Generation of Unstructured Meshes for Process and Device Simulation by Means of Partial Differential Equations

Johann Cervenka, Wilfried Wessner, Elaf Al-Ani, Tibor Grasser, *Senior Member, IEEE*, and Siegfried Selberherr, *Fellow, IEEE*

***Abstract*—For process and device simulation, very high mesh densities are often required to obtain accurate simulation results. Unfortunately, the required mesh densities depend often on a direction. Conventional mesh-refinement strategies generate isotropic meshes with a high amount of mesh points, reaching the memory and time limits in particular for three-dimensional simulations. For a better resolution of the carrier concentrations, for instance, a boundary-conforming mesh-generation method with tunable mesh spacings in almost orthogonal directions was developed. Similar to elliptic mesh generation, the mesh points are placed inside the simulation regions based on the solution of partial differential equations. The method used can produce highly anisotropic mesh densities in the regions of particular interest. In contrast to elliptic grid generation, which produces structured grids, the method used generates triangular or tetrahedral (unstructured) Delaunay meshes in two or three dimensions, respectively, which are very well suitable for the process and device simulators.**

***Index Terms*—Anisotropy, mesh generation, microelectronics, optimization techniques, semiconductors, simulation.**

I. Introduction

FOR THE development of modern semiconductor devices, the use of accurate process and device simulators has been proven to be highly beneficial. Usually, the mathematical description of the problem cannot be solved analytically, and numerical approximations have to be employed [1], [2].

The simulation domain is tessellated and based on this mesh an approximation of the solution is found using the methods of Finite Elements or Finite Boxes; see for instance [3] and [4]. To achieve accurate numerical results, the simulation meshes have to be very dense in certain regions and coarse in others, which often demands anisotropy.

As a particularly important example, consider an MOS transistor where the carriers and the current in the channel under the gate oxide vary slightly along the channel, but perpendicularly to the semiconductor surface the quantities change over several orders of magnitude. To resolve the carrier concentrations properly, a dense mesh is required perpendicular to the surface, whereas along the surface, a coarser mesh is sufficient. Applying conventional mesh-refinement strategies, the resulting mesh often shows a too large number of mesh points along the channel or a too crude discretization of the carriers perpendicular to the channel.

Therefore, a new strategy of point placement inside the simulation domain was developed. In addition to this anisotropy, the method of Finite Boxes employed by most device simulators requires Delaunay meshes [5]. Therefore, the shape of the mesh elements cannot be chosen arbitrarily, because the Delaunay criterion must always be satisfied [6], [7]. Thus, highly sophisticated anisotropic mesh-refinement strategies, as introduced for instance in [8] and [9], can usually not be used, because these techniques do not account for the Delaunay criterion.

The basic idea of our new method is inspired by the well-known ortho-product grids. The grid densities of the ortho-product grids can be tuned arbitrarily in every direction of the rectangles or cuboids. However, this method cannot properly resolve nonplanar and nonaxial boundaries. The number of grid points increases rapidly when certain boundary regions have to be refined. Classical triangular meshes, on the other hand, are not limited by nonplanarities of the boundaries, but the mesh elements show a nearly isotropical mesh density and the result is not satisfying either. In our newly developed method, the benefits of both mesh types are combined.

II. Basic Idea

To demonstrate the basic algorithm, a two-dimensional domain is considered first. The extension to three dimensions is discussed in Section VII. In our partial differential equation (PDE) method, the simulation area is treated like a capacitor. Two electrodes are placed at the boundary (see Fig. 1) where a contact voltage is applied. For the remaining parts of the boundary, homogenous Neumann boundaries are used. Based on the electric field inside the capacitor, the mesh points and edges for the actual simulation mesh are placed along selected contour and field lines. As the contour and field lines stay orthogonal, the grid elements build quadrangles with nearly orthogonal edges, similar to the ortho grids. As a consequence of the electrode-conforming contour lines and Neumann boundary-conforming field lines, we obtain boundary-conforming grid

Manuscript received May 2, 2005. This paper was recommended by Associate Editor S. Saxena.

J. Cervenka and T. Grasser are with the Christian Doppler Laboratory for Technology Computer Aided Design (TCAD), Institute for Microelectronics, Vienna University of Technology, 1040 Vienna, Austria (e-mail: cervenka@iue.tuwien.ac.at; grasser@iue.tuwien.ac.at).

