LA-UR-98-3381

Non-Parametric Volume Conserving Smoothing

Andrew Kuprat and Ahmed Khamayseh
Los Alamos National Laboratory
Los Alamos, New Mexico 87545, U.S.A.

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 will produce incorrect results in subsequent simulations unless the interface is smoothed. Another example would be Lagrangian surface motion under 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 [3,6]. 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 [1,4,5]. This approach will easily erase "stair-steps", 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 surface grids. Our approach will, for example, rapidly deform a "stair-stepped" 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.

Additionally, the notion of volume conservation is generalized in a natural way to allow us to extend our scheme to non-closed surface grids. Finally, our conservative smoothing schemes are naturally applicable to the lower-dimensional case of area conservation for open and closed curves in the plane. Examples of conservative smoothing of open and closed curves and surfaces will be presented.

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 $I\!R^2$. Say Γ encloses a region R (Figure 1). We seek a smoothing operation 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.

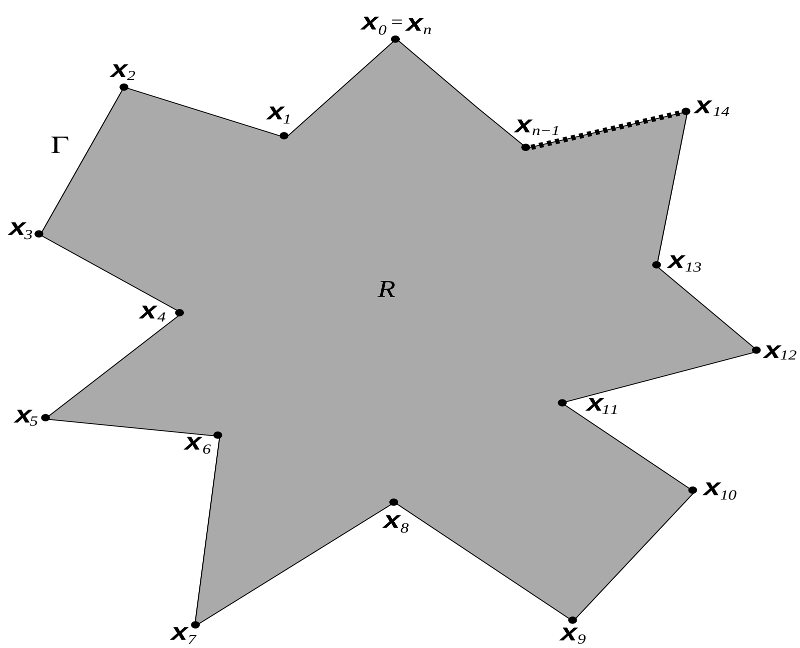

Figure 1. *Closed curve* Γ *enclosing region* R.

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 Figure 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 $\overline{\mathbf{x}_0\mathbf{x}_2}$.

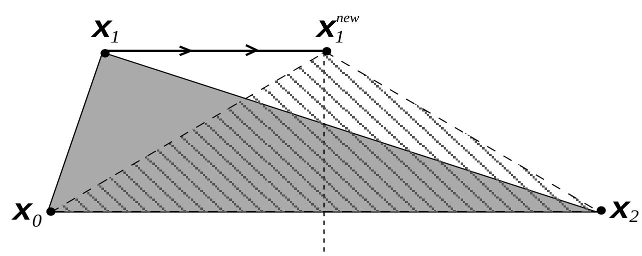

Figure 2. *One-point smoothing operation:*

Movement of $\mathbf{x}_1$ *parallel to* $\overline{\mathbf{x}_0\mathbf{x}_2}$ *assures conservation of area under curve* $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$.

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 $A_{021} = \frac{1}{2}(\mathbf{x}_2 - \mathbf{x}_0)^\perp \cdot (\mathbf{x}_1 - \mathbf{x}_0)$ be the (signed) area of triangle $\Delta\mathbf{x}_0\mathbf{x}_2\mathbf{x}_1$. Then

$$h = \frac{2A_{021}}{||\mathbf{x}_2 - \mathbf{x}_0||}$$

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

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

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

Algorithm 1: Area-Conserving smoothing of a 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 $\mathbf{x}_{i+1}$)]

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

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

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

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

Algorithm 1, although simple, suffers from the following serious deficiency. Referring to Figure 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 (Figure 3) will not be affected by the operation. We conclude it is necessary to design a local smoothing operation that includes normal smoothing.

