Anisotropic Mesh Refinement for the Simulation of Three-Dimensional Semiconductor Manufacturing Processes

Wilfried Wessner, Johann Cervenka, Clemens Heitzinger, Andreas Hössinger, and Siegfried Selberherr, *Fellow, IEEE*

***Abstract*—This paper presents an anisotropic adaptation strategy for three-dimensional unstructured tetrahedral meshes, which allows us to produce thin mostly anisotropic layers at the outside margin, i.e., the skin of an arbitrary meshed simulation domain. An essential task for any modern algorithm in the finite-element solution of partial differential equations, especially in the field of semiconductor process and device simulation, the major application is to provide appropriate resolution of the partial discretization mesh. The start-up conditions for semiconductor process and device simulations claim an initial mesh preparation that is performed by so-called Laplace refinement. The basic idea is to solve Laplace's equation on an initial coarse mesh with Dirichlet boundary conditions. Afterward, the gradient field is used to form an anisotropic metric that allows to refine the initial mesh based on tetrahedral bisection.**

***Index Terms*—Anisotropy, mesh refinement, tetrahedral bisection, tetrahedral meshes.**

I. Motivation

BEGINNING in the mid-1970s, aggressive efforts were undertaken to scale down metal–oxide–semiconductor (MOS) devices. Transistor dimensions soon reached a point at which first-order assumptions about physical effects and dopant distributions began to break down. An intimate coupling of process conditions, device behavior, and circuit performance to a degree previously unknown was observed. This situation prompted the introduction of computer simulation into the design process. Many important interrelated process and device effects were identified by means of simulation tools. Computer simulation of the fabrication process of semiconductor devices is attractive because of its rapid turn-around time and lower cost when compared to experimental approaches. Comparably inexpensive "computer experiments" can be made, instead of performing actual experiments by fabricating test structures. Technology computer-aided design (TCAD) programs for simulating the fabrication process and the electrical behavior of transistors are now well accepted in the semiconductor industry. The individual processing steps involved in the fabrication of integrated circuits (IC) can be arranged in four groups [1]:

1) pattern definition: lithography;
2) pattern transfer: etching;
3) layer formation: oxidation, deposition;
4) layer modification: diffusion, ion implantation.

Our in-house process simulator FEDOS [2] deals with two major process steps, namely, diffusion and oxidation. Diffusion, in the sense of an IC processing step, refers to the controlled migration of dopants within the substrate or adjacent material. Dopants typically diffuse both vertically and laterally from the surface at comparable rates. Dopants are introduced mainly by ion implantation, whereby dopant atoms are accelerated to selectively bombard the substrate so that they actually become lodged inside the substrate near the surface.

Oxidation is a process whereby oxygen or H_2O molecules from a gas above the substrate cause the growth of an oxide on the surface. When applied to silicon substrates, the oxidation process produces silicon dioxide.

Diffusion, oxidation, and ion implantation are directional and near-surface process steps. This means that the interesting simulation region is near the surface and therefore it is important to guarantee good spatial resolution at the skin of the simulation domain.

This can be efficiently achieved only by use of anisotropic meshes. Strict isotropic three-dimensional (3-D) regular meshes are impractical for realistic structures due to the resolution required as compared to the size of the simulation domain [3], [4]. The demand on calculation time and the limitation of memory call for anisotropic appropriately adapted meshes.

We first want to give a short introduction as to what is meant by the term anisotropy, with emphasis on a mathematical formulation of a tensor-based function that describes anisotropy as a distributed quantity. After an explanation of the so-called tetrahedral bisection refinement method in Section III, we are well prepared to have a close look on how Laplace's equation can help us define a tensor-based anisotropy suitable for refinement demands. The paper is concluded with a section about representative examples.

II. Anisotropy

As applied in [5], the element shapes are controlled by a tensor-based metric space for representing mesh anisotropy over the domain.

Anisotropy is defined by three orthogonal principal directions and an aspect ratio in each direction. The three principal directions are represented by three orthogonal unit vectors $\vec{\xi}$, $\vec{\eta}$, and $\vec{\zeta}$.

Manuscript received March 29, 2005; revised September 13, 2005. This paper was recommended by Associate Editor W. Schoenmaker.

The authors are with the Institute for Microelectronics, Technical University Vienna, A-1040 Wien, Austria (e-mail: wessner@iue.tuwein.nc.at).

Digital Object Identifier 10.1109/TCAD.2005.862750

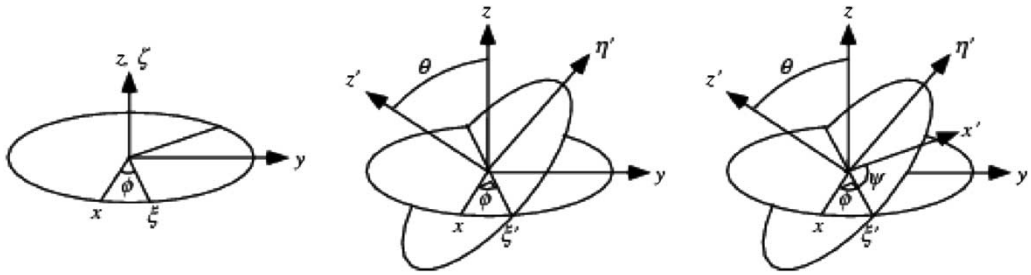

Fig. 1. Definition of Euler angles (ϕ, μ, ψ) in so-called x-convention rotation scheme according to rotation components given in (2).

According to Euler's Rotation theorem, any rotation can be described using three angles. If the rotations are written in terms of rotation matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, then a general rotation $\mathbf{R}$ can be written as

$$\mathbf{R} = \mathbf{ABC}. \tag{1}$$

The three angles giving the three rotation matrices are called Euler angles. There are several conventions of Euler angles, depending on the axes about which the rotations are carried out. The so-called x-convention (see Fig. 1) is the most common definition. In this convention, the rotation is given by Euler angles (ϕ, θ, ψ), where the first rotation is by an angle ϕ about the z axis, the second is by an angle $\theta \in [0, \pi]$ about the x axis, and the third is by an angle ψ about the z axis (again).

In the x-convention, the component rotations are then given by

$$\mathbf{A} := \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$\mathbf{B} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}$$
$$\mathbf{C} := \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

Hence, the general 3×3 rotation matrix $\mathbf{R}$ is given by

$$\begin{aligned} r_{11} &= \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi \\ r_{12} &= \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi \\ r_{13} &= \sin\psi\sin\theta \\ r_{21} &= -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi \\ r_{22} &= -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi \\ r_{23} &= \cos\psi\sin\theta \\ r_{31} &= \sin\theta\sin\psi \\ r_{32} &= -\sin\theta\sin\psi \\ r_{33} &= \cos\theta. \end{aligned} \tag{3}$$

The idea used in our approach is to apply a combination of rotation and dilation to define an anisotropic metric. The dilation is represented by three scalar values λ_ξ, λ_η, and λ_ζ, respectively. Using $(\vec{\xi}, \vec{\eta}, \vec{\zeta})$ and $(\lambda_\xi, \lambda_\eta, \lambda_\zeta)$, we define two matrices

$$\mathbf{R} := \begin{pmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{pmatrix} \quad \text{and} \quad \mathbf{S} := \begin{pmatrix} \lambda_\xi & 0 & 0 \\ 0 & \lambda_\eta & 0 \\ 0 & 0 & \lambda_\zeta \end{pmatrix}. \tag{4}$$

Combining matrices $\mathbf{R}$ and $\mathbf{S}$ gives a 3×3 positive definite matrix

$$\mathbf{M} := \mathbf{RSR}^T \tag{5}$$

which describes 3-D anisotropy. In our implementation, the anisotropic metric $\mathbf{M}$ is a tensor function that varies over the domain $\mathbf{M} = \mathbf{M}(x, y, z)$. The tensor function is symmetric and positive definite, which allows to use this tensor as a metric tensor. Very roughly, the metric tensor $\mathbf{m}_{ij}$ determines how to compute the distance between any two points in a given space. Its components can be viewed as multiplication factors that must be placed in front of the differential displacements dx_i in a generalized Pythagorean theorem $ds^2 = m_{11}dx_1^2 + m_{12}dx_1dx_2 + m_{22}dx_2^2 + \cdots$.