W. Wessner, E. Al-Ani, and S. Selberherr are with the Institute for Microelectronics, Vienna University of Technology, 1040 Vienna, Austria (e-mail: wessner@iue.tuwien.ac.at; alani@iue.tuwien.ac.at; selberherr@iue.tuwien.ac.at).

Digital Object Identifier 10.1109/TCAD.2006.876514

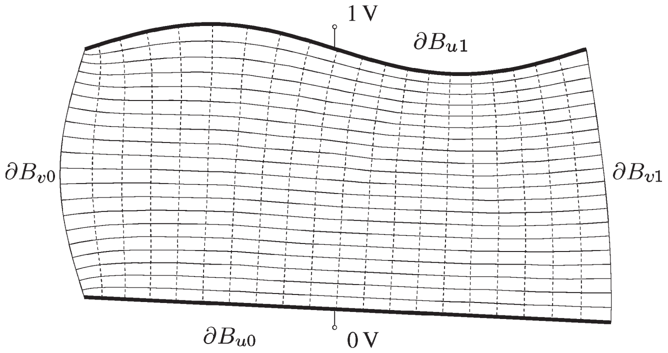


Fig. 1. Nearly rectangular grid elements achieved by selection of contour lines inside capacitor.

elements. By selection of different contour and field lines, the grid spacing along and across the electrode boundaries can be controlled, and the aspect ratios of the resulting quadrangles can be tuned as desired. However, commonly used device simulators are not designed to perform calculations on these quadrangles. Therefore, the set of grid points must be meshed by a Delaunay triangulator to obtain a conventional mesh. Because of the nearly rectangular shape of the elements, no additional mesh points have to be inserted to maintain the Delaunay criterion, and the preferred mesh densities and main directions are not disturbed. Moreover, the denser the grid spacing is chosen, the more rectangular the grid elements become.

III. Calculation of Contour and Field Lines

For the calculation of the contour lines, the Laplace equation with Dirichlet conditions (electrodes) and homogenous Neumann conditions is solved. The domain is initially meshed, and the solution is found numerically by the method of Finite Boxes. Due to the smoothness of the Laplace equation as an elliptic PDE, the requirements on this initial mesh and the equation solver are not very high compared to the complexity of solving the actual semiconductor equations. Additionally, only a linear PDE has to be solved. However, too coarse meshes result in a difference of the computed and actual field distributions, and, therefore, in a possible nonorthogonality of the computed contour and field lines, which results also in a lack of orthogonality of the mesh lines. Particularly, this lack of orthogonality can be observed near boundary corners, and mesh refinement at these regions, which preserves the orthogonality of the contour and field lines, is required.

For numerical reasons, the field lines themselves are not calculated directly by walking along the directions of the electric field. Instead, a more elegant approach via a two-dimensional duality is employed, which can be applied for the given set of boundary conditions (refer to Appendix). By replacing the electrodes of the capacitor with the homogenous Neumann conditions and the initial Neumann boundaries with the electrodes (Dirichlet conditions), a dual system is defined. The contour lines (field lines) of this dual system describe the field lines (contour lines) of the former, and vice versa.

In summary, the set of differential equations for the dual scalar fields u and v in two dimensions are

$$u_{xx} + u_{yy} = 0 \tag{1}$$

$$v_{xx} + v_{yy} = 0 \tag{2}$$

which describe the contour lines shown in Fig. 1, with $u = \text{const}$ and $v = \text{const}$. By introduction of the boundary conditions

$$\begin{aligned} \partial B_{u0} &\rightarrow u = 0, \quad \partial_n v = 0 \\ \partial B_{u1} &\rightarrow u = 1, \quad \partial_n v = 0 \\ \partial B_{v0} &\rightarrow v = 0, \quad \partial_n u = 0 \\ \partial B_{v1} &\rightarrow v = 1, \quad \partial_n u = 0 \end{aligned} \tag{3}$$

a normalization of the form

$$u \in [0,1] \qquad \text{and} \qquad v \in [0,1] \tag{4}$$

is achieved due to the maximum principle [7], [10].