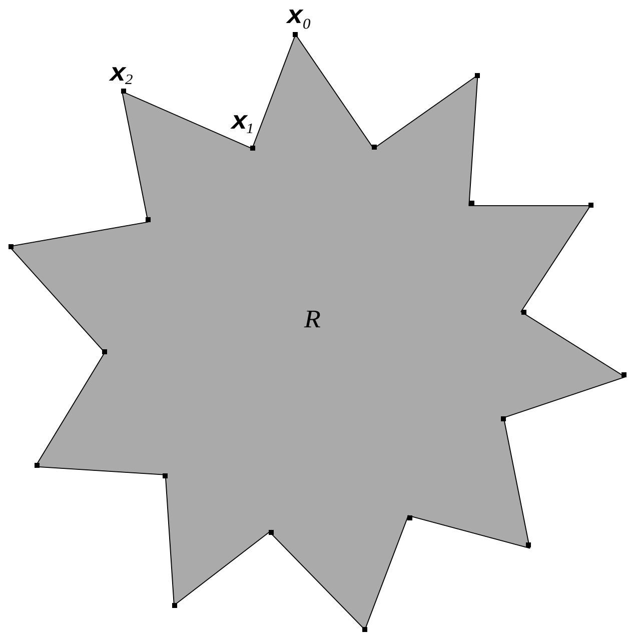

Figure 3. *Star-shaped region is invariant under (and hence not smoothed by) Algorithm 1.*

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 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 Figure 4) is taken to conserve area. If this is done, smoothing occurs in the normal direction, as well as in the tangential direction.

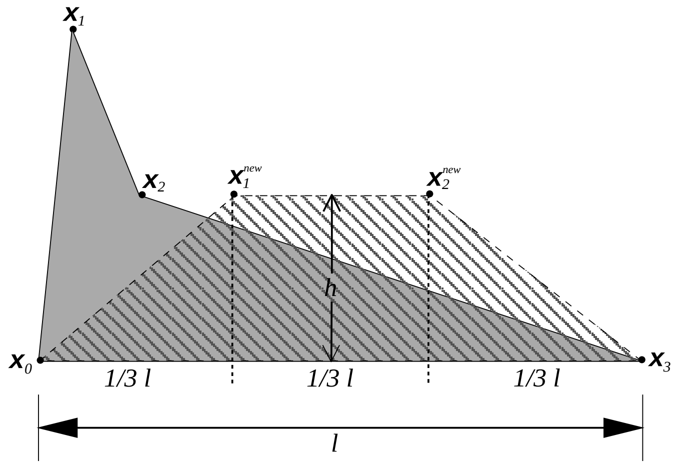

Figure 4. *Two point (edge) smoothing.* $\overline{\mathbf{x}_1\mathbf{x}_2}$ *moved to be parallel to* $\overline{\mathbf{x}_0\mathbf{x}_3}$ *with projected endpoints at* $\frac{1}{3}l$ and $\frac{2}{3}l$. Choosing $h = \frac{3}{2}\frac{A_{0123}}{l}$ conserves area A_{0123}.

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 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 had smoothed the last edge $\overline{\mathbf{x}_n\mathbf{x}_1}$.

Algorithm 2: Area-conserving smoothing of 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 $(\mathbf{x}_i, \mathbf{x}_{i+3}, \mathbf{x}_{i+2}, \mathbf{x}_{i+1})$]

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

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

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

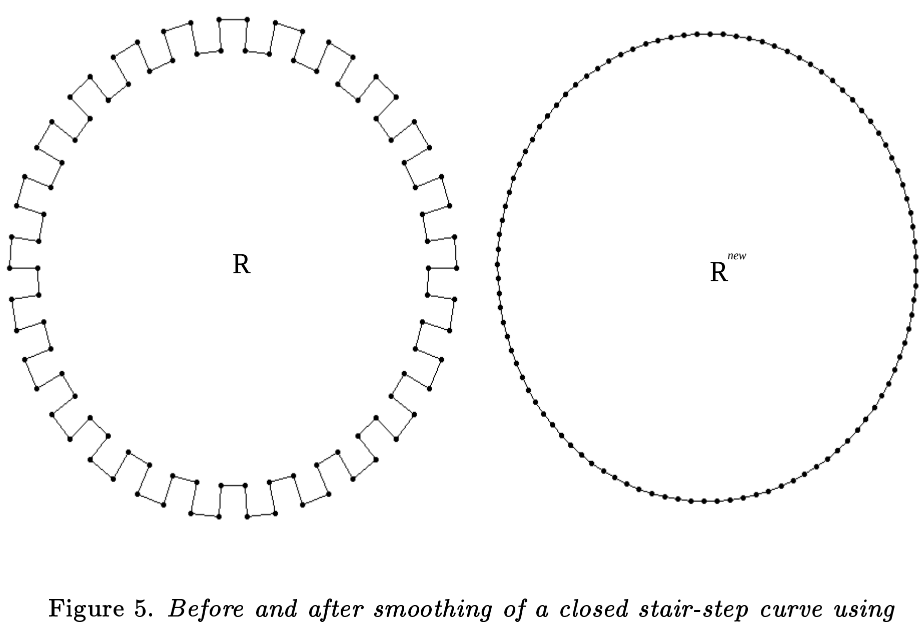

Figure 5. *Before and after smoothing of a closed stair-step curve using* Algorithm 2 with 20 sweeps. Area of region R conserved down to round-off.

