\mathbf{x}_1$ to $\mathbf{x}_1 + d\mathbf{x}_1$ is thus equal to the volume of the tetrahedra $\{(\mathbf{x}_1, \mathbf{x}_1 + d\mathbf{x}_1, \mathbf{x}_1^{(j)}, \mathbf{x}_1^{(j+1)}) \mid 1 \le j \le n_1\}$, or

$$6\, dV_1 = \sum_{j=1}^{n_1} d\mathbf{x}_1 \cdot \mathbf{e}_1^{(j)} \times \mathbf{e}_1^{(j+1)} = d\mathbf{x}_1 \cdot \mathbf{A}_1. \tag{9}$$

Next, the movement of the node at $\mathbf{x}_2$ to $\mathbf{x}_2 + d\mathbf{x}_2$ creates a volume change similar to (9), but we must take into account that the node at $\mathbf{x}_1 = \mathbf{x}_2^{(1)}$ has already been moved to $\mathbf{x}_1 + d\mathbf{x}_1$. That is, $\mathbf{e}_2^{(1)}$ has been changed to $\mathbf{e}_2^{(1)} + d\mathbf{x}_1$. Thus, defining

$$\widetilde{\mathbf{e}_2^{(1)}} = \mathbf{e}_2^{(1)} + d\mathbf{x}_1$$
$$\widetilde{\mathbf{e}_2^{(j)}} = \mathbf{e}_2^{(j)}, \quad 2 \le j \le n_2,$$

we have

$$\begin{aligned} 6\,dV_2 &= \sum_{j=1}^{n_2} d\mathbf{x}_2 \cdot \widetilde{\mathbf{e}_2^{(j)}} \times \widetilde{\mathbf{e}_2^{(j+1)}} \\ &= \sum_{j=1}^{n_2} d\mathbf{x}_2 \cdot \mathbf{e}_2^{(j)} \times \mathbf{e}_2^{(j+1)} + d\mathbf{x}_2 \cdot d\mathbf{x}_1 \times \mathbf{e}_2^{(2)} + d\mathbf{x}_2 \cdot \mathbf{e}_2^{(n_2)} \times d\mathbf{x}_1 \\ &= d\mathbf{x}_2 \cdot \mathbf{A}_2 + d\mathbf{x}_2 \cdot \left(\mathbf{e}_2^{(n_2)} - \mathbf{e}_2^{(2)}\right) \times d\mathbf{x}_1 \\ &= d\mathbf{x}_2 \cdot \mathbf{A}_2 + d\mathbf{x}_2 \cdot \mathbf{v} \times d\mathbf{x}_1, \end{aligned}$$

where

$$\mathbf{v} \equiv \mathbf{e}_2^{(n_2)} - \mathbf{e}_2^{(2)} = \mathbf{e}_1^{(2)} - \mathbf{e}_1^{(n_1)}. \tag{10}$$

Thus conservation of volume requires us to have

$$0 = 6\,dV = 6\,dV_1 + 6\,dV_2 = d\mathbf{x}_1 \cdot \mathbf{A}_1 + d\mathbf{x}_2 \cdot \mathbf{A}_2 + d\mathbf{x}_2 \cdot \mathbf{v} \times d\mathbf{x}_1. \tag{11}$$

Substituting (8) into (11) and solving for h yields

$$h = -\frac{d\mathbf{x}_1^{\mathrm{s}} \cdot \mathbf{A}_1 + d\mathbf{x}_2^{\mathrm{s}} \cdot \mathbf{A}_2 + d\mathbf{x}_2^{\mathrm{s}} \cdot \mathbf{v} \times d\mathbf{x}_1^{\mathrm{s}}}{\hat{\mathbf{n}} \cdot \left(\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{v} \times \left(d\mathbf{x}_1^{\mathrm{s}} - d\mathbf{x}_2^{\mathrm{s}}\right)\right)}. \tag{12}$$

Thus $\|h\hat{\mathbf{n}}\|$ is minimized if we choose

$$\hat{\mathbf{n}} = \frac{\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{v} \times \left(d\mathbf{x}_1^{\mathrm{s}} - d\mathbf{x}_2^{\mathrm{s}}\right)}{\left\|\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{v} \times \left(d\mathbf{x}_1^{\mathrm{s}} - d\mathbf{x}_2^{\mathrm{s}}\right)\right\|}. \tag{13}$$

That is, the distance the edge $\overline{\mathbf{x}_1\mathbf{x}_2}$ is translated to recover volume is minimal when we choose h and $\hat{\mathbf{n}}$ according to (12) and (l3).

It remains to specify the smoothing scheme that yields the $\mathbf{x}_i^{\mathrm{s}}$ based on nearest neighbor information. We choose simultaneous Laplacian smoothing of both $\mathbf{x}_i$. That is, we require

$$\mathbf{x}_1^{\mathrm{s}} = \frac{1}{n_1}\left(\mathbf{x}_2^{\mathrm{s}} + \sum_{j=2}^{n_1} \mathbf{x}_1^{(j)}\right) \tag{14}$$

and

$$\mathbf{x}_2^{\mathrm{s}} = \frac{1}{n_2}\left(\mathbf{x}_1^{\mathrm{s}} + \sum_{j=2}^{n_2} \mathbf{x}_2^{(j)}\right). \tag{15}$$

Substituting (15) into (14), we obtain

$$\mathbf{x}_1^s = \frac{1}{n_1 n_2 - 1} \sum_{j=2}^{n_2} \mathbf{x}_2^{(j)} + \frac{n_2}{n_1 n_2 - 1} \sum_{j=2}^{n_1} \mathbf{x}_1^{(j)}. \tag{16}$$

We can now compute $\mathbf{x}_1^s$, using (16), and then compute $\mathbf{x}_2^s$ by substituting the result into (15).