In Euclidean space, $m_{ij} = \delta_{ij}$, where δ is the Kronecker delta (which is 0 for $i \neq j$ and 1 for $i = j$), reproducing the usual form of the Pythagorean theorem $ds^2 = dx_1^2 + dx_2^2 + \cdots$.

A set S with a global distance function (the metric g) that for every two points x, y in S gives the distance between them as a nonnegative real number $g(x, y)$ is called a metric space [6]. The distance function must also satisfy

$$\begin{aligned} g(x, y) = 0 &\Leftrightarrow x = y \\ g(x, y) &= g(y, x) \\ g(x, y) + g(y, z) &\geq g(x, z). \end{aligned} \tag{6}$$

Applying this metric to tetrahedral meshes means that calculating the length of an edge of an element in a metric space can be seen as calculating a line integral. An arc length ℓ_C is defined as the length along a curve C : $\ell_C = \int_C ds$. The length of a line segment $\overline{PQ}$ in a metric space is calculated by

$$\ell_{PQ} = \int_0^1 \sqrt{\overline{PQ}^T \mathbf{M}(P + t\overline{PQ})\overline{PQ}}\, dt \tag{7}$$

where $\mathbf{M}(P + t\overline{PQ})$ is the metric at point $P + t\overline{PQ}$, $t \in [0, 1]$.

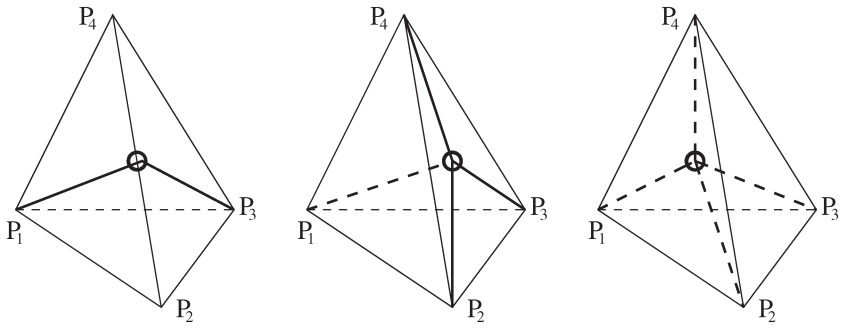

Fig. 2. Local 3-D simplex partitioning.

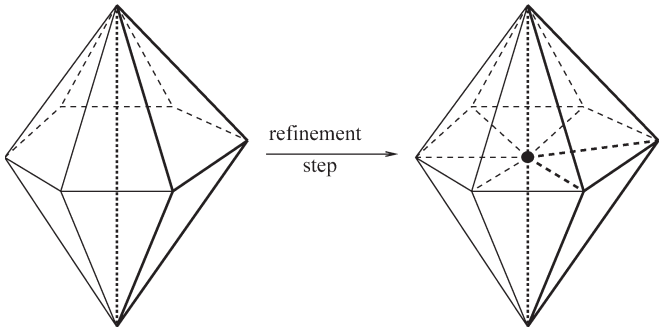

Fig. 3. Tetrahedral bisection influences whole refinement edge batch.

III. Mesh Adaptation Methods

In general, there are three mesh adaptation methods, namely, r-method, h-method, and p-method [7].

Using the r-method, the mesh connectivity is unchanged. Instead, node relocation is used to move the mesh nodes either by means of a weighted barycentric smoothing based on the location and the weight of the nodes in some neighborhood, or by means of element distortion. The criteria (weights) governing these operations are obtained by analyzing the current solution.

The h-method adaptation is defined in terms of local or global mesh enrichment by means of refining (by partitioning) or coarsening selected elements or all the elements in a mesh.

The p-method approach is based on an invariant mesh (in terms of points (nodes) and elements) and adjusts the degree (in terms of the interpolation functions) of the finite elements constructed on the mesh elements as a function of the current solution analysis.

To achieve a local partitioning that is required for the h-method, three configurations of inserting a new point (node) on the 3-D simplex are possible. It is possible to define a new point along an edge, on a face, or inside the tetrahedron (see Fig. 2).

In our implementation, we use a special h-method adaptation in which only mesh refinement by inserting a point on an edge is allowed, which is well investigated [8]–[10]. When bisecting a tetrahedron, a particular edge—called the refinement edge—is selected and split into two edges by a new vertex (cf., Fig. 3).

As new tetrahedra are constructed by refinement, their refinement edges must be carefully selected to take anisotropy into account without producing degenerately shaped elements. Therefore, one always has to keep in mind that bisecting a particular edge always influences the whole batch. To avoid ill-shaped elements, the longest edge of the refinement tetrahedron is used as refinement edge [11]. It is obvious that the longest edge of one tetrahedron is not necessarily the longest edge of all attached tetrahedra. To tackle this problem, all new tetrahedra are directly tested and included into the refinement procedure.

A recursive approach for local mesh refinement, which was suggested, e.g., in [12], cannot be applied in our situation due to the fact that anisotropy has to be taken into account.

To embrace anisotropy, the basic idea of our refinement strategy is to calculate the length of an edge in a certain metric space [13], i.e., the strain of the space varies from point to point with the consequence that the length of an edge depends on its position in space. The edge with the longest anisotropic edge length is cut in the middle.

IV. Surface Distance Transform

As mentioned in Section I, diffusion, oxidation, and ion implantation are directional and near-surface process steps. To set up an accurate simulation of these processes, it is essential to provide a dense mesh of the simulation domain, especially in near-surface regions, including the surface itself. To determine such regions, one has to compute the Euclidean distance transform, also known as distance map or distance field, of the entire simulation domain, which can be seen as a solution of the Eikonal equation $|\nabla u| = 1$. In [14], an algorithm is presented in which the Eikonal equation is solved by the method of characteristics. The drawback of this promising method is the need of a special representation of vertex polyhedra, which increases the memory demand dramatically.

Due to the fact that our simulator FEDOS has an adaptable finite-element kernel that allows to use arbitrary boundary conditions in a very flexible way, we decided to solve Laplace's equation as an approximation for the surface distance function. This approach is, compared to the pure calculation of the distance transform, more flexible and can also handle multiple segment domains with appropriate interface conditions.