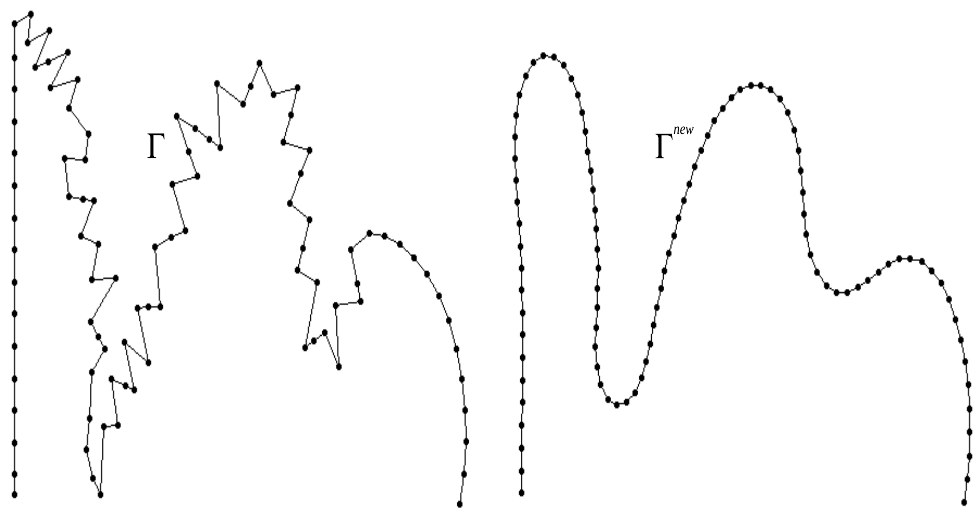

Figure 6. *Before and after smoothing of an open curve using*
Algorithm 2 with 10 sweeps.

In Figure 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 Figure 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 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 affect 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 but "closable" (such as the grid in Figure 10 which can be closed by adding triangles to “cap” the ends), 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 $\mathbf{x}$ based on data from its immediate neighbors. Consider Figure 7, here $\mathbf{x}$ is surrounded by the points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}$, which form a counter-clockwise cycle when viewed from “outside” the surface. We define

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

We contemplate moving $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}^{\mathrm{tang}}$, where $d\mathbf{x}^{\mathrm{tang}}$ is in a direction tangential to the surface at $\mathbf{x}$ so that the neighborhood of $\mathbf{x}$ is “tangentially smoothed”. Then we further reposition $\mathbf{x}$ by $h\hat{\mathbf{n}}$. That is, to ensure conservation of volume, we further move $\mathbf{x}$ by some multiple h of the unit normal $\hat{\mathbf{n}}$ at the surface.

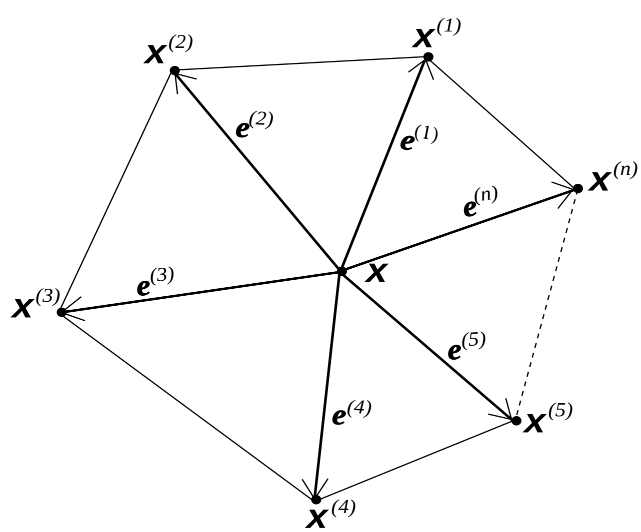

Figure 7. *Node* $\mathbf{x}$ *on triangular faceted surface surrounded by* n *neighbors* $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$.

The "normal" direction chosen is somewhat arbitrary, since the faceted surface is not smooth at $\mathbf{x}$ and so no true normal exists, but a reasonable choice would be the normalized sum of area vectors of all the triangles incident on $\mathbf{x}$. That is

$$\hat{\mathbf{n}} = \frac{\sum_{i=1}^{n} \mathbf{e}^{(i)} \times \mathbf{e}^{(i+1)}}{|| \sum_{i=1}^{n} \mathbf{e}^{(i)} \times \mathbf{e}^{(i+1)} ||}. \tag{1}$$

We are contemplating a move of $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}^{\text{tang}} + h\hat{\mathbf{n}}$, where h is to be chosen so that volume is conserved. Now the change of volume is given by

$$\begin{aligned} 6dV &= \sum_{i=1}^{n} (d\mathbf{x}^{\text{tang}} + h\hat{\mathbf{n}}) \cdot \mathbf{e}^{(i)} \times \mathbf{e}^{(i+1)} \\ &= (d\mathbf{x}^{\text{tang}} + h\hat{\mathbf{n}}) \cdot \sum_{i=1}^{n} \mathbf{e}^{(i)} \times \mathbf{e}^{(i+1)}. \end{aligned} \tag{2}$$