IV. Differences to Elliptic Grid Generation

Examination of the above equations reveals a similarity to elliptic mesh generation [11], [12]. As illustrated in Fig. 2, (1)–(3) are a transformation of the physical domain (x, y) to a computational domain (u, v), with a one-to-one mapping of the physical points to the computational points in the range $(u, v) \in [0,1] \times [0,1]$. In the formulation of elliptic grid generation, the differential equations are first transformed to the computational domain, delivering the nonlinear-equation system

$$\begin{aligned} &\alpha(x_v, y_v)\, x_{uu} \\ &\qquad - 2\,\beta(x_u, x_v, y_u, y_v)\, x_{uv} + \gamma(x_u, y_u)\, x_{vv} = 0 \end{aligned} \tag{5}$$

$$\begin{aligned} &\alpha(x_v, y_v)\, y_{uu} \\ &\qquad - 2\,\beta(x_u, x_v, y_u, y_v)\, y_{uv} + \gamma(x_u, y_u)\, y_{vv} = 0. \end{aligned} \tag{6}$$

The solutions $x(u, v)$ and $y(u, v)$ of this equation system directly deliver the physical point coordinates x and y [13]. For mesh-generation purposes, a rectangular grid $[0, u_1, \ldots, u_m] \times [0, v_1, \ldots, v_n]$ is spanned over the computational domain. Based on this grid, the transformed differential equation (5) and (6) have to be solved, usually iteratively by the method of finite differences. The solution directly delivers the physical grid points (x_i, y_j). The physical grid can be interpreted as a warped ortho grid with the grid points (x_i, y_j) according to the computational points (u_i, v_j) with $i = 0, 1, \ldots m$ and $j = 0, 1, \ldots n$.

In contrast, the PDE mesh-generation algorithm proceeds differently. The differential equation (1)–(3) are solved directly on the physical domain, and the mesh points $(x, y)_k$ are found by point location inside the distributions (u, v)

$$(u, v)_k \xrightarrow{\text{Pointlocation}} (x, y)_k. \tag{7}$$

In other words, the corresponding triangle t is searched in the (u, v) regime, which involves the value pair $(u, v)_k$. The linear

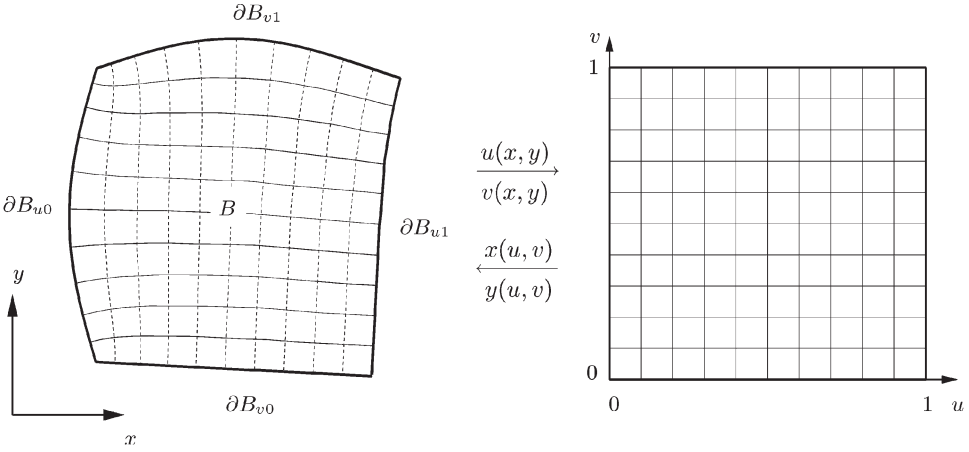


Fig. 2. Transformation of physical domain (x, y) to computational domain (u, v).

interpolation of u, v, x, and y inside the triangle t delivers the final coordinate values $(x, y)_k$. This technique shows two advantages compared to the elliptic method. On one hand, only a linear differential equation system has to be solved to obtain the distributions u and v. On the other hand, any desired computational point $(u, v)_k$ inside the boundaries delivers a physical mesh point $(x, y)_k$. The elliptic formulation requires an ortho grid in the computational domain; whereas, the PDE method is not limited to such a grid. Once the distributions u and v have been calculated, a point insertion of any desired point set is possible. The impact of this advantage will be shown in Section V.

