LA-UR-95-2205

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: **ANISOTROPIC SMOOTHING AND SOLUTION ADAPTION FOR UNSTRUCTURED GRIDS**

AUTHOR(S): Ahmed Khamayseh, Andrew Kuprat

SUBMITTED TO: International Journal for Numerical Methods in Engineering

Los Alamos

Los Alamos National Laboratory
Los Alamos New Mexico 87545

ANISOTROPIC SMOOTHING AND SOLUTION ADAPTION FOR UNSTRUCTURED GRIDS

AHMED KHAMAYSEH AND ANDREW KUPRAT
Los Alamos National Laboratory
Los Alamos, New Mexico 87545, U.S.A.

SUMMARY

An elliptic smoothing scheme for 2-D structured meshes is generalized to the case of 2-D unstructured meshes. The resulting scheme is similar to the familiar Laplacian smoothing scheme, but exhibits superior node diffusion in anisotropic domains. We then show further improvement of grid quality when smoothing is alternated with Lawson flipping (a technique commonly used to generate Delaunay triangulations). Two additional enhancements ("controlled" and "adaptive" smoothing) allow us to create grids suitable for a realistic MOSFET semiconductor application.

1. INTRODUCTION

In the structured mesh community, smoothing schemes have been developed to a high degree for the production of high quality quadrilateral meshes.[1,2] In contrast, smoothing in the unstructured world of mainly triangular meshes is considerably less developed. One major problem is the absence of global curvilinear coordinates on these meshes. With structured meshes, one can easily construct a conformal or quasiconformal mapping between a logical (ξ, η)-space and the physical (x, y)-space. With unstructured meshes the situation is confused by the irregularity of the topological connectivity, leading to no such obvious mapping.

In the intermediate case of a *regular* triangular mesh, Winslow was able to find global curvilinear coordinates, and constructed a smoothing scheme by requiring the coordinates to each satisfy Laplace's equation.[3] Far from boundaries, this scheme tends towards making the position of each node equal to the average position of its neighbours. Subsequently, researchers in need of a smoothing algorithm for fully unstructured meshes have continued with the idea of replacing a node by the average position of its neighbours (even though no obvious global mapping exists to justify this process), and this has come to be known as Laplacian smoothing for unstructured grids.

One well known problem with this is that near boundaries, Laplacian smoothing can produce node spillover (where the orientation of triangles are reversed, and the mesh becomes unusable). Another problem that we have observed is poor node diffusion in anisotropic domains. Unweighted node position averaging is inherently isotropic; in this paper we will introduce weights (depending on distances between nodes) that make the scheme anisotropic and more appropriate for anisotropic domains.

Indeed, we view Laplacian smoothing for unstructured grids as a generalization of a conformal mapping smoothing scheme for structured grids. Conformal maps are isotropic; squares are mapped to squares, not rectangles. Using a *quasi*-conformal mapping (a generalization of a conformal mapping which allows for a non-unit aspect ratio) we derive a new smoothing scheme which we call Elliptic Smoothing for Unstructured Grids (ESUG). In an example, we show superior diffusion of a source of points into an anisotropic domain.

In light of the extreme mobility of points under ESUG, we then develop *controlled* ESUG which limits node mobility to a user-desired degree, for those cases where complete node diffusion is not desired. Also developed is a solution adaption capability in which the mesh smoothing algorithm is modified to move grid points into areas where an objective function has large gradients. We present some real-world applications of these algorithms to MOSFET semiconductor modeling at the end of this paper.

2. ELLIPTIC SMOOTHING FOR A 2D STRUCTURED MESH

Elliptic systems have been widely recognized among the structured grid generation community to be an efficient tools to construct high quality meshes.[1] In 2D a robust elliptic system can be based on the quasiconformal mapping equations[2] to produce smooth, adaptive, orthogonal coordinates. Solution of these equations constitutes an elliptic smoothing scheme for these kinds of meshes. Here we review this scheme for structured quadrilateral meshes; this scheme will be generalized to unstructured meshes in the following section.

We assume that the boundary nodes in our quadrilateral mesh are fixed, but we are free to move interior nodes. It is desired that interior node positions be adjusted to achieve a smooth variation in quadrilateral shape and area given the constraints on the boundary nodes. Now consider a quasiconformal mapping $\varphi = \xi(x,y) + i\eta(x,y)$ from the region D in (x,y)-space to a region R in (ξ,η)-space, Figure 1.

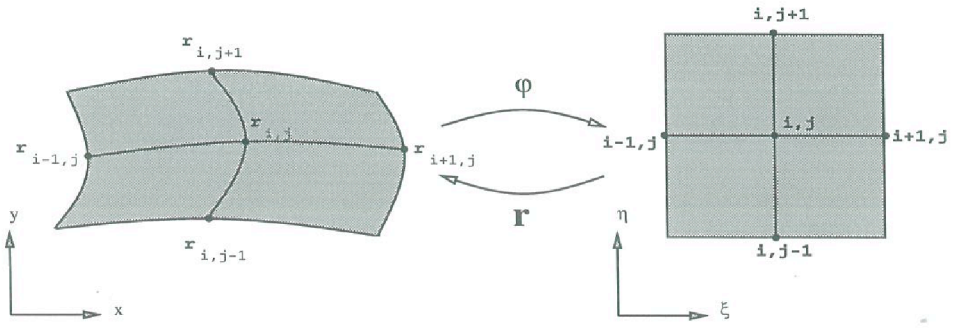

Figure 1. Quasiconformal mapping φ from physical space to logical space.