(The "6" arises from use of the volume formula for tetrahedra employed for the terms in the sum in (2).) Thus $dV = 0$ implies

$$\begin{aligned} h &= \frac{-d\mathbf{x}^{\text{tang}} \cdot \sum_{i=1}^{n} \mathbf{e}^{(i)} \times \mathbf{e}^{(i+1)}}{\hat{\mathbf{n}} \cdot \sum_{i=1}^{n} \mathbf{e}^{(i)} \times \mathbf{e}^{(i+1)}} \\ &= -d\mathbf{x}^{\text{tang}} \cdot \hat{\mathbf{n}}, \end{aligned}$$

using (1). Now if $d\mathbf{x}^{\text{tang}}$ were in the plane orthogonal to $\hat{\mathbf{n}}$, then we would have $h = 0$ and no normal correction is necessary. However, if our "tangential" smoothing scheme produces a displacement $d\mathbf{x}^{\text{tang}}$ that is only *roughly* tangential, then the correction $h\hat{\mathbf{n}} = -(d\mathbf{x}^{\text{tang}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$ will be nonzero.

This is the case for the choice of *Laplacian* smoothing for the tangential smoothing scheme. In Laplacian smoothing, a point is simply moved to the average position of its neighbors. That is

$$\mathbf{x} + d\mathbf{x}^{\text{tang}} = \frac{\sum_{i=1}^{n} \mathbf{x}^{(i)}}{n}.$$

In this case, with the $\mathbf{x}^{(i)}$ being only roughly in the tangent plane of the surface at $\mathbf{x}$, the displacement $d\mathbf{x}^{\text{tang}}$ will be essentially tangential, but a small correction $h\hat{\mathbf{n}}$ will have to be made. Note: Planar smoothing schemes more sophisticated than Laplacian smoothing are available [2].

Algorithm 3: Volume-Conserving smoothing of a surface using single-node relaxations.

Repeat (sweep) until "done"

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

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

$$d\mathbf{x}^{\mathrm{tang}} \leftarrow \frac{\sum_{i=1}^{n} \mathbf{x}^{(i)}}{n} - \mathbf{x}$$

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

The flaw of this scheme is identical to that of the analogous scheme (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 Figure 3) unchanged.

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

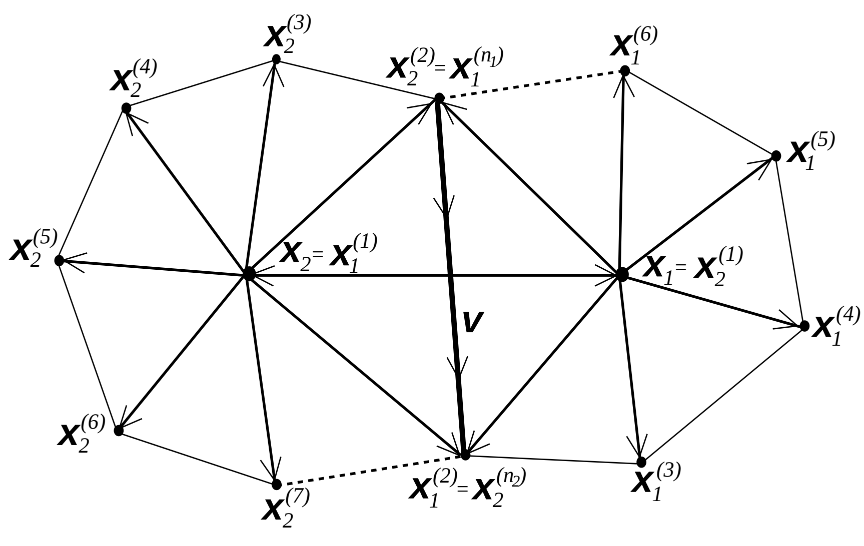

Figure 8. *Nomenclature for nodes surrounding edge* $\overline{\mathbf{x}_1\mathbf{x}_2}$ *on triangular faceted surface.*

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

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

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

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

is the "area" vector associated with $\mathbf{x}_i$ which is the sum of the area vectors of the triangles incident on $\mathbf{x}_i$. (Note: Actually it would be more proper to call $\mathbf{A}_i$ the "double area vector", since it is formed by summing vectors which are normal to each triangle with magnitude equal to double the area of each triangle.) A reasonable choice for the direction "normal" to the surface at the edge $\overline{\mathbf{x}_1\mathbf{x}_2}$ would thus be

$$\hat{\mathbf{n}} = \frac{\mathbf{A}_1 + \mathbf{A}_2}{||\mathbf{A}_1 + \mathbf{A}_2||}.$$