The following section gives a detailed description of the surface distance approximation concept with the help of Laplace's equation demonstrated on a simple physical device—the plate capacitor. The section is closed by a mathematical formulation of the gradient field based on Laplace's equation solution, which is used later on for the derivation of the anisotropy metric definition function.

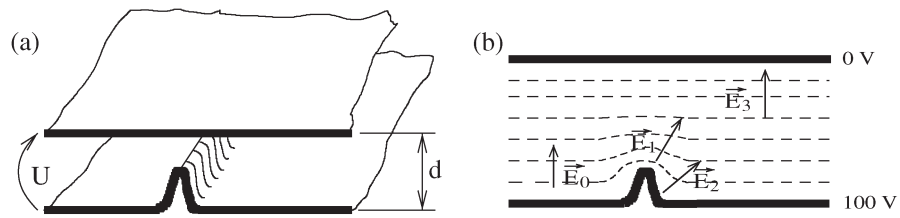

Fig. 4. (a) Typical plate capacitor structure. Two coplanar metal planes are connected to voltage supply. Lower plane is riddled with a bead. (b) Resulting electrostatic field. Electrical field intensity is vector quantity that is perpendicular to iso-surfaces of electrostatic potential ψ.

A. Laplace Equation

The scalar form of Laplace's equation is the partial differential equation

$$\nabla^2 \psi = 0. \tag{8}$$

A function ψ that satisfies Laplace's equation is said to be harmonic. Solutions have no local maxima or minima. Because Laplace's equation is linear, the superposition of any two solutions is also a solution. A solution to Laplace's equation is uniquely determined if appropriate boundary conditions are posed [15].

B. Plate Capacitor

Our idea is to use the solution of Laplace's equation as an approximation for a surface distance function. The imagination is based on electrostatic field calculations of the plate capacitor [16] (see Fig. 4). A typical plate capacitor structure is formed by two coplanar metal planes that are connected to a voltage supply. We neglect the surrounding area by applying homogenous Neumann boundary conditions at open borders of the capacitor and Dirichlet boundary conditions at the electrodes (assumes an infinitely expanded capacitor).

Iso-surfaces of the electrostatic potential inside the plate capacitor also form coplanar planes that can be used as a measure for the perpendicular distance to the surface. This measure is exact if and only if the plates are real coplanar. For nonplanar structures, the electrostatic potential is only an approximation for the surface distance [see Fig. 4(b)].

In technical terms, the electrical field $\vec{E}$ can be written as a gradient field of the electrostatic potential ψ as

$$\vec{E} = -\nabla \psi. \tag{9}$$

C. Gradient Field

The gradient $\nabla \psi = \mathrm{grad}(\psi)$ of a scalar field $\psi = \psi(x, y, z)$ in Cartesian coordinates is given by

$$\nabla \psi = \frac{\partial \psi(x,y,z)}{\partial x}\vec{i} + \frac{\partial \psi(x,y,z)}{\partial y}\vec{j} + \frac{\partial \psi(x,y,z)}{\partial z}\vec{k}. \tag{10}$$

We are looking for (10) expressed through local coordinates on the 3-D unit simplex T. The gradient can be calculated by using linear basis functions [17] applied to the 3-D unit simplex. The coordinate transformation

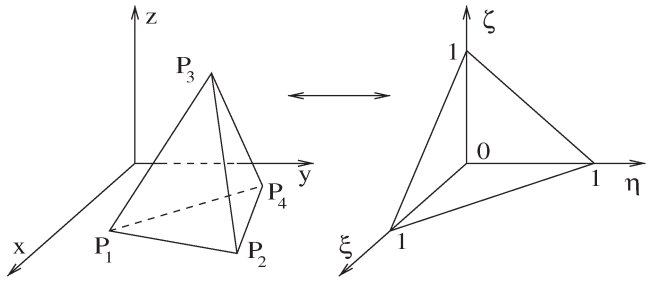

Fig. 5. Coordinate transformation from arbitrary tetrahedron to unit simplex.

$$\begin{aligned} x &= x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta + (x_4 - x_1)\zeta \\ y &= y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta + (y_4 - y_1)\zeta \\ z &= z_1 + (z_2 - z_1)\xi + (z_3 - z_1)\eta + (z_4 - z_1)\zeta \end{aligned} \tag{11}$$

allows the mapping of an arbitrary tetrahedron at global coordinates (x, y, z) to the unit simplex T (cf. Fig. 5) with local element coordinates (ξ, η, ζ). In matrix notation, this can be written as

$$\vec{r} - \vec{r_1} = \mathbf{J} \cdot \vec{\delta} \tag{12}$$

where $\vec{r} = (x, y, z)^T$, $\vec{r_1} = (x_1, y_1, z_1)$, $\vec{\delta} = (\xi, \eta, \zeta)^T$, and $\mathbf{J}$ denotes the Jacobian matrix that when applied to (11) results in

$$\mathbf{J} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{pmatrix}. \tag{13}$$

Using linear basis functions on the 3-D unit simplex [18] allows a linear approximation of the scalar field over the element in the form

$$\psi_T(\xi, \eta, \zeta) = \sum_{k=1}^{4} N_k(\xi, \eta, \zeta)\psi_k \tag{14}$$

where ψ_k denotes the scalar value of the solution on vertex k of the 3-D unit simplex T.

Applying (10) to the linear approximation, given by (14), results in

$$\nabla \psi_T(\xi, \eta, \zeta) = \begin{pmatrix} -\psi_1 + \psi_2 \\ -\psi_1 + \psi_3 \\ -\psi_1 + \psi_4 \end{pmatrix} \tag{15}$$

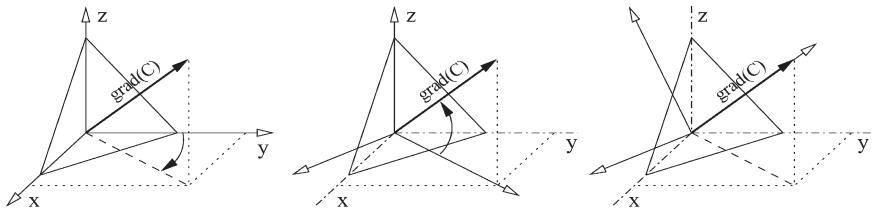

Fig. 6. Rotation evolution for the anisotropy function at the unit simplex.

for the gradient. Using the inverse of the transposed Jacobian matrix, the gradient in global coordinates can now be expressed by

$$\nabla\psi_T(x,y,z) = (\mathbf{J}^T)^{-1}\nabla\psi_T(\xi,\eta,\zeta). \tag{16}$$

It is in the nature of this approach that the gradient $\nabla\psi_T(x,y,z)$ (16) is constant over an element T and forms a piecewise constant gradient field that gives a granular approximation of the proper gradient field given by (10).

V. Metric Function

For the construction of the metric tensor function, we first calculate the solution of Laplace's equation under consideration of appropriate given Dirichlet boundary conditions on the initial coarse mesh. This approach allows us to define in a very flexible way where the refinement should take place.

To take anisotropy into account, we use the derivative of the electrostatic potential ψ [see (9)] as the primary stretching direction for the anisotropic metric description (5).