By definition, the real and imaginary parts of φ are solutions to Beltrami's equations

$$\begin{aligned} \xi_x &= \mathcal{M}(c\eta_y + b\eta_x) \\ -\xi_y &= \mathcal{M}(a\eta_x + b\eta_y) \end{aligned} \tag{2.1}$$

where a, b, c are functions of x and y with $a, c > 0$ and satisfy the equation $ac - b^2 = 1$ on D and $\mathcal{M}$ is the dilation of the mapping.

We only consider the special case where $a = c = 1$ and $b = 0$. Then (2.1) reduces to

$$\begin{aligned} \xi_x &= \mathcal{M}\eta_y \\ -\xi_y &= \mathcal{M}\eta_x. \end{aligned} \tag{2.2}$$

This system is just the Cauchy-Riemann equations, except for the stretching factor $\mathcal{M}$. Under these conditions, we see that the mapping is orthogonal and satisfies Laplace's equation:

$$\Delta\varphi = 0.$$

The inverse mapping satisfies

$$g_{22}\mathbf{r}_{\xi\xi} + g_{11}\mathbf{r}_{\eta\eta} = 0,$$

where $\mathbf{r} = \big(x(\xi,\eta), y(\xi,\eta)\big)$, $g_{11} = \mathbf{r}_\xi \cdot \mathbf{r}_\xi$, and $g_{22} = \mathbf{r}_\eta \cdot \mathbf{r}_\eta$.[2] This system is then discretized to be

$$\mathbf{r}_{i,j} = \frac{\omega_{i+1,j}^2\mathbf{r}_{i+1,j} + \omega_{i-1,j}^2\mathbf{r}_{i-1,j} + \omega_{i,j+1}^2\mathbf{r}_{i,j+1} + \omega_{i,j-1}^2\mathbf{r}_{i,j-1}}{\omega_{i+1,j}^2 + \omega_{i-1,j}^2 + \omega_{i,j+1}^2 + \omega_{i,j-1}^2}, \tag{2.3}$$

where the weights $\omega_{i+1,j} = \sqrt{g_{22}}$, $\omega_{i-1,j} = \sqrt{g_{22}}$, $\omega_{i,j+1} = \sqrt{g_{11}}$, and $\omega_{i,j-1} = \sqrt{g_{11}}$.

A smoothing algorithm based on (2.3) is then implemented using Gauss-Seidel relaxations. Nodes are relaxed in sequential order using (2.3), and each new node location is immediately incorporated in subsequent relaxations. After a sufficient number of Gauss-Seidel sweeps, (2.3) is approximately satisfied, and the scheme is deemed to have converged.

3. GENERALIZATION OF ELLIPTIC SMOOTHING TO UNSTRUCTURED GRIDS

It is now desired to generalize (2.3) to the case where we have a 2-dimensional *unstructured* mesh. Hence, we do not assume a quadrilateral element shape or any regular mesh connectivity. Typically in such a mesh, we are dealing with triangular elements, but we are not restricting ourselves to this case.

For a node q in such an unstructured mesh, list the neighbours p of q in counterclockwise order $\{p_1, p_2, ..., p_{\deg(q)}\}$. Consider the following smoothing scheme:

$$\mathbf{r}_q = \frac{\sum_{k=1}^{\deg(q)} (d_{p_{k-1},p_{k+1}})^2 \mathbf{r}_{p_k}}{\sum_{k=1}^{\deg(q)} (d_{p_{k-1},p_{k+1}})^2}. \tag{3.1}$$

Here, $d_{p_{k-1},p_{k+1}}$ refers to the distance between nodes p_{k-1} and p_{k+1}, and the subscripts "$k-1$" and "$k+1$" are evaluated *modulo* $\deg(q)$. (See Figure 2.)

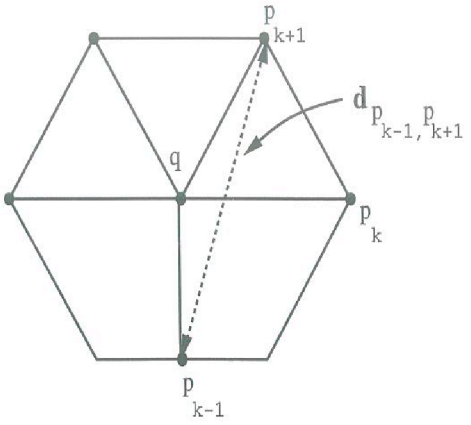

Figure 2. Distance used for weight of p_k in the relaxation of node q in an unstructured grid.