An example of an MOS structure is shown in Fig. 3. The silicon segment of this transistor will be meshed by the PDE method. The initial mesh of this segment with the distribution u is drawn in this figure based on the Dirichlet boundary setting, which is schematically drawn. A relatively crude mesh can be used. Only at the boundary corners, a refinement that maintains the orthogonality of the two dual scalar fields is implemented. In detail, the included angle between the gradients of the two dual distributions is calculated for each triangle. The deviation of this angle to the right angle is a measure for the directional error of the dual contour lines, and the triangles are refined, if necessary. Due to its robustness, in our two-dimensional applications, the triangular Delaunay mesh generator TRIANGLE [14] is applied to produce the initial and the entire simulation meshes.

Derived from this calculation, a final triangulated simulation mesh with 19 × 12 contour ticks in the (u, v) space is produced, which is shown in Fig. 4. Frequently, the mesh in the other segments of the device is of minor interest and can, therefore, be meshed without placing the restrictive demands on the mesh density or orthogonality. However, because of the rectangular structure of the contacts, the PDE method is used as well and works without requiring point insertions inside the contact regions.

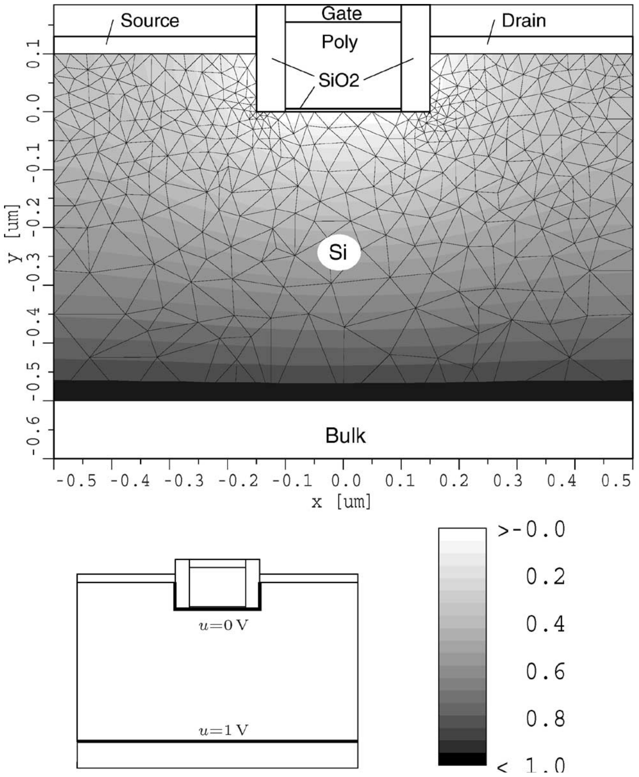


Fig. 3. Two-dimensional MOS transistor. Silicon segment is meshed by PDE method. On this segment, initial mesh and potential distribution are shown, based on Dirichlet boundaries shown on bottom.

V. Feature Edges, Additional Refinement, and Terminating Lines

For device simulation, it is necessary to preserve the connectivity between the different segments. At the boundaries, the meshes of the different segments have to be point conforming, and feature edges and points, especially those points where more than two materials touch–the so-called triple points, must be preserved. In the basic formulation of elliptic mesh generation, a rectangular grid is spanned over the computational domain. Therefore, this technique might possibly lose structural edges and points, if they are not located on the computational coordinates u_i or v_j. Often, a parameterization of the geometry in dependence of the arc length is chosen. In this case, the different structural corners can be assigned to the computational contour ticks u_i or v_j. Unfortunately, it is impossible to preserve the orthogonality of the grid lines at the boundary simultaneously, as the normal derivation at the boundaries

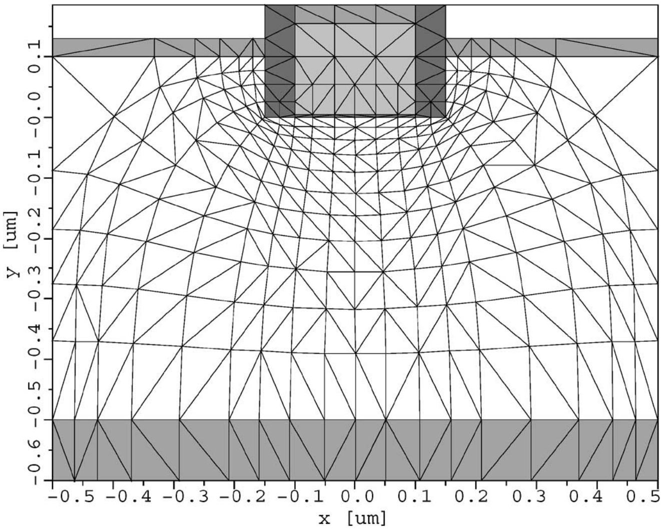


Fig. 4. Resulting mesh produced by PDE method.