However, the correction given by (8), (12), and (l3) can be used with any smoothing scheme that modifies the edge $\overline{\mathbf{x}_1\mathbf{x}_2}$. For one of our test problems, we instead used underrelaxed Laplacian smoothing:

$$\mathbf{x}_i^s \leftarrow (1-\omega)\mathbf{x}_i + \omega \mathbf{x}_i^s, \quad i = 1, 2,$$

where the $\mathbf{x}_i^s$ on the right-hand side are the positions yielded by simultaneous Laplacian smoothing (14), (15) and the $\mathbf{x}_i^s$ on the left-hand side are the positions given by underrelaxed Laplacian smoothing with $0 < \omega \leq 1$. This allows smoothing to be slowed down in order to more finely control surface deformation from iteration to iteration. Algorithm 4 gives our volume conserving smoothing scheme with edge relaxation by underrelaxed simultaneous Laplacian smoothing.

ALGORITHM 4. Volume conserving smoothing of a surface using edge relaxations.

Repeat (sweep) until "done"

For each edge $\overline{\mathbf{x}_1\mathbf{x}_2}$ surrounded by neighbors $\{\mathbf{x}_i^{(j)}\}_{i=1,2}^{j=1,\ldots,n_i}$ (Fig. 8), relax edge:

$\mathbf{A}_i \leftarrow \sum_{j=1}^{n_i} \mathbf{e}_i^{(j)} \times \mathbf{e}_i^{(j+1)}, \quad i = 1, 2$

$\mathbf{v} \leftarrow \mathbf{e}_2^{(n_2)} - \mathbf{e}_2^{(2)}$

$\mathbf{x}_1^s \leftarrow \frac{1}{n_1 n_2 - 1}\left(\sum_{j=2}^{n_2} \mathbf{x}_2^{(j)} + n_2 \sum_{j=2}^{n_1} \mathbf{x}_1^{(j)}\right)$

$\mathbf{x}_2^s \leftarrow \frac{1}{n_2}\left(\mathbf{x}_1^s + \sum_{j=2}^{n_2} \mathbf{x}_2^{(j)}\right)$

$d\mathbf{x}_i^s \leftarrow \omega(\mathbf{x}_i^s - \mathbf{x}_i), \quad i = 1, 2, \quad 0 < \omega \leq 1$

$\mathbf{A} \leftarrow \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{v} \times (d\mathbf{x}_1^s - d\mathbf{x}_2^s)$

If ($\|\mathbf{A}\| >$ "a tiny number") then

$\hat{\mathbf{n}} \leftarrow \mathbf{A}/\|\mathbf{A}\|$

$h \leftarrow -(d\mathbf{x}_1^s \cdot \mathbf{A}_1 + d\mathbf{x}_2^s \cdot \mathbf{A}_2 + d\mathbf{x}_2^s \cdot \mathbf{v} \times d\mathbf{x}_1^s)/\|\mathbf{A}\|$

$\mathbf{x}_i \leftarrow \mathbf{x}_i + d\mathbf{x}_i^s + h\hat{\mathbf{n}}, \quad i = 1, 2.$

Figures 9 and 10 show results of smoothing grids with nonsmooth features using Algorithm 4. In Fig. 9, a cube is depicted after 0, 10, 100, and 1000 sweeps with $\omega = 1$. Volume is conserved throughout. Note that after only 10 sweeps the cube has been well smoothed at the 12 edges.

In Fig. 10, we initially randomly perturb the node positions of the lower half of a sphere and use Algorithm 4 for 1, 5, 10, 100, and 1000 sweeps. We use $\omega = 0.1$ to allow the algorithm to make changes more gradually. (If we had used $\omega = 1$, it would roughly take $1/10$ as many iterations to produce the results depicted.) In both these examples, the algorithm seems to be moving the grid toward a spherical shape even though the grid topology is nearly regular in Fig. 10 and highly irregular in Fig. 9.

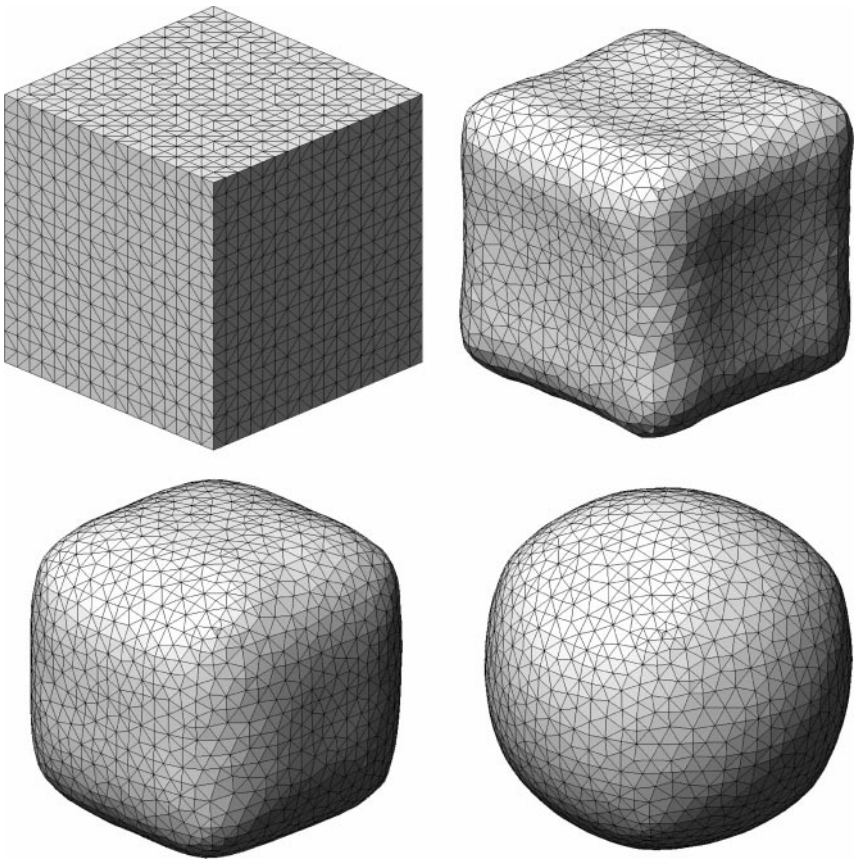

FIG. 9. Cube after 0, 10, 100, and 1000 sweeps over the surface using Algorithm 4 with $\omega = 1$.