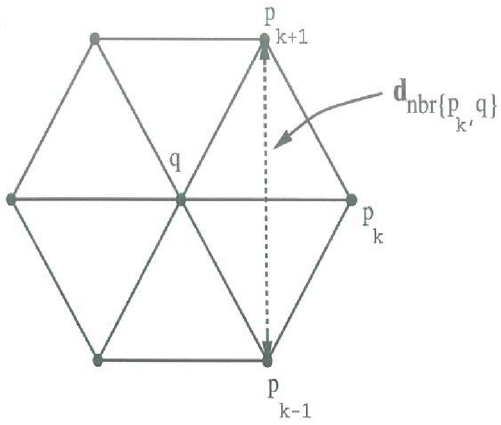

Figure 3. In a *triangular* unstructured grid, distance used for weight of p_k in the relaxation of node q equals that for q in the relaxation of node p_k.

Thus $\mathbf{r}_q$ is taken to be the weighted sum of the positions of the neighbours, and the weight for $\mathbf{r}_{p_k}$ is taken to be the square of the distance between the neighbours that come immediately before and after p_k in the listing of neighbours of q. In the case of triangular unstructured grids, p_{k-1} and p_{k+1} have the property that they are the *two unique mutual neighbours* of q and p_k. Then (3.1) is more naturally written as

$$\mathbf{r}_q = \frac{\sum_p (d_{\mathrm{nbr}\{p,q\}})^2 \mathbf{r}_p}{\sum_p (d_{\mathrm{nbr}\{p,q\}})^2}, \tag{3.2}$$

where $d_{\mathrm{nbr}\{p,q\}}$ denotes the distance between the two mutual neighbours of nodes p and q, and the sum is taken over all neighbours p of q. (See Figure 3.)

We now claim that this scheme is identical to (2.3) in the special case of *structured* quadrilateral meshes. Indeed, observe that for the node at $\mathbf{r}_{i,j}$ in (2.3), the neighbours in counterclockwise order are $(i+1,j),(i,j+1),(i-1,j)$, and $(i,j-1)$. Now in the notation of (3.1), let $q=(i,j)$ and consider $p_1=(i+1,j)$. Then,

$$\begin{aligned} d_{p_4,p_2} &= \text{distance between } \mathbf{r}_{i,j-1} \text{ and } \mathbf{r}_{i,j+1} \\ &= 2|\mathbf{r}_\eta| \\ &= 2\omega_{i+1,j}. \end{aligned}$$

Similarly, we find that

$$d_{p_1,p_3} = 2\omega_{i,j+1},$$

$$d_{p_2,p_4} = 2\omega_{i-1,j},$$

$$d_{p_3,p_1} = 2\omega_{i,j-1},$$

and so all the unnormalized weights are identical to within a factor of 2^2, and hence the schemes are identical after normalization of the weights.

Equations (3.1) or (3.2) represent a generalization of elliptic smoothing to unstructured meshes. Note that if we had restricted ourselves to using merely a *conformal* mapping, then we would have had (2.2) with $\mathcal{M} = 1$. In this case, it is clear that we would have been led to (3.1) or (3.2) with the weights all set equal, which is the usual Laplacian smoothing scheme.

What we have *not* exhibited in our generalization is a global quasiconformal mapping, but that would be a daunting task, given the fact that the mesh connectivity is completely arbitrary. Also, our generalization is certainly not the only one possible. What we *do* exhibit in our generalization is the essential feature of the structured case algorithm: distance (squared) weighting, with the distance measured in a "transverse" direction. Hence it is not surprising that ESUG works in practice; we will see in the next section the superior ability of ESUG to diffuse points into an anisotropic domain. An added advantage of distance weighting is that the scheme is then naturally generalizable to solution adaption, as will be seen in Section 6.

4. COMPARISON OF ESUG WITH LAPLACIAN SMOOTHING IN AN ANISOTROPIC DOMAIN

In Figure 4, we consider a rectangle with 4:1 aspect ratio with an isotropic "source" of points in the middle. In Figures 5 and 6, we compare the effects of ESUG versus Laplacian smoothing. Clearly, the dense source of points in the middle has diffused far more in ESUG than in Laplacian smoothing. In the ESUG case, we see that the boundary triangles closest to the source of nodes have been allowed to deform in an appropriate anisotropic fashion, thus allowing the isotropic source of points to expand more than in the Laplacian case. In contrast, the boundary triangles nearest the middle in the Laplacian case are more 'rigid' and expansion of the source of points is retarded.

Next we consider the effect of a sequence of smoothings and Lawson flips on these meshes. Lawson flips break the connectivity of the mesh and establish a Delaunay triangulation which is a common requirement for computation.[4] One might speculate that a repeated sequence of smoothings and Lawson flips would produce a uniform Delaunay mesh. In Figure 8, we have subjected our initial grid to an alternating sequence of ESUG and Lawson flips, performing both procedures three times each. The result is a Delaunay

mesh which appears as diffused as is possible given the fact that we have chosen not to move the points on the boundary.