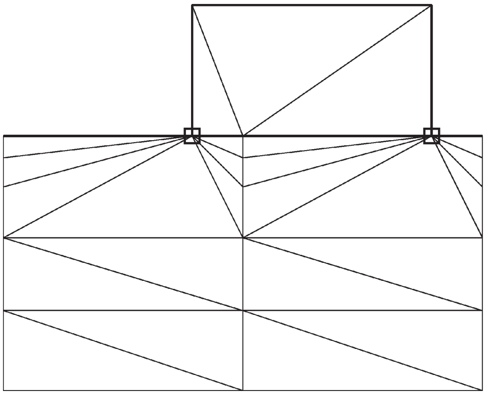


Fig. 5. Inclusion of corner points in mesh-generation process. Orthogonality of triangles at boundary is disturbed, caused by feature points marked with □. Feature points cause curtain-like mesh with badly shaped triangles to maintain Delaunay properties.

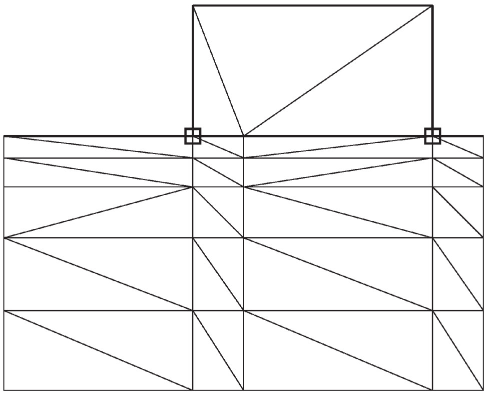


Fig. 6. Repaired mesh. Marked feature points are inserted as contour ticks too, which propagate through whole region. As a consequence, a lot of mesh points are generated at intersections with orthogonal contour lines.

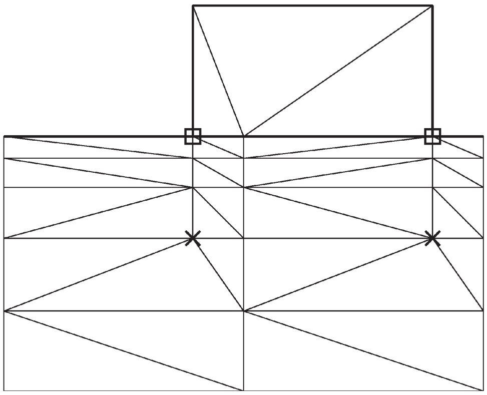


Fig. 7. Additional contour lines are terminated inside region at areas where mesh can be allowed to be coarser, and distortion of orthogonality does not seriously harm calculation. Termination points are marked with ×.

cannot be chosen independently any longer. When using the above boundary conditions, not even an assertion of the required grid spacing or location of the boundary points in the (u, v) regime can be chosen. However, for device simulation, these structural corners must be preserved. This behavior can be easily handled by the PDE mesh-generation method with the additional inclusion of these geometry points and edges in the final triangulation process, which is shown in Fig. 5.

However, as demonstrated in Fig. 5, badly shaped mesh elements near these feature points are created. The orthogonality and main directions of the mesh lines will be destroyed. An improvement is obtained if these necessary boundary points are delivered to the PDE mesh generation as additional contour ticks u_i or v_j, which is shown in Fig. 6.

Of course, this improvement by the additional contour ticks has the disadvantage that the mesh lines induced by such an additional tick will propagate through the whole simulation domain. As a consequence, additional mesh points are created at the intersections to its orthogonal contour ticks. This flaw can only be resolved by allowing termination of the contour lines, realized by a terminating-lines algorithm, shown in Fig. 7. The continuation of the contour lines is stopped at locations where a coarser mesh can be allowed and the disturbed orthogonality does not influence the simulation.

Even a desired mesh spacing along or normal to the boundaries or in areas of special interest can be achieved in the same way. Mesh refinement and coarsement at some areas can be handled by the additional contour ticks in combination with the termination of the mesh lines in the regions of minor interest.