4. VOLUME CONSERVING SMOOTHING OF TRIPLE LINES

In simulations involving several volume regions whose union comprises the full computational domain, it may be necessary to perform smoothing on the surfaces of all the volumes with the requirement that each individual volume is conserved. (We refer to each distinct region as a "material," and we are here concerned with the problem of smoothing the boundaries of all materials such that all material volumes are conserved.) Algorithms 3 and 4 are adequate for smoothing nodes that exist on surfaces separating two distinct materials or one material and the exterior of the domain but cannot be used for lines of multiple intersection where three or more materials intersect or two or more materials intersect with the external boundary.

We now generalize our schemes to allow smoothing of nodes along these intersection lines. Suppose three materials "1," "2," and "3" surround a line of multiple intersection (a "triple" line). For a node at $\mathbf{x}$ in the interior of the triple line, Algorithm 3 allows us to conserve material 1 (and thus the sum of the volumes of materials 2 and 3) by correcting smoothing of the node with respect to the boundary surface of material 1. This involves restricting motions of the node to a plane perpendicular to $\hat{\mathbf{n}}^{(1)}$ given by (5) with the $\mathbf{e}^{(j)}$ chosen to lie on the surface bounding material 1. Further, if we now consider the conservation

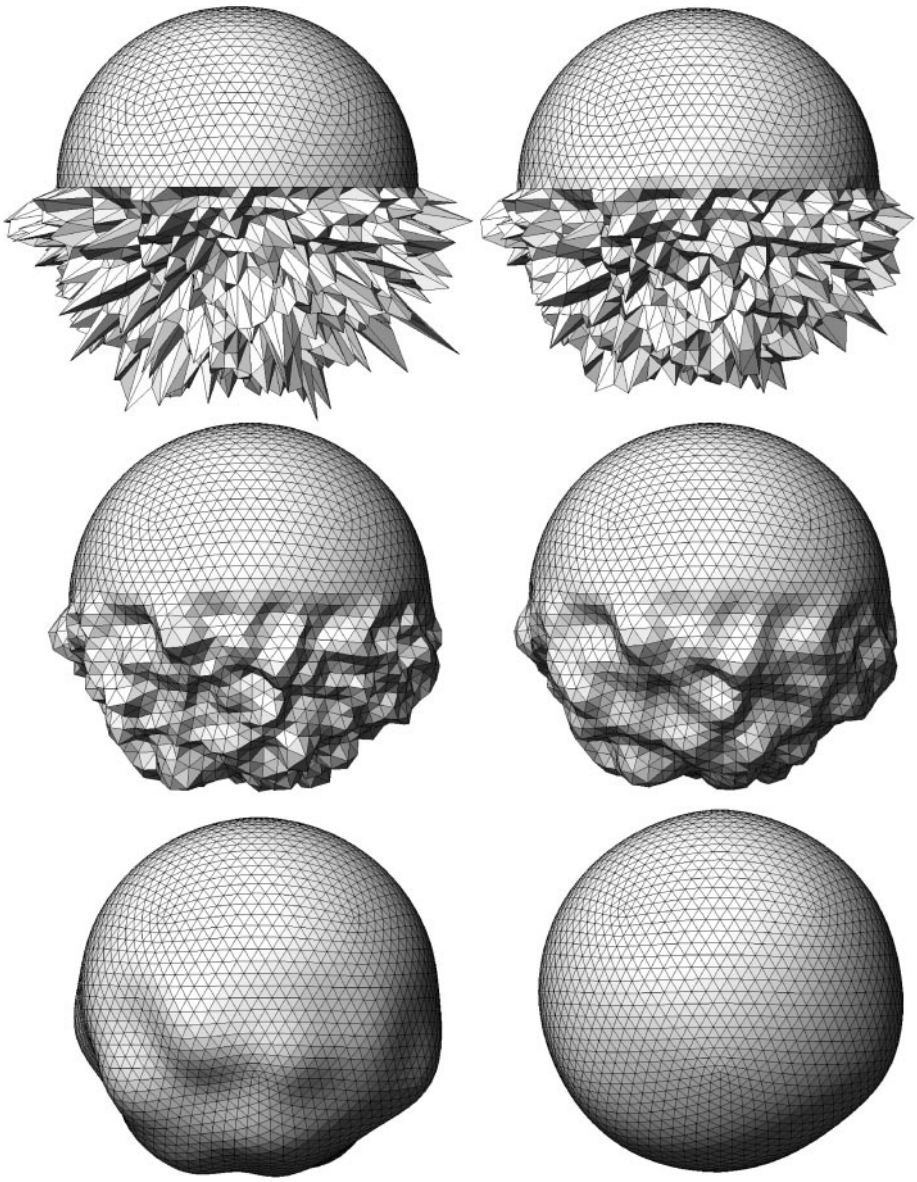

FIG. 10. "Noisy" sphere after 0, 1, 5, 10, 100, and 1000 sweeps over the surface using Algorithm 4 with $\omega = 0.1$.

of material 2 (versus the union of materials 1 and 3), we are forced to restrict motions of the node to a plane perpendicular to $\hat{\mathbf{n}}^{(2)}$ given by (5), with the $\mathbf{e}^{(j)}$ chosen to lie on the surface bounding material 2.