In contrast, the same procedure with Laplacian smoothing substituted for ESUG yields Figure 7. Here the diffusion process has come to a standstill and does not appear to be enhanced by Lawson flipping. It is here quite apparent that the exclusive use of Laplacian smoothing along with changes of mesh connectivity may be *insufficient* to equilibrate node densities in many kinds of problem domains.

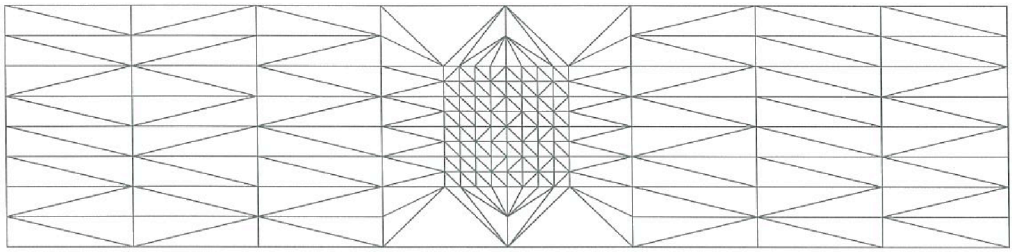

Figure 4. Initial grid.

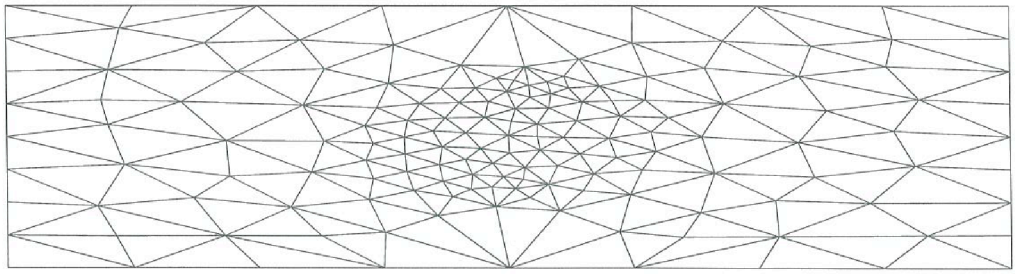

Figure 5. Grid after Laplacian smoothing only.

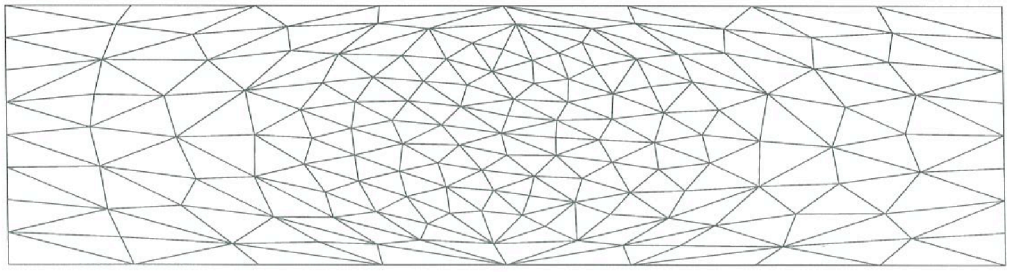

Figure 6. Grid after anisotropic elliptic smoothing only.

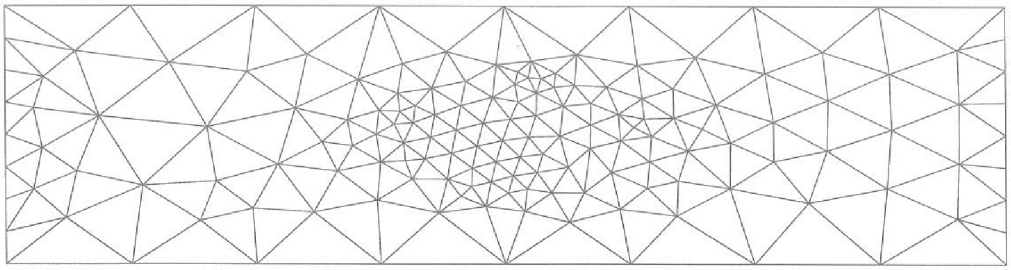

Figure 7. Grid after Laplacian smoothing with Lawson flips.

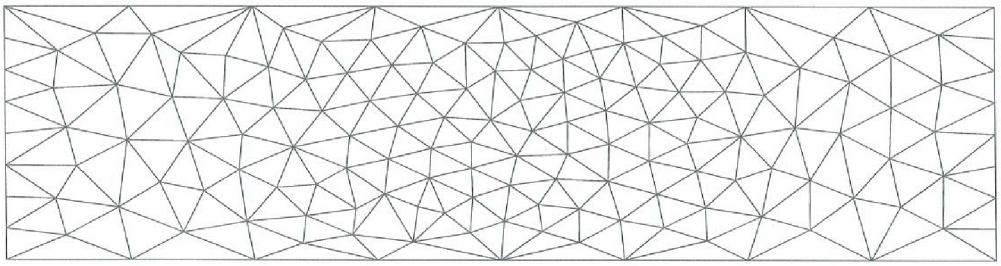

Figure 8. Grid after anisotropic elliptic smoothing with Lawson flips.