To accomplish this task, we first rotate the three axes of the Cartesian coordinate system (x, y, and z axes) so that the new y axis is parallel to the gradient vector $\nabla\psi$. The evolution of this rotation is illustrated in Fig. 6. The rotation matrix can be easily calculated according to Euler rotation angles (see Section II). At the second step, we apply a dilation factor function

$$\lambda_\eta = \lambda_\eta(\psi). \tag{17}$$

Thus, the dilation along the gradient direction depends on the potential ψ. Preliminary experiments showed that a Gaussian distribution (18) is a good choice for a smooth and well-adjustable dilation factor function.

A normal distribution in a variate X with mean μ and variance σ^2 is a statistic distribution with probability function

$$P(X) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-(X-\mu)^2}{2\sigma^2}\right) \tag{18}$$

on the domain $X \in (-\infty, \infty)$.

Fig. 7 shows a typical dilation function in which a belt of approximately 20% from the maximum of the electrostatic potential is influenced by the dilation. This means that in regions where the electrostatic potential drops beyond 80% of the maximum, no refinement takes place. Since the solution of the Laplace equation is said to be harmonic, the minimum and the maximum of the solution are located on the border of the simulation domain (Dirichlet boundary conditions). Due to the special choice for the dilation function, also an element grading is adjusted, so that surface-near elements (within the 20% belt and close to 100%) are dilated more, which means stronger refinement.

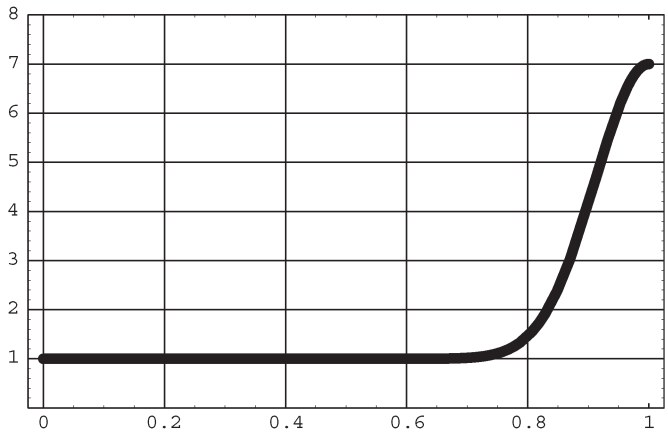

Fig. 7. Dilation function $f(\psi)$ for stretching parameter λ_η according to (17).

All other stretching weights, which are not gradient directional, are set to unity. This guarantees a dilation only along the gradient field. According to (5), the anisotropic metric function is now completely specified over the element. For 3-D simplex partitioning, the anisotropic lengths of all tetrahedron edges are calculated under consideration of (7). The longest anisotropic edge that transcends the maximum edge length value is chosen as the refinement edge and cut at the middle. This procedure is repeated until no edge of the whole simulation domain is longer than the maximum edge length value.

VI. Examples

In this section, we present four refinement examples. The first one is a propaedeutic one that shows the functionality on an eighth of sphere. The second is an oxidation simulation example, which explains the underlying refinement strategy in detail. The others are from a real simulation cycle that mirrors practical usage.

A. Propaedeutic Example

To illustrate how our Laplace refinement strategy works, we start with an elementary example in which primarily element grading, without anisotropy, plays a role. The task is to provide a dense isotropic mesh in an area around the tip of an eighth of a sphere, all other regions should be mostly filled with an coarse mesh. First, we start with the initial coarse mesh provided by

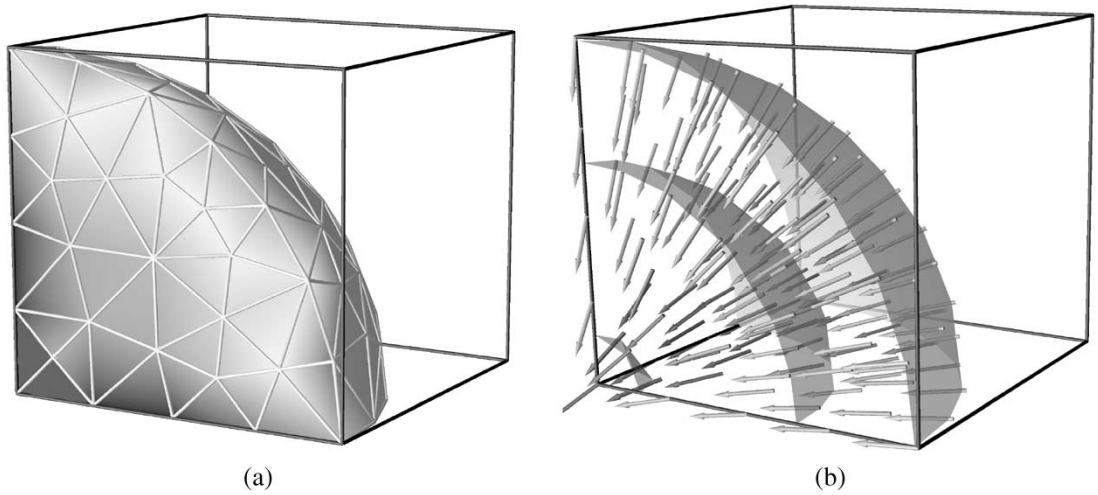

Fig. 8. Left figure shows initial mesh of an eighth of sphere. Appropriate boundary conditions are applied where spike is set to unity and outer rounded hull is grounded. All other boundaries are supplied with zero Neumann boundary conditions. Solution of Laplace equation is carried out and iso-surfaces and gradient field are depicted in (b). (a) Initial coarse mesh of eighth of sphere with 151 points and 561 tetrahedra. Gray-scale coloration shows solution of Laplace equation. (b) Iso-surfaces of Laplace's equation solution and according to gradient field vectors on initial coarse mesh.

an arbitrary mesh generator, e.g., our in-house product DELINK [19], shown in Fig. 8(a).

Due to the fact that we need a fine mesh around the tip and a coarse mesh on the outer rounded hull of the sphere, we apply Dirichlet boundary conditions so that the tip is set to unity and the outer rounded hull is grounded. All other boundaries are supplied with zero Neumann boundary conditions. Endowed with this setup, we now calculate the solution of Laplace's equation on the initial coarse mesh and compute the gradient field as well, which is depicted in Fig. 8(b). Applying the dilation-function, cf. Section V, shown in Fig. 7 to all stretching directions $\lambda_\xi, \lambda_\eta$, and λ_ζ, cf. (4), causes an isotropic dilation in all directions by the same amount and therefore isotropic refinement in this region. In this particulare case the isotropic refinement procedure result is shown in Fig. 9.