Thus if motion of the node is restricted to be in the line in the direction $\frac{\hat{\mathbf{n}}^{(1)} \times \hat{\mathbf{n}}^{(2)}}{\|\hat{\mathbf{n}}^{(1)} \times \hat{\mathbf{n}}^{(2)}\|}$, materials 1 and 2 (and hence material 3) are conserved. This yields Algorithm 5. In Algorithm 5, the smoothing scheme only uses data given by the positions of the preceding and following nodes on the triple line, and can be viewed as a combination of Algorithms 1 and 3.

ALGORITHM 5. Volume conserving smoothing of a triple line using single-node relaxations.

Repeat (sweep) until "done"

For each node $\mathbf{x}_i$ in the interior of the triple line preceded by $\mathbf{x}_{i-1}$ and succeeded by $\mathbf{x}_{i+1}$

$\hat{\mathbf{n}}^{(1)} \leftarrow \dfrac{\sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}}{\left\| \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)} \right\|}$ using edges from the surface of material "1"

$\hat{\mathbf{n}}^{(2)} \leftarrow \dfrac{\sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}}{\left\| \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)} \right\|}$ using edges from the surface of material "2"

$\texttt{norm} \leftarrow \|\hat{\mathbf{n}}^{(1)} \times \hat{\mathbf{n}}^{(2)}\|$

If ($\texttt{norm} >$ "a tiny number") then

$\mathbf{t} \leftarrow (\hat{\mathbf{n}}^{(1)} \times \hat{\mathbf{n}}^{(2)})/\texttt{norm}$

$d\mathbf{x}_i^{\mathrm{s}} \leftarrow \frac{1}{2}(\mathbf{x}_{i-1} + \mathbf{x}_{i+1}) - \mathbf{x}_i$

$d\mathbf{x}_i \leftarrow (d\mathbf{x}_i^{\mathrm{s}} \cdot \mathbf{t})\mathbf{t}$

$\mathbf{x}_i \leftarrow \mathbf{x}_i + d\mathbf{x}_i$.

From this analysis, we can see that, in the absence of some special restrictions, we cannot smooth single nodes on quadruple intersection lines, because all conditions of volume conservation would require that motion of the node be orthogonal to three vectors, which usually implies zero motion.

Our final algorithm involves smoothing edges on a triple intersection line between materials 1, 2, and 3. As in the previous cases where we compared edge relaxation to single node relaxation, relaxing edges is preferable to relaxing single nodes due to the possibility of smoothing action orthogonal to the triple line.

If we conserve volumes of materials 1 and 2, volume of material 3 will automatically be conserved. Thus looking at (11), we write

$$0 = 6\,dV^{(\alpha)} = d\mathbf{x}_1 \cdot \mathbf{A}_1^{(\alpha)} + d\mathbf{x}_2 \cdot \mathbf{A}_2^{(\alpha)} + d\mathbf{x}_2 \cdot \mathbf{v}^{(\alpha)} \times d\mathbf{x}_1, \quad \alpha = 1, 2. \tag{17}$$

Here the meaning of (17) is identical to (11), except here the parameter α refers to the particular material or material surface. Thus for $\alpha = 1$, $\mathbf{A}_i^{(1)}$ and $\mathbf{v}^{(1)}$ are computed using (7) and (10) with the $\mathbf{e}^{(j)}$ chosen to lie on the surface bounding material 1. If $\alpha = 2$, quantities are computed using edges lying on the surface bounding material 2. The $d\mathbf{x}_i$ are the displacements of the endpoints of the triple line edge being relaxed. The displacements $d\mathbf{x}_i$ are assumed to be of the form

$$d\mathbf{x}_i = d\mathbf{x}_i^{\mathrm{s}} + \mathbf{c}, \quad i = 1, 2. \tag{18}$$

Here the $d\mathbf{x}_i^{\mathrm{s}}$ are displacements due to a smoothing operation, and $\mathbf{c}$ is a rigid displacement of the whole edge designed to restore the volumes of materials 1 and 2. For given $d\mathbf{x}_i^{\mathrm{s}}$, we will determine the $\mathbf{c}$ of least norm.

Indeed, substituting (18) into (17), we obtain

$$\mathbf{c} \cdot \mathbf{A}^{(\alpha)} = g^{(\alpha)}, \quad \alpha = 1, 2, \tag{19}$$

where

$$\mathbf{A}^{(\alpha)} \equiv \mathbf{A}_1^{(\alpha)} + \mathbf{A}_2^{(\alpha)} + \mathbf{v}^{(\alpha)} \times \left(d\mathbf{x}_1^{\mathrm{s}} - d\mathbf{x}_2^{\mathrm{s}}\right)$$

and

$$g^{(\alpha)} \equiv -d\mathbf{x}_1^{\mathrm{s}} \cdot \mathbf{A}_1^{(\alpha)} - d\mathbf{x}_2^{\mathrm{s}} \cdot \mathbf{A}_2^{(\alpha)} - d\mathbf{x}_2^{\mathrm{s}} \cdot \mathbf{v}^{(\alpha)} \times d\mathbf{x}_1^{\mathrm{s}}, \quad \alpha = 1, 2.$$