VI. Terminating Lines Algorithm

The terminating-lines algorithm plays an important role in the quality of the mesh. Based on the origin of the contour tick, different mesh lines have different priorities regarding termination. The top priority is given to the contour lines, which are caused by the feature points. These contour ticks are not necessarily required with regard to the mesh densities. If no deletion of the contour line occurs and the contour line propagates to the opposite boundary, a feature point is produced there as well. Thus, a possible "infection" with the contour ticks; therefore, mesh points to the neighboring segments are induced. In other words, a contour tick caused by a feature point is absolutely undesired at the opposite boundary. For the other mesh lines, the priorities for termination can be related to the amount of saved points, for instance.

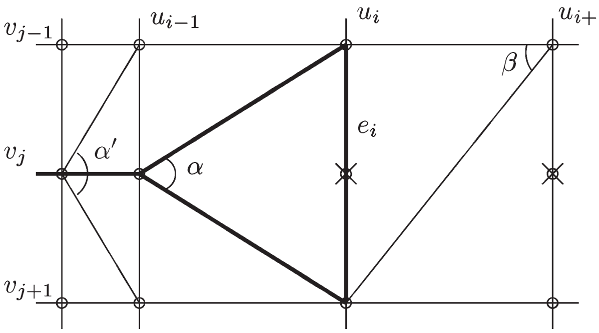


Fig. 8. Termination of mesh line v_j. Point (u_i, v_j) can be safely removed.

For an allowed termination of the mesh lines, the following conditions have to be satisfied.

1) An optional criterion decides if a coarser grid is desired. This criterion can be realized as a doping or a surface-distance dependent criterion, for instance. For the feature-point-induced contour lines, this criterion may always be satisfied.
2) Optionally a criterion decides if the disturbed orthogonality is acceptable. Here, also surface-distance criterions can be applied.
3) Finally, it must be checked if the Delaunay criterion allows a deletion of the mesh line. Of course, the entire Delaunay mesh generator would produce a valid Delaunay mesh. However, the deletion can only be allowed if the distortion of the mesh stays local.

In detail, the following consideration, depicted in Fig. 8, delivers a criterion for a possible termination. Following a contour line v_j, it should terminate as soon as possible, which implies that criteria 1) and 2) are fulfilled already. A nearly rectangular shape of the mesh cells, almost satisfied by the dual contour lines, is also supposed. Crucial is only the first deletion point, here (u_i, v_j). The edge e_i must be a valid mesh line; otherwise, the curtain-like mesh, already shown in Fig. 5, is generated by the Delaunay mesher. If this point is removed, the following $u_{i+1}, \ldots$ can be deleted safely. The edge e_i itself is only valid if the sum of its opposing angles α and β is less than 180° (a consequence of the Delaunay postulation [15]). Because of the rectangular shape, $\beta < 90°$ is always satisfied. Therefore, if $\alpha < 90°$ is satisfied too, the edge is valid, and the point can be removed. As can be seen in the figure, the deletion of (u_{i-1}, v_j) with $\alpha' > 90°$ may cause an angle sum of more than 180°, which results in the curtain-like mesh.

Fig. 9(a) shows the result of the PDE mesh generation of the previously introduced MOS transistor. Under the gate contact, a very dense mesh is produced. Potential high-carrier-concentration variations can be accurately resolved; whereas, the termination of the mesh lines nearly guarantees not to affect the coarse mesh elements at the bottom of the device, where only a bulk current has to be handled. In the detailed view [Fig. 9(b)], it can be seen that the anisotropic mesh densities are maintained as expected. Aspect ratios of the mesh elements of 5–20 can be handled easily.

VII. Extension to Three Dimensions

In a straightforward manner, the two-dimensional strategy is extended to three dimensions. Here, a set of three differential equations is used for the PDE point placement as

$$u_{xx} + u_{yy} + u_{zz} = 0 \tag{8}$$

$$v_{xx} + v_{yy} + v_{zz} = 0 \tag{9}$$

$$w_{xx} + w_{yy} + w_{zz} = 0. \tag{10}$$

On the simulation domain B, the according boundary conditions read

$$\begin{aligned}
\partial B_{u0} &\rightarrow u = 0, \quad \partial_n v = 0, \quad \partial_n w = 0\\
\partial B_{u1} &\rightarrow u = 1, \quad \partial_n v = 0, \quad \partial_n w = 0\\
\partial B_{v0} &\rightarrow v = 0, \quad \partial_n u = 0, \quad \partial_n w = 0\\
\partial B_{v1} &\rightarrow v = 1, \quad \partial_n u = 0, \quad \partial_n w = 0\\
\partial B_{w0} &\rightarrow w = 0, \quad \partial_n u = 0, \quad \partial_n v = 0\\
\partial B_{w1} &\rightarrow w = 1, \quad \partial_n u = 0, \quad \partial_n v = 0.
\end{aligned} \tag{11}$$