We now contemplate moving $\mathbf{x}_i$ to $\mathbf{x}_i + d\mathbf{x}_i^{\text{tang}}$, where the $d\mathbf{x}_i^{\text{tang}}$ are roughly tangential to the surface, so that the neighborhood of $\overline{\mathbf{x}_1\mathbf{x}_2}$ is tangentially smoothed. Then we move $\mathbf{x}_1$, $\mathbf{x}_2$ in the normal direction $\hat{\mathbf{n}}$ so that volume is conserved *and* normal smoothing is undertaken. As in the curve smoothing case, the meaning of "normal smoothing" is equalization of the normal components of the positions of $\mathbf{x}_1$ and $\mathbf{x}_2$.

As in Algorithm 3, we can define the $d\mathbf{x}_i^{\text{tang}}$ by Laplacian smoothing. That is

$$d\mathbf{x}_i^{\text{tang}} = \frac{\sum_{j=1}^{n_i} \mathbf{x}_i^{(j)}}{n_i} - \mathbf{x}_i, \quad i = 1, 2.$$

Now, defining

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + d\mathbf{x}_i^{\text{tang}},$$

and if we define

$$d\mathbf{x}_i = d\mathbf{x}_i^{\text{tang}} + (h - \tilde{\mathbf{x}}_i \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}, \tag{3}$$

then

$$(\mathbf{x}_i + d\mathbf{x}_i) \cdot \hat{\mathbf{n}} = h, \quad i = 1, 2.$$

Thus $(h - \tilde{\mathbf{x}_i} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$ represents a readjustment in the normal direction that will equalize the normal components of the $\mathbf{x}_i$'s. It thus remains for us to derive the value that will result in conservation of volume.

The change in volume caused by the move of the $\mathbf{x}_i$ to $\mathbf{x}_i + d\mathbf{x}_i$ is computed as follows. First, the movement of $\mathbf{x}_1$ 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 $\mathbf{x}_1$ 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

$$\begin{aligned} 6dV_1 &= \sum_{j=1}^{n_1} d\mathbf{x}_1 \cdot \mathbf{e}_1^{(j)} \times \mathbf{e}_1^{(j+1)} \\ &= d\mathbf{x}_1 \cdot \mathbf{A}_1. \end{aligned} \tag{4}$$

Next, the movement of $\mathbf{x}_2$ to $\mathbf{x}_2 + d\mathbf{x}_2$ creates a volume change similar to (4), but we must take into account that $\mathbf{x}_1 = \mathbf{x}_2^{(1)}$ 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

$$\begin{aligned} \widetilde{\mathbf{e}_2^{(1)}} &= \mathbf{e}_2^{(1)} + d\mathbf{x}_1 \\ \widetilde{\mathbf{e}_2^{(j)}} &= \mathbf{e}_2^{(j)}, \qquad\qquad 2 \le j \le n_2, \end{aligned}$$

we have

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

where $\mathbf{v} \equiv \mathbf{e}_2^{(n_2)} - \mathbf{e}_2^{(2)}$ $(= \mathbf{e}_1^{(2)} - \mathbf{e}_1^{(n_1)})$. Thus conservation of volume requires us to have

$$\begin{aligned} 0 = 6dV &= 6dV_1 + 6dV_2 \\ &= d\mathbf{x}_1 \cdot \mathbf{A}_1 + d\mathbf{x}_2 \cdot \mathbf{A}_2 + d\mathbf{x}_2 \cdot \mathbf{v} \times d\mathbf{x}_1. \end{aligned}$$

Substituting (3) into this expression, we obtain

$$\begin{aligned}
0 &= [d\mathbf{x}_1^{\text{tang}} + (h - \tilde{\mathbf{x}_1} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \cdot \mathbf{A}_1 + [d\mathbf{x}_2^{\text{tang}} + (h - \tilde{\mathbf{x}_2} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \cdot \mathbf{A}_2 \\
&\quad + [d\mathbf{x}_2^{\text{tang}} + (h - \tilde{\mathbf{x}_2} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \cdot \mathbf{v} \times [d\mathbf{x}_1^{\text{tang}} + (h - \tilde{\mathbf{x}_1} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \\
&= [d\mathbf{x}_1^{\text{tang}} - (\tilde{\mathbf{x}_1} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \cdot \mathbf{A}_1 + [d\mathbf{x}_2^{\text{tang}} - (\tilde{\mathbf{x}_2} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \cdot \mathbf{A}_2 \\
&\quad + h\hat{\mathbf{n}} \cdot (\mathbf{A}_1 + \mathbf{A}_2) + [d\mathbf{x}_2^{\text{tang}} - (\tilde{\mathbf{x}_2} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \cdot \mathbf{v} \times [d\mathbf{x}_1^{\text{tang}} - (\tilde{\mathbf{x}_1} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}] \\
&\quad + h(d\mathbf{x}_2^{\text{tang}} - d\mathbf{x}_1^{\text{tang}}) \cdot \mathbf{v} \times \hat{\mathbf{n}}.
\end{aligned}$$