From this we can see that for $\mathbf{c}$ to be of minimum norm, it must be of the form

$$\mathbf{c} = h^{(1)}\mathbf{A}^{(1)} + h^{(2)}\mathbf{A}^{(2)}. \tag{20}$$

(Otherwise, suppose $\mathbf{c} = h^{(1)}\mathbf{A}^{(1)} + h^{(2)}\mathbf{A}^{(2)} + \mathbf{d}$ satisfies (19), for some nonzero $\mathbf{d} \perp \text{span}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)})$. Then $\mathbf{c}' \equiv h^{(1)}\mathbf{A}^{(1)} + h^{(2)}\mathbf{A}^{(2)}$ satisfies (19) with $\|\mathbf{c}'\| < \|\mathbf{c}\|$.) Therefore, assuming the form (20), (19) yields the system

$$\begin{bmatrix} \mathbf{A}^{(1)} \cdot \mathbf{A}^{(1)} & \mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)} \\ \mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)} & \mathbf{A}^{(2)} \cdot \mathbf{A}^{(2)} \end{bmatrix} \begin{bmatrix} h^{(1)} \\ h^{(2)} \end{bmatrix} = \begin{bmatrix} g^{(1)} \\ g^{(2)} \end{bmatrix}. \tag{21}$$

If $\mathbf{A}^{(1)}$ is not parallel to $\mathbf{A}^{(2)}$, the matrix on the left-hand side of (21) is symmetric positive definite and hence invertible. In this case, the solution is given by

$$h^{(1)} = \frac{1}{\|\mathbf{A}^{(1)}\|^2\|\mathbf{A}^{(2)}\|^2 - (\mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)})^2} \left(\|\mathbf{A}^{(2)}\|^2 g^{(1)} - \mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)} g^{(2)}\right) \tag{22}$$

$$h^{(2)} = \frac{1}{\|\mathbf{A}^{(1)}\|^2\|\mathbf{A}^{(2)}\|^2 - (\mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)})^2} \left(-\mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)} g^{(1)} + \|\mathbf{A}^{(1)}\|^2 g^{(2)}\right). \tag{23}$$

ALGORITHM 6. Volume conserving smoothing of a triple line using edge relaxations.

Repeat (sweep) until "done"
  For each edge $\overline{\mathbf{x}_1\mathbf{x}_2}$ in the interior of the triple line preceded by $\mathbf{x}_0$ and succeeded by $\mathbf{x}_3$
    $\mathbf{A}_i^{(1)} \leftarrow \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}$ using edges from the surface of material "1," $i = 1, 2$
    $\mathbf{v}^{(1)} \leftarrow \mathbf{e}^{(n_2)} - \mathbf{e}^{(2)}$ using edges from the surface of material "1"
    $\mathbf{A}_i^{(2)} \leftarrow \sum_{j=1}^{n} \mathbf{e}^{(j)} \times \mathbf{e}^{(j+1)}$ using edges from the surface of material "2," $i = 1, 2$
    $\mathbf{v}^{(2)} \leftarrow \mathbf{e}^{(n_2)} - \mathbf{e}^{(2)}$ using edges from the surface of material "2"
    $d\mathbf{x}_1^{\mathrm{s}} \leftarrow (\frac{2}{3}\mathbf{x}_0 + \frac{1}{3}\mathbf{x}_3) - \mathbf{x}_1$
    $d\mathbf{x}_2^{\mathrm{s}} \leftarrow (\frac{1}{3}\mathbf{x}_0 + \frac{2}{3}\mathbf{x}_3) - \mathbf{x}_2$
    $\mathbf{A}^{(\alpha)} \leftarrow \mathbf{A}_1^{(\alpha)} + \mathbf{A}_2^{(\alpha)} + \mathbf{v}^{(\alpha)} \times (d\mathbf{x}_1^{\mathrm{s}} - d\mathbf{x}_2^{\mathrm{s}}), \quad \alpha = 1, 2$
    $\mathtt{det} \leftarrow \|\mathbf{A}^{(1)}\|^2\|\mathbf{A}^{(2)}\|^2 - (\mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)})^2$
    If ($\mathtt{det} >$ "a tiny number") then
      $g^{(\alpha)} \leftarrow -d\mathbf{x}_1^{\mathrm{s}} \cdot \mathbf{A}_1^{(\alpha)} - d\mathbf{x}_2^{\mathrm{s}} \cdot \mathbf{A}_2^{(\alpha)} - d\mathbf{x}_2^{\mathrm{s}} \cdot \mathbf{v}^{(\alpha)} \times d\mathbf{x}_1^{\mathrm{s}}, \quad \alpha = 1, 2$
      $h^{(1)} \leftarrow (\|\mathbf{A}^{(2)}\|^2 g^{(1)} - \mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)} g^{(2)})/\,\mathtt{det}$
      $h^{(2)} \leftarrow (-\mathbf{A}^{(1)} \cdot \mathbf{A}^{(2)} g^{(1)} + \|\mathbf{A}^{(1)}\|^2 g^{(2)})/\,\mathtt{det}$
      $\mathbf{x}_i \leftarrow \mathbf{x}_i + d\mathbf{x}_i^{\mathrm{s}} + h^{(1)}\mathbf{A}^{(1)} + h^{(2)}\mathbf{A}^{(2)}, \quad i = 1, 2.$

In Algorithm 6, the smoothing scheme only uses data given by the positions of the preceding and following nodes on the triple line, and hence can be viewed as a combination of Algorithms 2 and 4.