5. CONTROLLED SMOOTHING

In practice, unstructured meshes used for the solution of partial differential equations (PDEs) are frequently refined in locations of critical behaviour. Refinement algorithms perform a standard action such as edge bisection in such regions. Certainly, after sufficient refinements, any desired node density can be achieved. However it is often the case that refinement results in a poor distribution of triangle areas and shapes. Such grids are excellent candidates for smoothing. However smoothing schemes, when implemented in unmodified "uncontrolled" form, will cause refined regions to "smear out", reducing the desired high node density in these regions. So, although smoothing is crucial for producing grids of satisfactory quality in the refined region, it must be *controlled* in some way to retain the presmoothed node density information.

Let us define a control parameter λ that ranges from 0 to 1. It is desired that a new *controlled* smoothing scheme be devised such that $\lambda = 0$ corresponds to the original, uncontrolled smoothing scheme, and $\lambda = 1$ corresponds to no mesh movement whatsoever (the identity map). As λ changes continuously from 0 to 1, it is desired that a continuum of smoothing schemes is obtained that are increasingly inhibited in the amount of node

movement allowed. The following simple modification to (3.2) satisfies these criteria:

$$\mathbf{r}_q = \lambda \mathbf{r}_q^0 + (1-\lambda)\frac{\sum_p (d_{\text{nbr}\{p,q\}})^2 \mathbf{r}_p}{\sum_p (d_{\text{nbr}\{p,q\}})^2}, \tag{5.1}$$

Here, $\mathbf{r}_q^0$ corresponds to the *original* undisturbed position of node q. In mathematical terms, we have constructed a simply homotopy between the identity map and our original smoothing scheme. Indeed, one could obtain such homotopies for any Gauss-Seidel node relaxation scheme by substituting those schemes into the righthand side of (5.1). So, for example, Laplace smoothing would become

$$\mathbf{r}_q = \lambda \mathbf{r}_q^0 + (1-\lambda)\frac{\sum_p \mathbf{r}_p}{\deg(q)}. \tag{5.2}$$

Not only is (5.1) a simple modification of (3.2), but we have found in practice that it retains the desirable element shape improving qualities of the original scheme. Although element areas are not *globally* equilibrated (for some nonzero value of λ), element areas are allowed to *locally* equilibrate—and this is exactly what is desired.

6. ADAPTIVE ELLIPTIC SMOOTHING FOR UNSTRUCTURED GRIDS

In many applications, it is desirable to smooth the mesh in such a fashion as to adapt it to some function defined over that mesh. Suppose the function $f(x,y)$ is the solution to a PDE, and it is desired to move nodes into regions where the gradient of f is large. This has the effect of packing grid into regions where the function f undergoes rapid change, and where resolution is needed. (It can be argued that in many cases it is in regions of *curvature* of f that high node density is desired[5], but this is essentially equivalent to packing grid into regions where the functions $g = \frac{\partial f}{\partial x}$ and $h = \frac{\partial f}{\partial y}$ have large gradients.)

Our original ESUG scheme can be readily turned into a scheme for adapting to ∇f by recognizing that the $d_{\text{nbr}\{p,q\}}$ in (3.2) have the dimensions of distance. These distances can readily be "warped" to force the scheme to adapt to the gradients of f. Indeed, if one considers the distance from a point p_k to a point $p_{k'}$ to be the distance along the *graph* of f:

$$(d_{p_k,p_{k'}})^2 = (x_{p_k} - x_{p_{k'}})^2 + (y_{p_k} - y_{p_{k'}})^2 + (f_{p_k} - f_{p_{k'}})^2, \tag{6.1}$$

then our elliptic scheme is being essentially performed *on* the graph of f (i.e., on the surface $z = f(x,y)$). Hence, if element areas are equidistributed on this surface, the effect of this will be to move grid points into the gradient regions (see Figure 9).

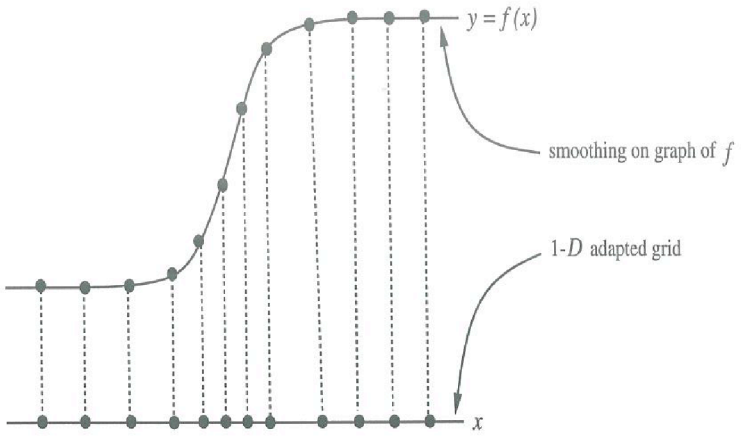

Figure 9. 1-D Simplification of Adaptive Smoothing Strategy.
In reality f is bivariate and we produce a 2-D adapted grid.