Defining $\mathbf{p}_i = d\mathbf{x}_i^{\text{tang}} - (\tilde{\mathbf{x}_i} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}},\ i = 1, 2$, we obtain

$$h = -\frac{\mathbf{p}_1 \cdot \mathbf{A}_1 + \mathbf{p}_2 \cdot \mathbf{A}_2 + \mathbf{p}_2 \cdot \mathbf{v} \times \mathbf{p}_1}{\hat{\mathbf{n}} \cdot (\mathbf{A}_1 + \mathbf{A}_2) + (d\mathbf{x}_2^{\text{tang}} - d\mathbf{x}_1^{\text{tang}}) \cdot \mathbf{v} \times \hat{\mathbf{n}}}.$$

The computation of h in this last expression might lead to problems: (1) the denominator is close to round-off, or (2) computed h is very large, leading to very large $d\mathbf{x}_i$. Case (1) is rarely expected, since

$$\hat{\mathbf{n}} \cdot (\mathbf{A}_1 + \mathbf{A}_2) = ||\mathbf{A}_1 + \mathbf{A}_2||$$

is positive for nonpathological cases, and the $d\mathbf{x}_i^{\text{tang}}$ are expected to be relatively small. Case (2) is possible only in some pathological cases where arbitrary movement of $\mathbf{x}_1, \mathbf{x}_2$ in the direction $\hat{\mathbf{n}}$ has a negligible effect on the magnitude of the volume enclosed. Although these cases rarely occur, a robust algorithm must anticipate them. Our algorithm detects if the computed $||d\mathbf{x}_i||$ exceed a maximum allowed movement distance and reverts to single node relaxation in this circumstance. Our choice for the maximum allowed movement distance is a small multiple (*e.g.* 2) of the length of the longest $\mathbf{e}_i^{(j)}$.

Figures 9 and 10 show results of smoothing highly jagged grids using Algorithm 4. In Figure 9, we converted the closed quadrilateral surface into a closed triangular surface by adding diagonal segments. It is the volume enclosed by the derived triangular surface that is preserved. In Figure 10, we only relaxed edges having both endpoints not on the two closed boundary curves at either end of the grid. Volume is conserved in the sense that if the grid were closed in some arbitrary fashion, then the performed smoothing operations would not have changed the volume of the closed grid. (*e.g.* the "capped" figure formed by adding planar disks at each end would not have had its volume altered by the performed smoothing operations.)

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 $\{\mathbf{x}_i^{(j)}\}_{i=1,2}^{j=1,\ldots,n_i}$, relax edge:

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

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

$\hat{\mathbf{n}} \leftarrow \frac{\mathbf{A_1}+\mathbf{A_2}}{||\mathbf{A_1}+\mathbf{A_2}||}$

$d\mathbf{x}_i^{\text{tang}} \leftarrow \frac{\sum_{j=1}^{n_i} \mathbf{x}_i^{(j)}}{n_i} - \mathbf{x}_i, \quad i = 1, 2$

$\mathbf{p}_i \leftarrow d\mathbf{x}_i^{\text{tang}} - [(\mathbf{x}_i + d\mathbf{x}_i^{\text{tang}}) \cdot \hat{\mathbf{n}}]\hat{\mathbf{n}}, \quad i = 1, 2$

$\text{denom} \leftarrow \hat{\mathbf{n}} \cdot [\mathbf{A}_1 + \mathbf{A}_2 + (d\mathbf{x}_2^{\text{tang}} - d\mathbf{x}_1^{\text{tang}}) \times \mathbf{v}]$

$\text{numer} \leftarrow -(\mathbf{p}_1 \cdot \mathbf{A}_1 + \mathbf{p}_2 \cdot \mathbf{A}_2 + \mathbf{p}_2 \cdot \mathbf{v} \times \mathbf{p}_1)$

If $|\text{denom}| < \epsilon$ then

$\hat{\mathbf{n}}_i \leftarrow \frac{\mathbf{A}_i}{||\mathbf{A}_i||}, \quad i = 1, 2$

$d\mathbf{x}_i \leftarrow d\mathbf{x}_i^{\text{tang}} - (d\mathbf{x}_i^{\text{tang}} \cdot \hat{\mathbf{n}}_i)\hat{\mathbf{n}}_i, \quad i = 1, 2$

else

$h \leftarrow \frac{\text{numer}}{\text{denom}}$

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

$e \leftarrow \max_{i,j}\{||\mathbf{e}_i^{(j)}||\}$

$k \leftarrow 2$

If $\max(||d\mathbf{x}_1||, ||d\mathbf{x}_2|| > ke)$ then

$\hat{\mathbf{n}}_i \leftarrow \frac{\mathbf{A}_i}{||\mathbf{A}_i||}, \quad i = 1, 2$