In Fig. 11, we show the action of Algorithm 6 on a multimaterial mesh consisting of four materials whose surface nodes have been perturbed to create a nonsmooth initial state. Interior edges on the surfaces were subjected to 20 sweeps of Algorithm 4 with $\omega = 1$, and interior edges on the triple lines were subjected to 20 sweeps with Algorithm 6. Note that there are two triple lines running through the interior of the figure and the other "triple lines"

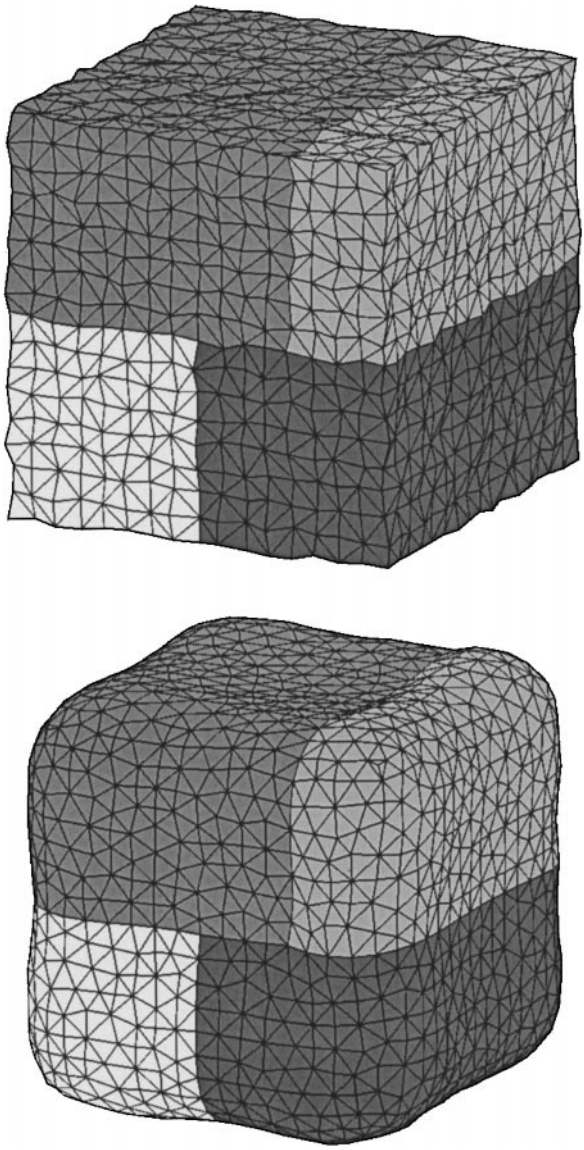

FIG. 11. "Noisy" four-material grid before and after 20 sweeps over all surfaces and triple lines using Algorithm 6 and Algorithm 4 with $\omega = 1$.

run on the surface at the intersection of two materials and the exterior boundary. For the external triple lines, one can consider the "third material" conserved to be the complement of all the materials—the "outside." In the final grid, the surfaces are smooth, the volumes of all four materials are preserved, and the triple lines have been smoothed out as desired.

5. APPLICATION TO THIN FILM EVOLUTION

To illustrate the application of volume conserving smoothing, we will show how it maintains mesh quality during simulation of the evolution of a thin film. TopoSim3D [9] is a software simulator that models the chemical and physical processes involved in the

formation of thin films. It models (a) the transport of materials from the vapor phase of a reactor to the growth surface, (b) the reactive chemistry that occurs at the vapor/surface interface, and (c) the motion of the vapor/surface interface as material is deposited or removed by the chemistry. Processes (a) and (b) provide an estimate of the rate at which