In mathematical terms, we are performing our ESUG scheme using a Riemannian metric for measuring the distance between points in the (x, y) plane.[6] This non-Euclidean distance, induced from the graph of a function via (6.1) can be used in any smoothing scheme involving distance weighting. It is apparent that shrinking or expanding the scale of the function f represents a free parameter that controls the degree of adaptivity. Indeed, as the scale of f is shrunken, the distance measure in (6.1) tends towards the standard Euclidean metric for $\mathbb{R}^2$, and the algorithm tends towards the standard (non-adaptive) smoothing algorithm. Also, it is clear that this adaptivity can be expanded to the case of adapting to n functions (as is the case in solving a *system* of PDEs) by incorporating a term for each function:

$$(d_{p_k,p_{k'}})^2 = (x_{p_k} - x_{p_{k'}})^2 + (y_{p_k} - y_{p_{k'}})^2 + (f^1_{p_k} - f^1_{p_{k'}})^2 + \ldots + (f^n_{p_k} - f^n_{p_{k'}})^2.$$

7. MAINTAINING A DELAUNAY TRIANGULATION AND MESH INTEGRITY

In practice, it is frequently necessary to follow smoothing with a Delaunay swap step[4,7], whereby Lawson flips are used to convert the smoothed grid to a Delaunay grid. Indeed for Voronoi cell finite volume calculations, it is necessary to have a Delaunay grid; for finite element calculations this may be unnecessary, but still may be beneficial. Hence, we commonly follow ESUG by Lawson swaps to produce a Delaunay mesh suitable for computing with a finite volume solver.

Another detail is "mesh integrity damping". Define "spillover" as the ejection of a node from the surrounding polygon formed by its neighbours during the relaxation process. Such an ejection reverses the orientation of triangles and destroys the integrity of the mesh. Note that both ESUG and Laplacian smoothing involve relaxing a node to a position equal to

the weighted average positions of its neighbours. Since the weights in both schemes are positive, it is clear that (assuming a Gauss-Seidel single node relaxation) node spillover cannot occur for either scheme when the surrounding neighbours form a convex polygon. However, when the neighbours of a node form a nonconvex polygon, both methods are prone to spillover.

To prevent spillover, we limit movement at each Gauss-Seidel node relaxation by multiplying the node displacement prescribed by (3.1), (3.2), or (5.1) by a damping factor which insures that no triangle has a reduction in area below a minimum threshold. However, this damping is only invoked in some cases of smoothing near nonconvex boundaries or adaptive smoothing to "challenging" functions with extreme changes in gradient.

8. MOSFET SEMICONDUCTOR EXAMPLE

In Figure 10 we show an initial grid taken from an actual MOSFET semiconductor application. In Figure 11 we show a perspective view of a doping function to which we wish to adapt our grid. (As can be seen, this function is piecewise linear and contains extremely sharp gradients. The fine grid in this figure is used only for the definition of the function and is *not* the grid that we are trying to adapt.) Adaptive smoothing is to be performed on the shaded portion of the grid in Figure 10, which corresponds to most of the Silicon substrate portion of the device.

In Figure 12 we show the effects of initially running the adaptive smoothing algorithm [(3.1) and (6.1)] on the grid in Figure 10. Good adaption to the steep gradient is observed. However we observe that better adaption can be obtained by using an alternating sequence of adaptive smoothing followed by Lawson flips, followed by more adaptive smoothing, etc. In Figure 13 we show the results of 10 adaptive smoothings, alternated with 10 rounds of Lawson flips. The flips cause topological changes that ultimately allow for a better adapted mesh. In fact, we note that Figure 10 and Figure 12 are topologically equivalent, and this topology is clearly not the best one for adapting to the function of Figure 11. This is indicated by the unnecessarily stretched triangles in Figure 12. In Figure 13 we see that Lawson flips have eliminated the unnecessarily stretched triangles, and have allowed more of the grid to move into the challenging steep gradient region.

A final technical note pertaining to the grid in Figure 13 is that the last stage of Lawson flipping also included the insertion of a small number of nodes (7) on the boundary of the smoothed region. This is to eliminate obtuse boundary-facing angles, which is a requirement of our finite volume solver. In general, obtuse boundary-facing angles (those angles opposite a boundary edge) cannot all be eliminated by Lawson flips of the interior edges. Hence the last stage of Lawson flipping actually consists of an alternating sequence of flips and boundary point insertions where necessary.