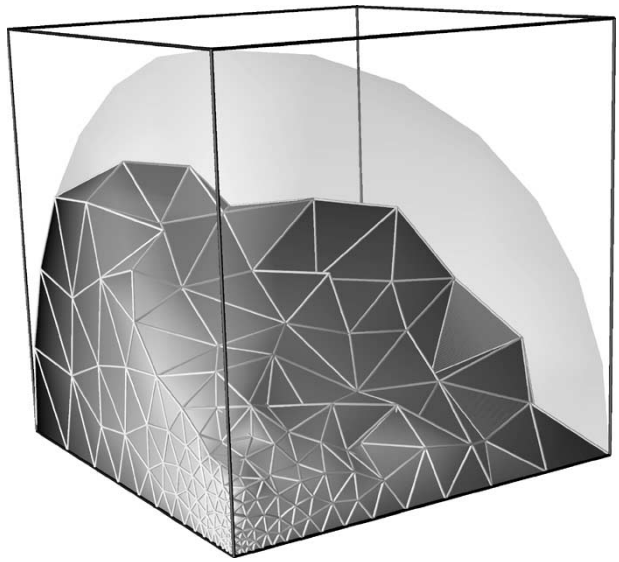

Fig. 9. Refined sphere part with 441 points and 1801 tetrahedra. The upper part is cut away to see the interior of the solid.

B. Demonstrative Oxidation Example

At the beginning of the simulation procedure, it is important to deliver a high spatial resolution in a small layer beneath the oxidant exposed surface Fig. 10. This can be achieved by our Laplace refinement method. Fig. 10(a) shows the initial coarse mesh that was generated with our in-house mesh generator Laygrid [20], [21]. We used as starting conditions a cubic silicon (Si) structure with a mostly regular mesh. One part of the cube is covered by an L-shaped silicon nitride (Si_3N_4) mask.

The Dirichlet boundary conditions where chosen as follows. The upper surface of Si, which is exposed to an oxidizing atmosphere, is set to unity and the opposite part of the Si body is set to zero. Fig. 10(b) shows the corresponding iso-surfaces and the gradient field of Laplace's equation solution. For the dilation function, (17), cf. Fig. 7, was used. The function approaches unity for ψ values smaller than 0.75, so refinement should mostly cover only regions where ψ is larger than 0.75. For higher ψ values, the dilation rises, so an element grading takes place from a very small edge length near the surface to the initial coarse mesh toward the interior.

The result of our refinement strategy applied to the demonstrative example is shown in Fig. 11. The refinement covered a well-defined layer along the surface. Anisotropic element shapes are achieved so that small point distances are obtained perpendicular and large edge lengths appear parallel to the surface. Fig. 11(b) shows a cross-section of the refined structure. The refinement covers exactly the dilation function given in Fig. 7. Regions with a ψ value smaller than 0.75 are untouched and so the original mesh could be kept, which guarantees usage of a (nearly) minimal number of grid points.

Under the mask (see Fig. 10), the elements are slightly influenced by the refinement. This is obvious due to the fact that for the elementary refinement a tetrahedral bisection was used. So inserting a new point on an edge refines the whole batch connected to this edge (see Fig. 3).

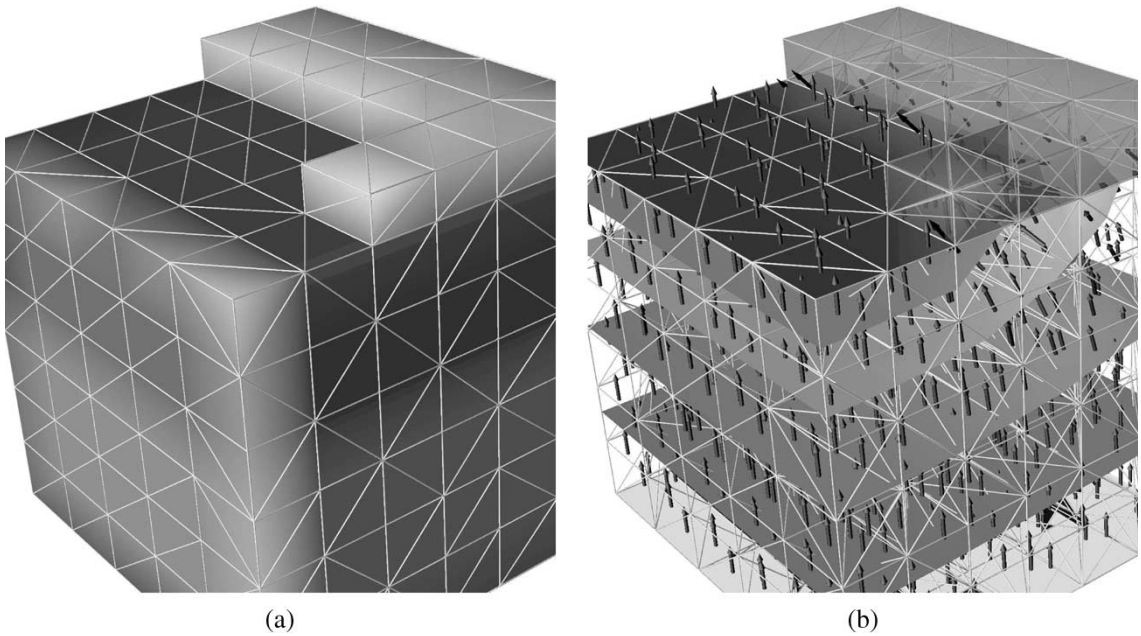

Fig. 10. (a) Initial mesh that was generated with Laygrid. Cubic silicon (Si) body with L-shaped silicon nitride (Si_3Ni_4) mask on top. Initial, mostly coarse regular mesh, 316 points and 1152 tetrahedra. (b) Appropriate boundary conditions are applied and solution of Laplace equation is carried out. Iso-surfaces and gradient field are depicted. Iso-surfaces of Laplace's equation solution and according gradient field vectors on initial coarse mesh.

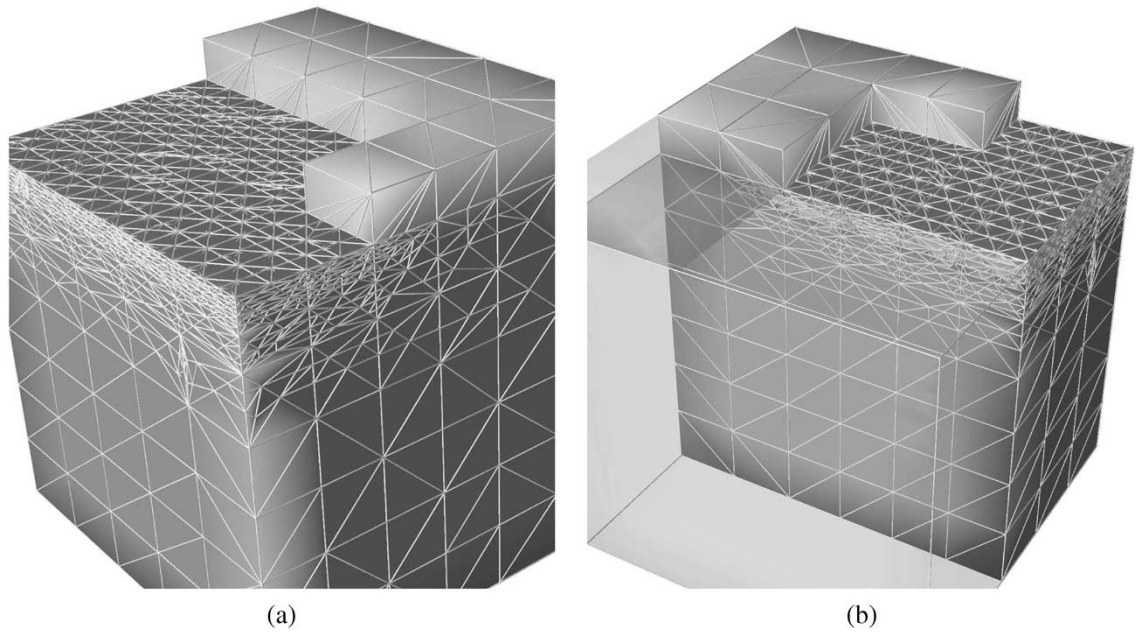

Fig. 11. Cubic silicon (Si) body with L-shaped silicon nitride (Si_3Ni_4) mask on top after anisotropic refinement step. Thin anisotropic surface-near mesh layer was generated and used as input for an oxidation simulation. (a) Highly anisotropic thin mesh layer after Laplace refinement in upper region of silicon body, 10 712 points and 55 438 tetrahedra. (b) Cross section of anisotropic refined mesh structure.