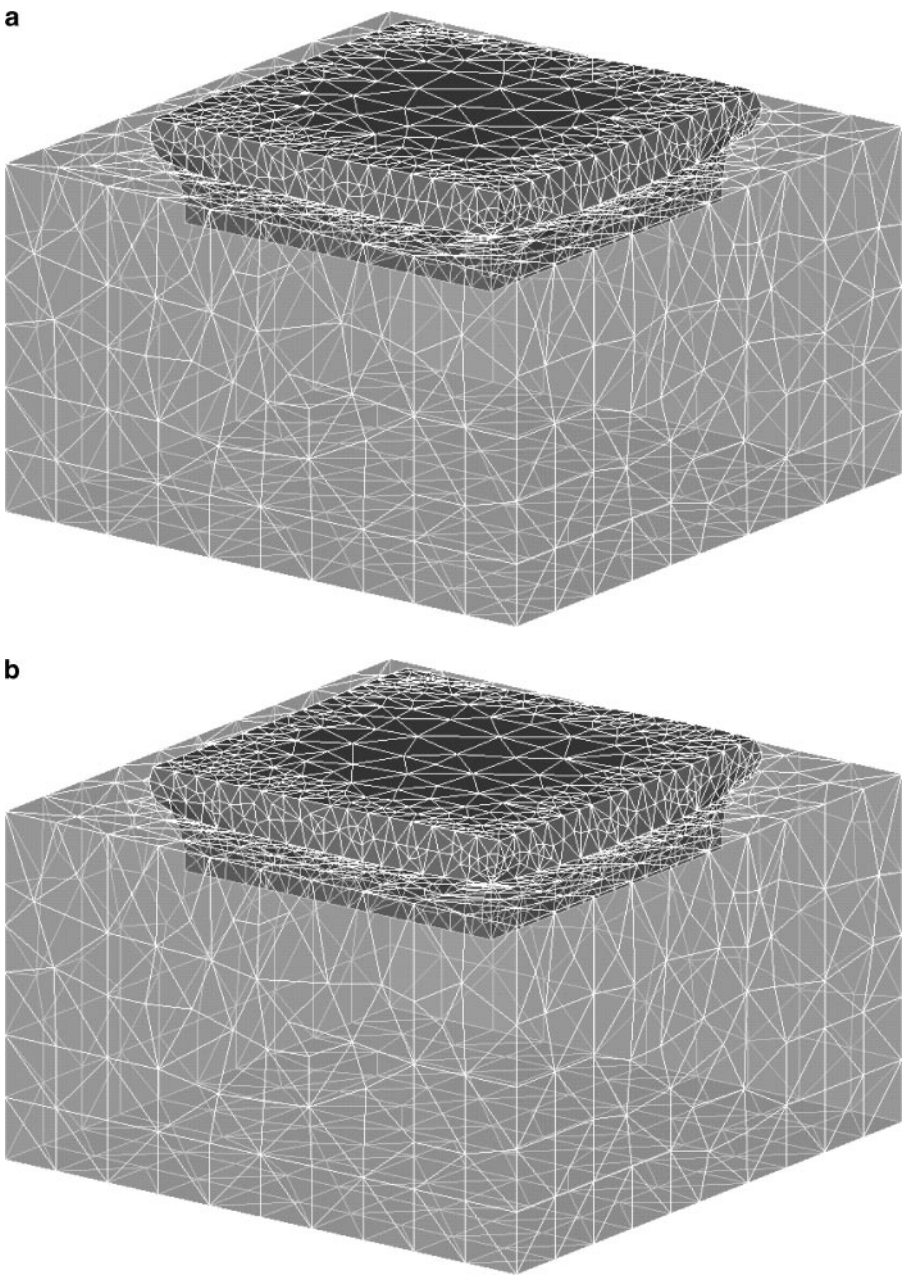

FIG. 12. (a)–(c). Thin film simulations with and without volume conserving smoothing. (a) Surfaces at time $t = 0.8$ s before three iterations of volume conserving smoothing. (b) Surfaces at time $t = 0.8$ s after three iterations of volume conserving smoothing—only very subtle changes. (c) Simulation without volume conserving smoothing at time of failure, $t = 0.504$ s.

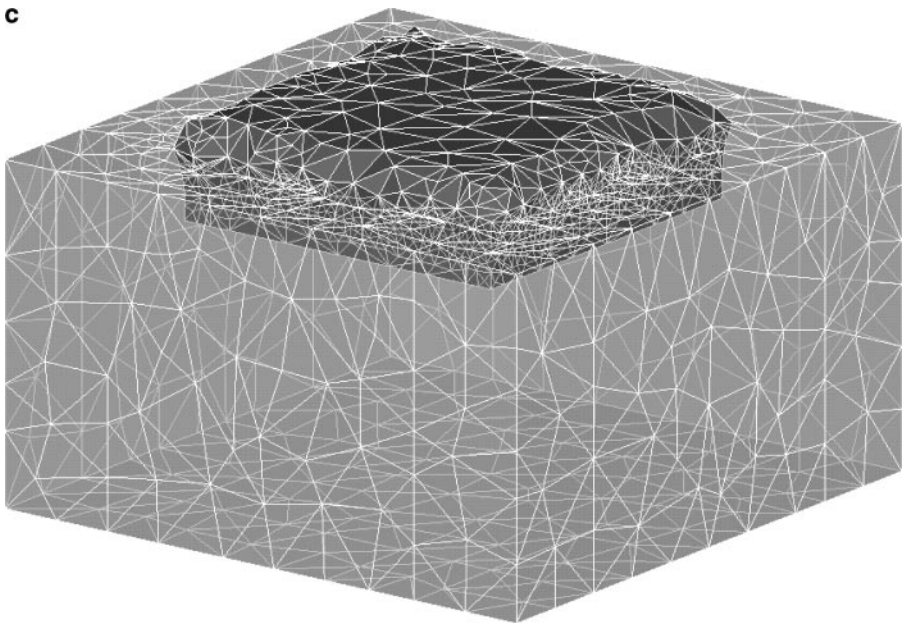

FIG. 12—*Continued*

the surface will grow at any point on the surface. When the mesh nodes that lie on the vapor/surface interface are moved, the mesh distorts. Unless mesh maintenance operations are performed, the distortion will accumulate such that any additional surface motion will cause a mesh element to invert which will, in turn, cause the simulation to halt. Because of the nature of the simulation, mesh maintenance must preserve the volume of the material deposited or removed. Volume conserving smoothing of surface nodes is one part of the mesh maintenance procedure; other parts include volume mesh smoothing (repositioning of nonsurface nodes—see Section 6), edge refinement, and grid connectivity changes.

In this simulation, we are depositing a material onto a substrate which has been partially masked with a second material. There is a square-shaped hole in the mask, and the unmasked surface beneath the hole has a much higher sticking coefficient than the surrounding masked substrate. Hence the major accumulation occurs in the central square region of the surface. In this simulation, the surface motion is known to be smooth, but the method of motion introduces unevenness and our volume conserving smoothing will recover the smooth surface motion. To illustrate the beneficial effects of volume conserving smoothing, we ran the simulation with and without the volume conserving smoothing step. In the simulation with smoothing, the mesh was smoothed for three iterations every 0.1 s of simulation time using Algorithm 4 with $\omega = 1$ and Algorithm 6 on the two sets of triple lines at and above the substrate. Figure 12a shows the surface before the volume conserving smoothing step at $t = 0.8$ s; view Fig. 12b after the smoothing step. As can be seen, very little change is made to the surface by the volume conserving smoothing at any given time-step. Although at each maintenance step the corrections to the mesh are minor, the cumulative effect is major. Figure 12c shows the surface after $t = 0.504$ s in a simulation where no volume conserving smoothing was performed. At this point the mesh is so distorted that the simulation cannot

proceed without inverting an element. We chose to perform volume conserving smoothing for three iterations every 0.1 s, simply because we found this was the smallest amount of smoothing that improved the mesh sufficiently for the simulation to complete successfully.