Next we try the alternate approach of using adaptive refinement followed by non-adaptive smoothing. In Figure 14 we have refined the initial grid in the shaded region where the doping function has the large gradient. Unfortunately the refinement has produced poorly-shaped triangles in this critical region, exhibiting a poor distribution of triangle areas. Also, due to asymmetry in the connectivity of the piecewise linear doping function, the refinement exhibits marked asymmetry. Then in Figure 15 we show the results of applying the plain (uncontrolled, non-adaptive) elliptic smoothing algorithm (3.2) on the shaded portion of Figure 14. The effect is that indeed the triangle areas are well equilibrated. However the regions of refinement have been oversmoothed, reducing the node density in these critical regions. Finally, in Figure 16, we show the results of our *controlled* elliptic smoothing algorithm (5.1). Here we have chosen $\lambda = \frac{1}{2}$. As can be seen, the node density in the critical regions is preserved but, as in Figure 15, triangle shape, area distribution, and symmetry are greatly improved. Finally we note that to make this grid suitable for finite volume computation, we should again follow smoothing by a round of Lawson flips with possible boundary point insertions.

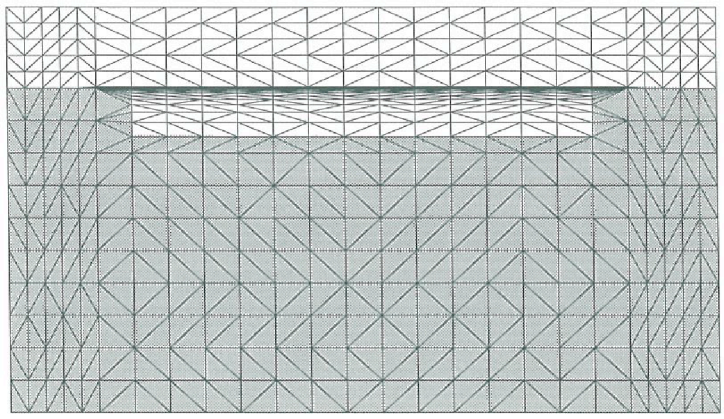

Figure 10. Initial MOSFET grid. Shaded region is targeted for adaptive smoothing.

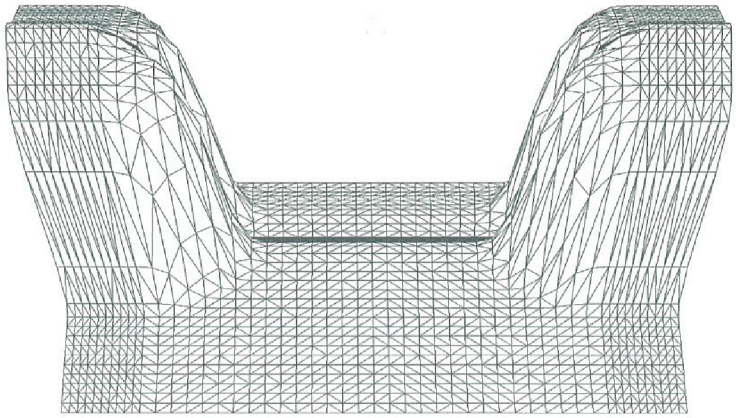

Figure 11. Perspective view of the doping function.

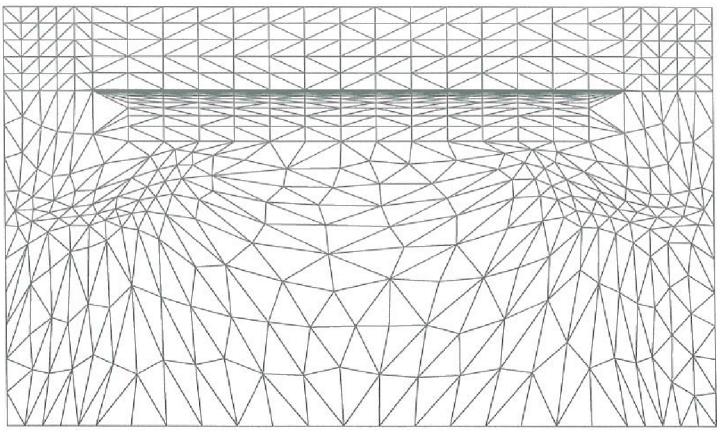

Figure 12. Grid after adaptive elliptic smoothing.

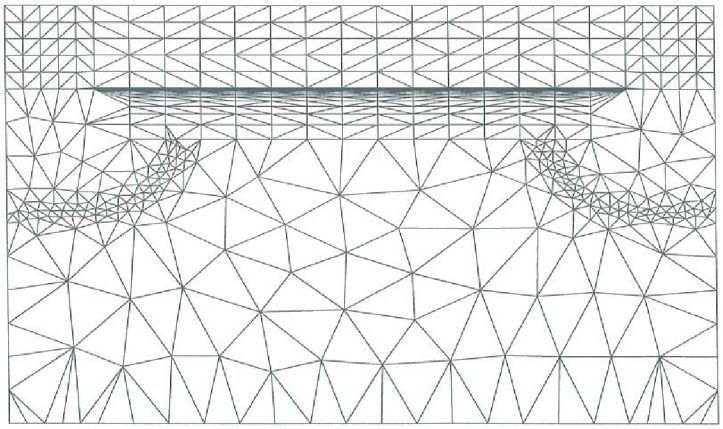

Figure 13. Grid after adaptive elliptic smoothing with Lawson flips.