The resulting refined structure forms the initial mesh input for oxidation simulation. The L-shaped silicon nitride (Si_3N_4) mask blocks the oxygen and therefore only the silicon body (Si) is influenced. During the transient simulation of oxidation, a conversion from silicon (Si) to silicon dioxide (SiO_2) takes place [22], [23]. This reaction consumes silicon and the newly formed silicon dioxide has more than twice the volume of the original silicon, so there is a strong volume deformation of the initial crystal structure that yields nonplanar surfaces. Fig. 12 shows the results of the oxidation simulation.

C. Nonplanar Surface

For a typical semiconductor process step, the surfaces of the structure are nonplanar [24]. This example demonstrates that our refinement strategy also works for curved surfaces.

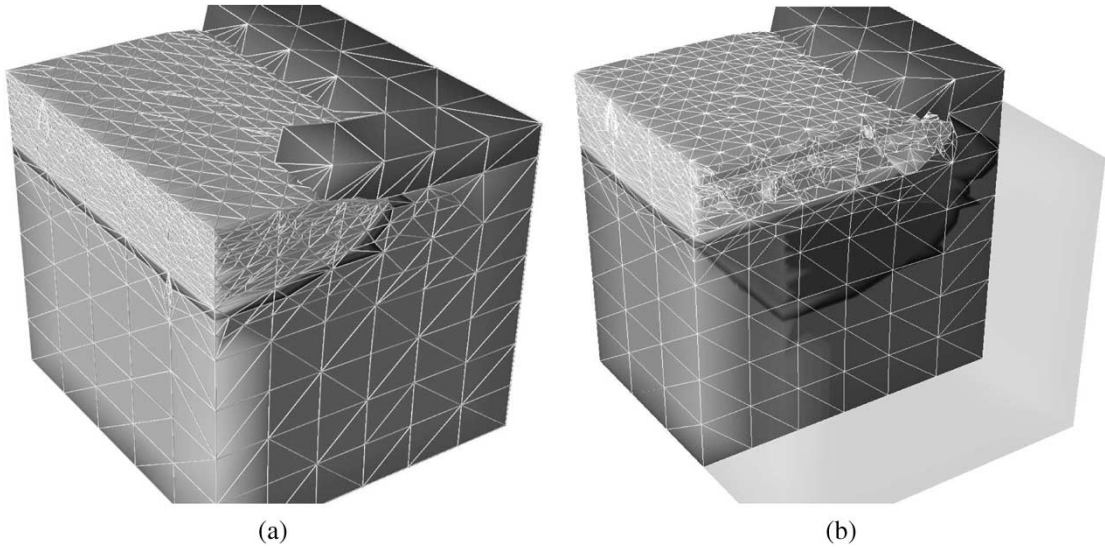

Fig. 12. Result of oxidation simulation example. For initial mesh, anisotropic refined structure was used. (a) Oxidation simulation result. (b) Cross section of oxidation simulation structure.

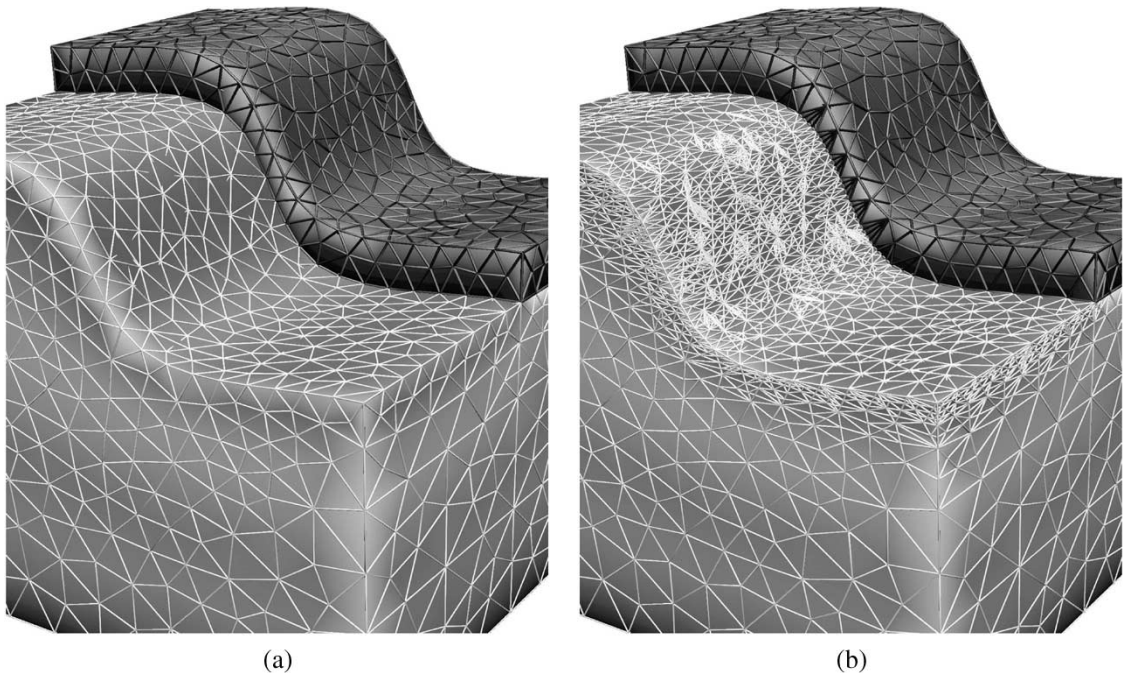

Fig. 13. Nonplanar surface example for anisotropic refinement. Refinement takes place only in well-defined layer beneath surface. Contour in middle of structure shows one iso-surface of Laplace equation solution. (a) Initial, mostly regular mesh, 3884 points and 20 534 tetrahedra. (b) Mesh after anisotropic refinement, 14 490 points and 72 523 tetrahedra.

Fig. 13(a) shows the initial regular coarse mesh. The exposed surface is again covered by a mask. For Dirichlet boundary conditions and for the dilation function, we used the same assumptions as in the first example (see Section VI-B).

The resulting mesh after refinement can be seen in Fig. 13(b). Due to the fact that the solution of Laplace's equation is only an approximation for the surface distance function, the refinement layer covers surface-near regions with different thickness, which is acceptable for our simulation tasks.