$d\mathbf{x}_i \leftarrow d\mathbf{x}_i^{\text{tang}} - (d\mathbf{x}_i^{\text{tang}} \cdot \hat{\mathbf{n}}_i)\hat{\mathbf{n}}_i, \quad i = 1, 2$

$\mathbf{x}_i \leftarrow \mathbf{x}_i + d\mathbf{x}_i, \quad i = 1, 2$

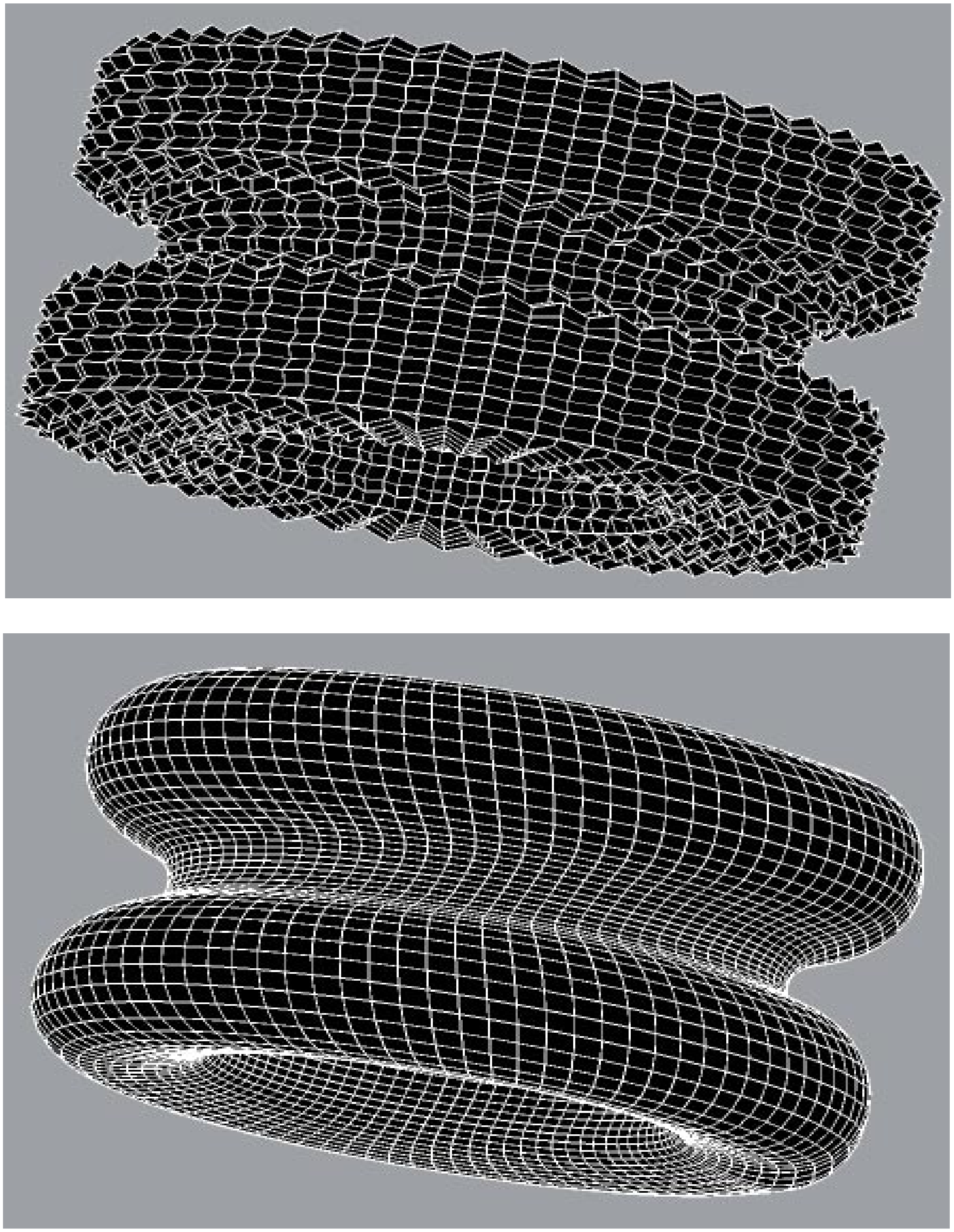

Figure 9. *Before and after smoothing of a closed quadrilateral surface grid using* 25 sweeps of Algorithm 4.