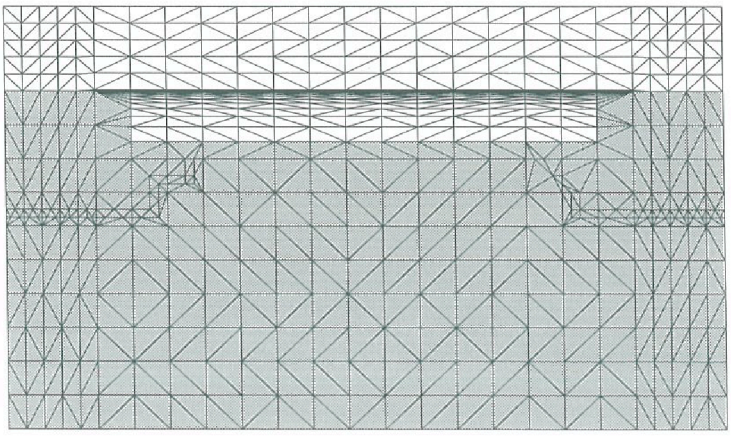

Figure 14. Grid after adaptive refinement. Shaded region is targeted for smoothing.

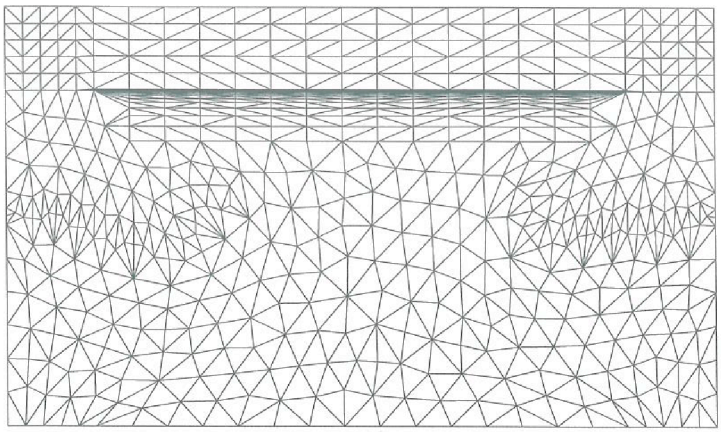

Figure 15. Grid after refinement followed by elliptic smoothing.

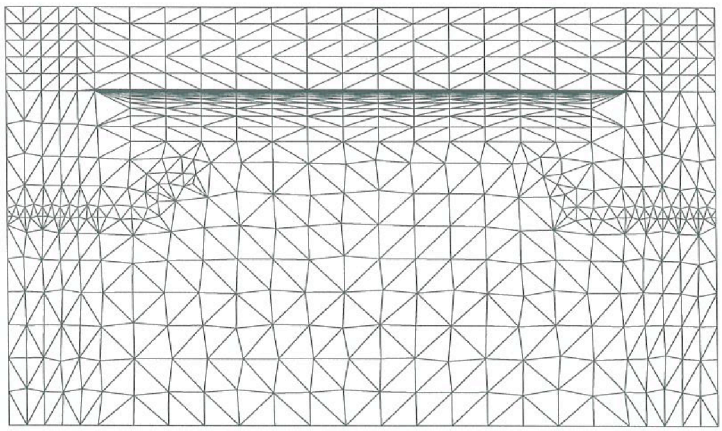

Figure 16. Grid after refinement and controlled elliptic smoothing.

REFERENCES

1. C. W. Mastin, 'Elliptic Grid Generation and Conformal Mapping', in *Mathematical Aspects of Numerical Grid Generation*, edited by Jose E. Castillo, SIAM, 1991.
2. C.W. Mastin and J.F. Thompson, 'Quasiconformal Mappings and Grid Generation', *SIAM J. Sci. Stat. Comput.* **5**, 305–310 (1984).
3. A. M. Winslow, 'Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh', *J. Comput. Phys.* **2**, 149–172 (1967).
4. C. L. Lawson, 'Software for C^1 Surface Interpolation', in *Mathematical Software III*, edited by John R. Rice, Academic Press, 1977.
5. R. E. Bank and R. K. Smith, 'Mesh smoothing using a posteriori error estimates', *submitted for publication.*
6. M. G. Vallet, F. Hecht, B. Mantel, 'Anisotropic Control of Mesh Generation based upon a Voronoi Type Method', in *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, edited by A. S.-Arcilla et al., North-Holland, 1991.
7. D. A. Field, 'Laplacian smoothing and Delaunay triangulation', *Commun. Appl. Numer. Methods* **4**, 709–712 (1988).