D. EEPROM Cell

This example shows an anisotropic refinement result of the simulation domain of an EEPROM cell, which was part of a full manufacturing cycle simulation [25]. Fig. 14 shows the initial coarse and the refined mesh (one quarter of the cell is shown). This was required in preparation for an oxidation process that forms the isolation between the two gates in the EEPROM cell. Therefore, an anisotropic refinement is required in the active

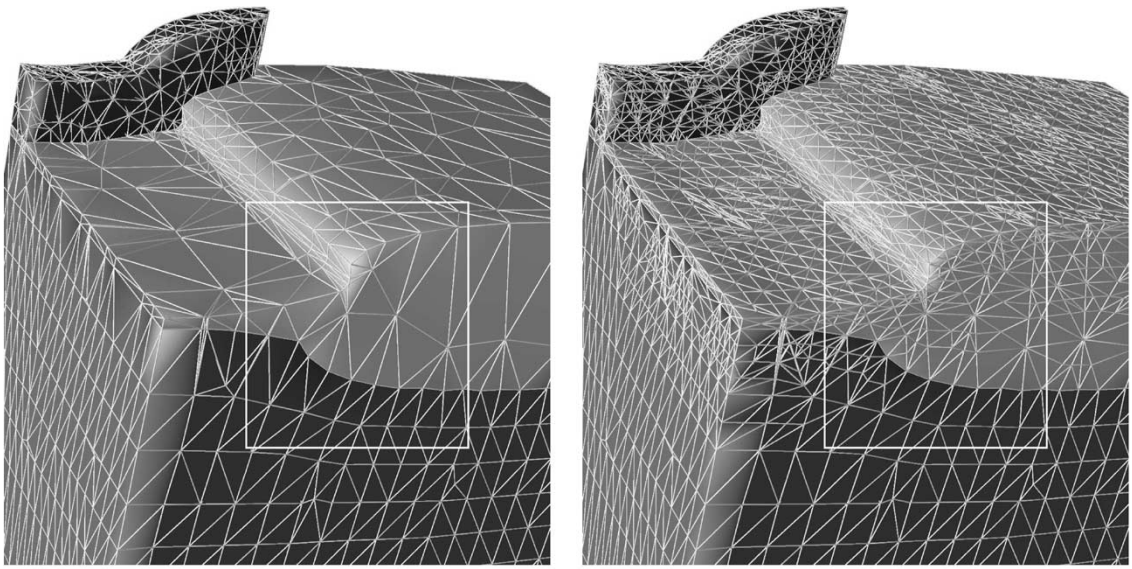

Fig. 14. One quarter of nonvolatile memory cell (EEPROM). Refinement covers top part of structure. White rectangles mark zooming region. (a) Initial unstructured mesh, 10 872 points and 39 142 tetrahedra. (b) Mesh after anisotropic refinement, 18 439 points and 58 922 tetrahedra.

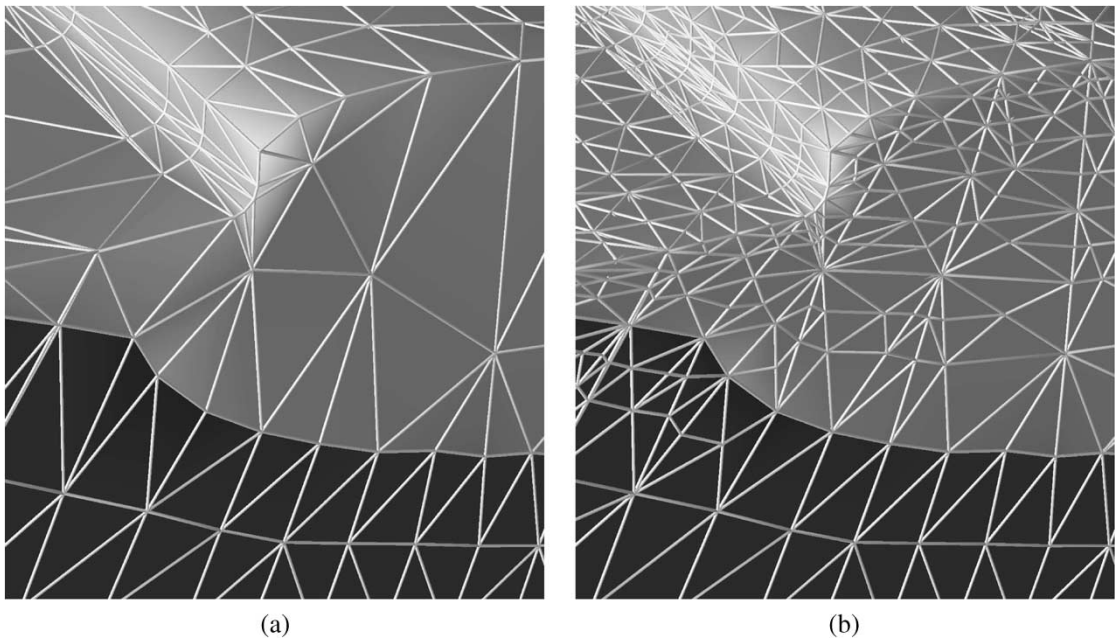

Fig. 15. Detail of nonvolatile memory cell (EEPROM) mesh before and after anisotropic refinement (see also Fig. 14). (a) Initial unstructured mesh. (b) Resulting mesh after anisotropic refinement.

area of the field oxide and along the floating gate, which are exposed to the oxidation.

Fig. 15 shows a detailed view of the initial coarse and the refined mesh structure whereby the initial mesh as part of the refined mesh can be clearly observed.

VII. Conclusion

The generation of small, strongly anisotropic, and unstructured mesh layers by 3-D mesh generators is, unfortunately from a technology point of view, still something of an art, as well as a science. A more robust approach, compared to anisotropic mesh generation, is to generate mostly isotropic coarse initial meshes, for instance, with a Delaunay mesh generator, followed by a mesh adaptation postprocessing step on demand.

We presented an efficient anisotropic refinement strategy that allows local modifications of an arbitrary 3-D mesh. Laplace equation calculations with special Dirichlet boundary conditions provide useful approximations of the surface distance functions and are therefore a good choice for introducing small anisotropic layers on an initial mesh. The flexible use of boundary conditions enables the application of our refinement strategy for arbitrary complex 3-D structures.

Practical usage of our refinement strategy was shown on the most challenging process simulation step, namely, 3-D oxidation. As a basic refinement step, tetrahedral bisection is used, which guarantees safe elementary partitioning and a valid mesh at every time of the refinement cycle.

ACKNOWLEDGMENT

The authors would like to thank R. Sabelka, S. Wagner, and H. Ceric for assistance during the preparation of this document and P. Fleischmann for fruitful discussions.