In analogy to the already shown two-dimensional dual distributions in Fig. 1, the three types of contour surfaces for the three dimensions are shown in Fig. 10. Usually, the orthogonality of the contour surfaces cannot be guaranteed for the three dimensions. Orthogonality can only be achieved if the curvilinear coordinates u, v, and w follow the lines of curvature, which are the lines of maximal and minimal curvature. A detailed discussion concerning the three-dimensional orthogonal meshes is given in [11], for instance. However, resulting from the chosen boundary conditions, at least at the boundaries, the orthogonality is achieved. For instance, on boundary ∂B_{u0}, which forms a contour surface of $u = 0$, it is guaranteed that this surface is perpendicular to the contour surfaces of $v = \text{const}$ and $w = \text{const}$. The orthogonality of the surfaces $v = \text{const}$ versus $w = \text{const}$, however, cannot be guaranteed at this boundary. Analogous considerations are valid at the other five boundaries with the corresponding contour surfaces. Consequently, the constructed mesh cells show clipping faces, which are conforming or perpendicular to the boundaries, approximately quadrilateral prismatic bodies. Inside the domain, orthogonality is not observed, but usually the cuboidal shape of the mesh elements is nearly fulfilled and the tetrahedral Delaunay mesh generator works fine. For the generation of the initial mesh and the simulation grid, our in-house tetrahedral Delaunay mesh generator DeLink [15] is invoked.

VIII. Two-Dimensional Examples

A. *Comparison of an Isotropic Method With PDE Method in Two Dimensions*

Based on an MOS transistor structure, the PDE method is compared to a common mesh-refinement technique concerning the rank of the system matrix and execution time of the electric simulation. As a criterion for the mesh quality, the variations of the simulated output characteristics are examined.