6. ADDITIONAL CONSIDERATIONS

Use of single node algorithms. In this paper we have presented three single node smoothing algorithms (1, 3, and 5) and three edge smoothing algorithms (2, 4, and 6) that perform volume conservative smoothing for curves, surfaces, and triple intersection lines. Because of the existence of nonsmooth grids that are unaffected by the single node algorithms, we advocate use of the edge relaxation algorithms. It must be noted that there are cases where only the single node algorithms can be used. For instance, if there is a node i in the interior of a surface and all neighbors of that node are on the boundary, then there is no "interior" edge that contains i that can be relaxed by Algorithm 4 in order to alter the position of i. Use of Algorithm 3 in this case, rather than simply leaving i untouched might be preferable. The same consideration applies to the middle node of a triple line consisting of only two segments. Algorithm 6 cannot be used to alter the position of this middle node, and so use of Algorithm 5 rather than leaving the node untouched might be preferable. In our examples, we did not do this extra coding and indeed it was unnecessary to do so since these problems only occur with extremely coarse grids.

Topological anomalies. In the case of edge relaxations on triple lines, it is assumed that each of materials 1, 2, and 3 are incident on the edge $\overline{\mathbf{x}_1\mathbf{x}_2}$ *as a single wedge*. If, e.g. material 1 is incident on $\overline{\mathbf{x}_1\mathbf{x}_2}$ as two separate wedges (i.e., the surface bounding material 1 intersects itself at $\overline{\mathbf{x}_1\mathbf{x}_2}$), the derivation of volume conservation for Algorithm 6 *does not apply* and volume will not be conserved. It is important, when coding this algorithm, to detect these (rare) cases and refrain from relaxing these kinds of edges. Alternatively, one could perform an *a posteriori* check of volumes to verify conservation and reject node movements that result in volume changes. (Of course, the volume check must only involve triangles *local* to the area—i.e., detect possible volume *change* rather than recalculate the whole volume every time an edge is relaxed.)

Quadruple lines. One could probably devise a scheme that could conserve all volumes incident on a quadruple line using edge smoothing, provided that corrective edge motions more general than the rigid translation (18) are considered. We do not pursue this here, and thus we leave any quadruple lines untouched.

Triangle collapse. Whenever using Laplacian smoothing on unstructured grids (or virtually any other kind of smoothing scheme on unstructured grids), there is the possibility that nodes are ejected from the polygon formed by their first neighbors and hence that triangles are inverted. Thus it is prudent to always check the orientation or quality of triangles after smoothing and reject disastrous moves (i.e., use "guards"). (In fact, we did not use any guards in the sample runs in this paper.)

Smoothing functions of one or two variables. If the "curves" or "surfaces" we are considering are the graphs of piecewise linear functions of one or two variables, we can use Algorithms 1–4 to smooth the positions of the interior nodes while preserving the integrals of the functions. However, one must reject any node or edge relaxation that would cause

the graphs to become multi-valued. (This could occur when relaxing near steep portions of the graphs.)

Volume mesh smoothing. If the goal of the user is to smooth surface meshes in a volume conserving fashion, then the algorithms in this paper suffice. If, however, there are volume elements conformally attached to the surface elements (e.g., triangles interior to a closed piecewise linear curve or tetrahedra interior to a closed piecewise linear surface), then it is possible to invert the volume elements when smoothing the surface elements. In this case one can smooth the volume elements by repositioning "volume" nodes (nodes not on, but interior to the enclosing surface) in tandem with smoothing of the surface nodes—and this smoothing will avoid inversion of volume elements. In [1] and [7] smoothing of volume elements is done by minimizing a functional which becomes infinite if volume elements invert, and thus moving volume nodes by requiring minimization of the functional usually avoids inversion of elements. Nevertheless, extreme deformation of surfaces and/or lack of volume nodes to move can in principle lead to situations where changing of grid connectivity might be the only way to avoid volume element inversion.

REFERENCES

1. Randolph E. Bank and R. Kent Smith, Mesh smoothing using a posteriori error estimates, *SIAM J. Numer. Anal.* **34**, No. 3, 979 (1997).
2. N. Carlson and K. Miller, Design and application of a gradient-weighted moving finite element code II: In two dimensions, *SIAM J. Sci. Comput.* **19**, 766 (1998).
3. N. Dyn, D. Levin, and S. Rippa, Data dependent triangulations for piecewise linear interpolation *IMA J. Numer. Anal.* **10**, 137 (1990).
4. A. Khamayseh and A. Kuprat, Anisotropic smoothing and solution adaption for unstructured grids, *Int. J. Numer. Meth. Eng.* **39**, 3163 (1996).
5. A. Khamayseh and C. W. Mastin, Computational conformal mapping for surface grid generation, *J. Comput. Phys.* **123**, 394 (1996).
6. A. Kuprat, Modeling microstructure evolution using gradient-weighted moving finite elements, *SIAM J. Sci. Comput.* **22**, No. 2 (2000).
7. Andrew Kuprat, Denise George, Eldon Linnebur, Harold Trease, and R. Kent Smith, Moving adaptive unstructured 3-D meshes in semiconductor process modeling applications, *VLSI Des.* **6**, 373 (1998).
8. K. Miller, A geometrical-mechanical interpretation of gradient-weighted moving finite elements, *SIAM J. Numer. Anal.* **34**, 67 (1997).
9. R. B. Walker, J. D. Kress, D. E. Hanson, A. F. Voter, J. T. Gammel, M. E. Coltrin, and P. Ho, *Modeling the Growth of Thin Films in Complex 3D Geometrical Structures*, Los Alamos National Laboratory Report LA-UR-00-1 (2000). [part]
10. Z. U. A. Warsi, Numerical grid generation in arbitrary surfaces through a second-order differential geometric model, *J. Comput. Phys.* **64**, 82 (1986).