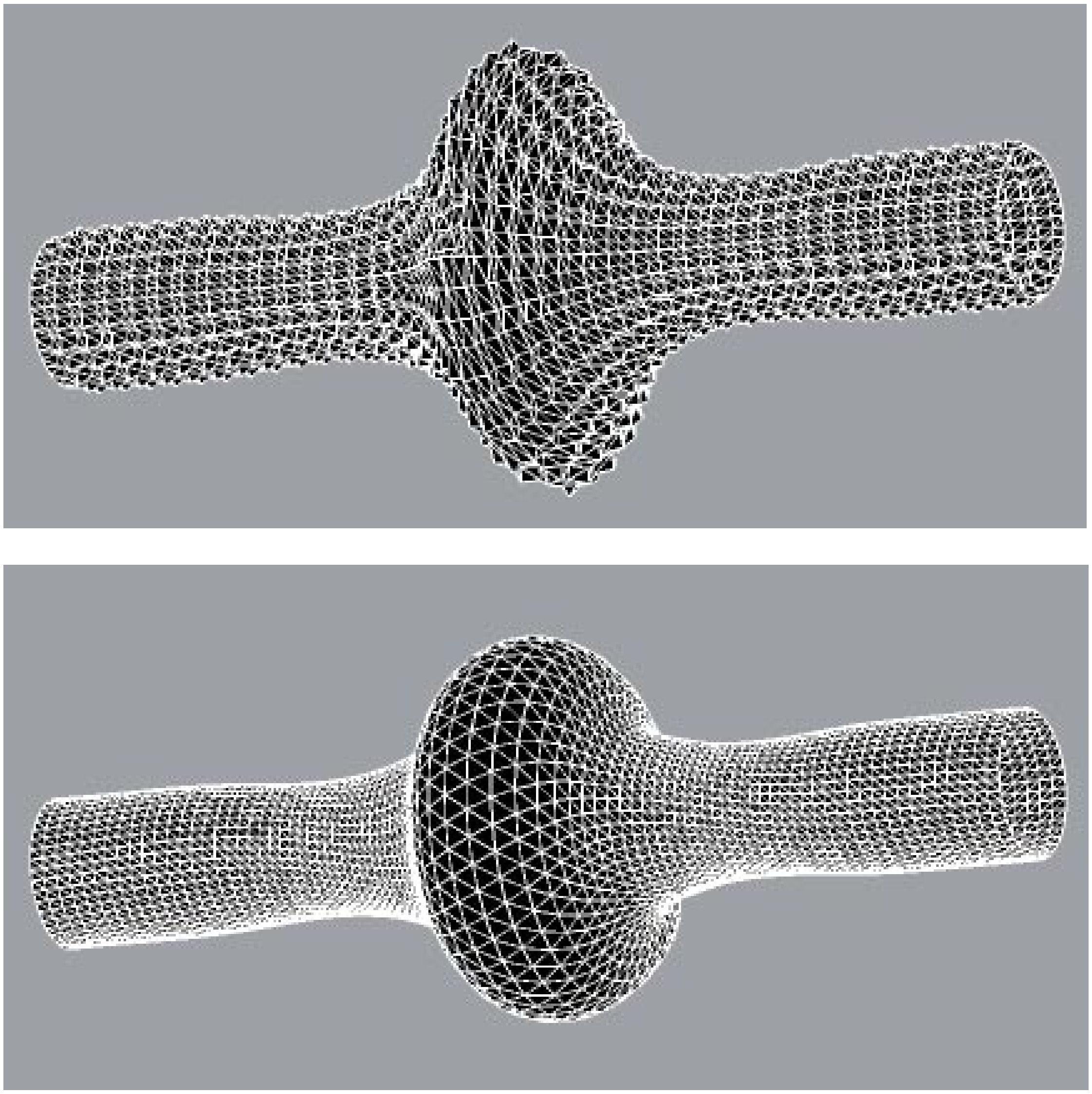

Figure 10. *Before and after smoothing of an open triangular surface grid using 25 sweeps of Algorithm 4.*

References

1. N. Carlson and K. Miller, *Design and Application of a Gradient-Weighted Moving Finite Element Code, Part II, in 2D*, *SIAM J. Sci. Comput.*, to appear.

2. A. Khamayseh and A. Kuprat, *Anisotropic Smoothing and Solution Adaption for Unstructured Grids*, *Intl. J. for Num. Meth. in Engr.*, **39**, pp. 3163-3174 (1996).

3. A. Khamayseh and C. W. Mastin, *Computational Conformal Mapping for Surface Grid Generation*, *J. Comput. Phys.*, **123**, pp. 394–401 (1996).

4. A. Kuprat and D. George, *Maintaining Tetrahedral Mesh Quality in Response to Time-Dependent Topological and Geometrical Deformation.* Proceedings of the Sixth International Conference on Numerical Grid Generation in Computational Field Simulation, pp. 1-10 (1998).

5. K. Miller, *A Geometrical-Mechanical Interpretation of Gradient-Weighted Moving Finite Elements*, SIAM J. Num. Anal., **34**, pp. 67-90 (1997).

6. Z. U. A. Warsi, *Numerical Grid Generation in Arbitrary Surfaces Through a Second-Order Differential Geometric Model*, *J. Comput. Phys.* **64**, pp. 82–96 (1986).