REFERENCES

[1] G. Schmicki and P. Seegebrecht, *Prozeßtechnologie*. New York: Springer-Verlag, 1991.
[2] C. Hollauer, S. Holzer, H. Ceric, S. Wagner, T. Grasser, and S. Selberherr, "Investigation of thermo-mechanical stress in modern interconnect layouts," in *Proc. 6th Int. Congress Thermal Stresses*, Vienna, Austria, 2005, vol. 2, pp. 637–640.
[3] P. J. Frey and P.-L. George, *Mesh Generation*, 1st ed. Oxford, U.K.: HERMES Science Europe Ltd., 2000.
[4] W. Wessner, C. Heitzinger, A. Hössinger, and S. Selberherr, "Error estimated driven anisotropic mesh refinement for three-dimensional diffusion simulation," in *Proc. Int. Conf. Simulation Semiconductor Processes Devices*, Boston, MA, 2003, pp. 109–112.
[5] S. Yamakawa and K. Shimada, "High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing," in *Proc. 9th IMR*, New Orleans, LA, 2000, pp. 263–273.
[6] L. Kelly, *The Geometry of Metric and Linear Spaces*. Heidelberg, Germany: Springer-Verlag, 1975.
[7] P.-L. George and H. Borouchaki, *Delaunay Triangulation and Meshing*, 1st ed. Paris, France: Editions HERMES, 1998.
[8] D. N. Arnold, A. Mukherjee, and L. Pouly, "Locally adapted tetrahedral meshes using bisection," *SIAM J. Sci. Comput.*, vol. 22, no. 2, pp. 431–448, 2000.
[9] J. F. Thompson, B. K. Soni, and N. P. Weatherill, *Handook of Grid Generation*, 1st ed. Boca Raton, FL: CRC, 1999.
[10] J. R. Shewchuk, "Delaunay refinement mesh generation," Ph.D. dissertation, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, May 1997. [Online]. Available: http://www.cs.berkeley.edu/~jrs/
[11] J. Cervenka, "Three-dimensional mesh generation for device and process simulation," Ph.D. dissertation, Inst. Micro., Tech. Univ. Vienna, Vienna, Austria, 2004. [Online]. Available: http://www.iue.tuwien.ac.at/
[12] I. Kossaczky, "A recursive approach to local mesh refinement in two and three dimensions," *J. Comput. Appl. Math.*, vol. 55, no. 3, pp. 275–288, Nov. 1994.
[13] S. Lo, "3D anisotropic mesh refinement in compliance with a general metric specification," *Finite Elem. Anal. Des.*, vol. 38, no. 1, pp. 3–19, Dec. 2001.
[14] S. Mauch, "A fast algorithm for computing the closest point and distance transform," California Inst. Tech., Pasadena, CA, pp. 1–17, Caltech ASCI Tech. Rep. 007, 2000. vol. caltechASCI/2000.077.
[15] S. G. Krantz, *Handbook of Complex Variables*. Boston, MA: Birkhäuser, 1999.
[16] A. Prechtl, *Vorlesungenüber die Grundlagen der Elektrotechnik*, vol. 1. New York: Springer-Verlag, 1994.
[17] O. Zienkiewicz and R. Taylor, *The Finite Element Method*, 4th ed, vol. 1. Berkshire, England: McGraw-Hill Book Company Europe, 1989.
[18] R. Bauer and S. Selberherr, "Calculating coupling capacitances of three-dimensional interconnections," in *Proc. Solid State and Integrated Circuit Technology Conf.*, Beijing, China, 1992, pp. 697–699.
[19] P. Fleischmann, R. Sabelka, A. Stach, R. Strasser, and S. Selberherr, "Grid generation for three-dimensional process and device simulation," in *Proc. Simulation Semiconductor Processes Devices*, Tokyo, Japan, 1996, pp. 161–166.
[20] R. Sabelka and S. Selberherr, "SAP—A program package for three-dimensional interconnect simulation," in *Proc. Int. Interconnect Technology Conf.*, San Francisco, CA, 1998, pp. 250–252.
[21] P. Fleischmann, "Mesh generation for technology CAD in three dimensions," Ph.D. dissertation, Inst. Micro., Tech. Univ. Vienna, Vienna, Austria, Dec. 1999. [Online]. Available: http://www.iue.tuwien.ac.at/
[22] B. E. Deal and A. Grove, "General relationship for the thermal oxidation of silicon," *J. Appl. Phys.*, vol. 36, no. 12, pp. 3770–3778, Dec. 1965.
[23] C. Hollauer, H. Ceric, and S. Selberherr, "Simulation of thermal oxidation: A three-dimensional finite element approach," in *Proc. ESSDERC*, Estoril, Portugal, 2003, pp. 383–386.
[24] S. M. Sze, *Semiconductor Devices, Physics and Technology*, 1st ed. New York: Wiley, 1985.
[25] A. Hössinger, R. Minixhofer, and S. Selberherr, "Full three-dimensional analysis of a non-volatile memory cell," in *Proc. Simulation Semiconductor Processes Devices*, Munich, Germany, 2004, pp. 129–132.

Wilfried Wessner was born in Horn, Austria, in 1977. He received the Diplom-Ingenieur degree (with honors) in computer engineering from Vienna University of Technology, Vienna, Austria, in 2002. He is currently working toward the Ph.D. degree at the same university.

He joined the Institute for Microelectronics, Vienna University of Technology, in the summer of 2002. His scientific interests include three-dimensional mesh generation, anisotropic mesh adaptation, computational geometry, and data visualization.

Johann Cervenka was born in Schwarzach, Austria, in 1968. He received the Diplom-Ingenieur degree (with honors) in electrical engineering and the Ph.D. degree in technical sciences from Vienna University of Technology, Vienna, Austria, in 1999 and 2004, respectively.

In November 1999, he joined the Institute for Microelectronics, Vienna University of Technology. His scientific interests include three-dimensional mesh generation as well as algorithms and data structures in computational geometry.

Clemens Heitzinger was born in Linz, Austria, on 1974. He received the Dipl.-Ing. degree (with honors) in technical mathematics and the Ph.D. degree (with honors) in technical sciences from Technical University of Vienna, Vienna, Austria, in 1999 and 2002, respectively.

In February 2000, he joined the Institute for Microelectronics, Technische Universität Wien. His research interests include applied mathematics for process and device simulation.

Dr. Heitzinger was awarded an Erwin Schrödinger Fellowship by the Austrian Science Fund (FWF) in January 2003.

Andreas Hössinger was born in St. Pölten, Austria, on 1969. He received the Dipl.-Ing. degree in technical physics from Technical University of Vienna, Vienna, Austria, in January 1996, and the Ph.D. degree in technical sciences from the Institute for Microelectronics, in September 2000.

In June 1996, he joined the Institute for Microelectronics, Technische Universität Wien. In 1998, he held a visiting research position at Sony, Atsugi, Japan. In 2001, he was a Visiting Researcher at LSI Logic, Santa Clara, CA, within the scope of a cooperate research project on three-dimensional process simulation. In 2001, he also received a grant from the Austrian Academy of Science within the scope of the Austrian Program for Advanced Research and Technology for his work on three-dimensional process simulation. He is currently a Post-Doctoral Researcher at the Institute for Microelectronics.

Siegfried Selberherr (M'79–SM'84–F'93) was born in Klosterneuburg, Austria, in 1955. He received the Diplom-Ingenieur degree in electrical engineering, the Ph.D. degree in technical sciences, and Venia Docendi degree in computer-aided design all from Vienna University of Technology, Vienna, Austria, in 1978, 1981, and 1984, respectively.

From 1988 to 1999, he was the Head of the Institute for Microelectronics, Vienna University of Technology. From 1998 to 2005, he served as the Dean with the Faculty of Electrical Engineering and Information Technology. His current research interests include analysis and simulation of problems for microelectronics engineering.