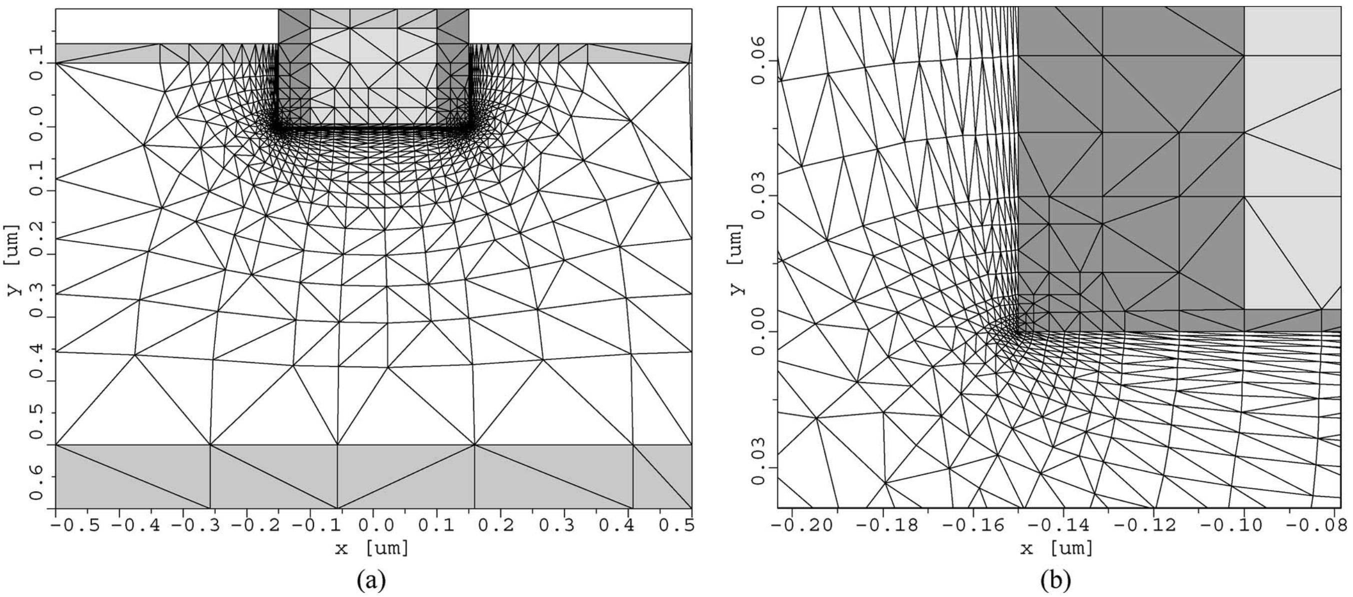


Fig. 9. Denser mesh, with terminating lines to prevent high mesh density in areas of low interest. (a) Whole device structure. (b) Detail of structure. High aspect ratios of mesh elements near surface can be seen.

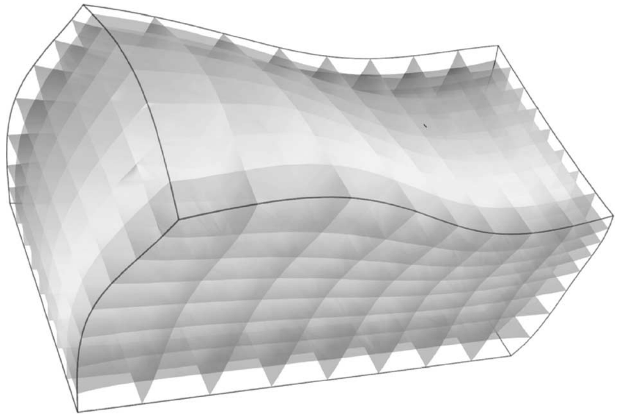


Fig. 10. Three-dimensional contour surfaces.

In the given structure, the field oxide has a thickness of about 1 nm. Therefore, under the polysilicon, a transversal mesh spacing of a fraction of a nanometer is required. In analogy to the PDE method, the conventional mesh-refinement method tunes the transversal mesh spacing by a maximum element area limit, which must not be reached by the triangles, starting with a small value at the silicon-oxide boundary, which grows towards the bulk contact. The mesh spacing ρ and maximum area limit $A_{\max}$ follow

$$A_{\max}(y) = \rho^2 \frac{\sqrt{3}}{4} \quad \text{with} \quad \rho(y) = ky + d_0 \qquad (12)$$

with an initial spacing d_0 and a growth factor of 20% $(k = 0.2)$. For the generation of the unstructured mesh, the triangular Delaunay mesh generator TRIANGLE [16] has been used again. A comparison of several calculations with different mesh densities has shown that the final output characteristic is reached if the initial mesh distance d_0 is set to 0.1 nm. Continuous reduction of the mesh densities shows a variation in the output characteristic of less than 0.5%. Accordingly, a detail of the final unstructured simulation mesh is depicted in Fig. 11(a). As expected, the triangles show a transversal spacing of nearly 0.1 nm, resulting in nearly equilateral triangles, and, therefore, also a lateral spacing of about this size.

In comparison, a detail of the PDE mesh is shown in Fig. 11(b). To achieve a simulation result with a variation of the output characteristic with less than 0.5% to the previous mesh method, the transversal mesh spacing must be set to 0.05 nm, but in contrast, the lateral mesh spacing can be much coarser, and a value of 0.5–1.0 nm has been used. The growth factor of the transversal mesh line spacing is also set to 20% from one to the next mesh line. The impact of these two techniques can be seen in the execution time and the rank of the system matrix required for simulating the output characteristics of the device, depicted in Table I. The rank of the system matrix of the isotropic method is about six times larger than that for the PDE-based method, which roughly translates to memory consumption. In contrast, a 12 times higher execution time is required for the isotropic method.

B. Two-Dimensional Nonplanar MOS Transistor

This example demonstrates the applicability of the PDE-based mesh-generation method to nonplanar surfaces. The device structure of a high-voltage MOS device together with the simulation mesh is shown in Fig. 12(a). To highlight the different material segments, only the mesh on the silicon domain is drawn. However, no special attention must be paid to the meshes of the other segments. Care is taken of the edges produced by the boundary points, which define the geometry of the silicon surface. Starting at these points, mesh lines are produced, which can be terminated close to the boundaries [confer to the detail of the entire structure in Fig. 12(b)], and the desired point density can be reconstituted shortly.

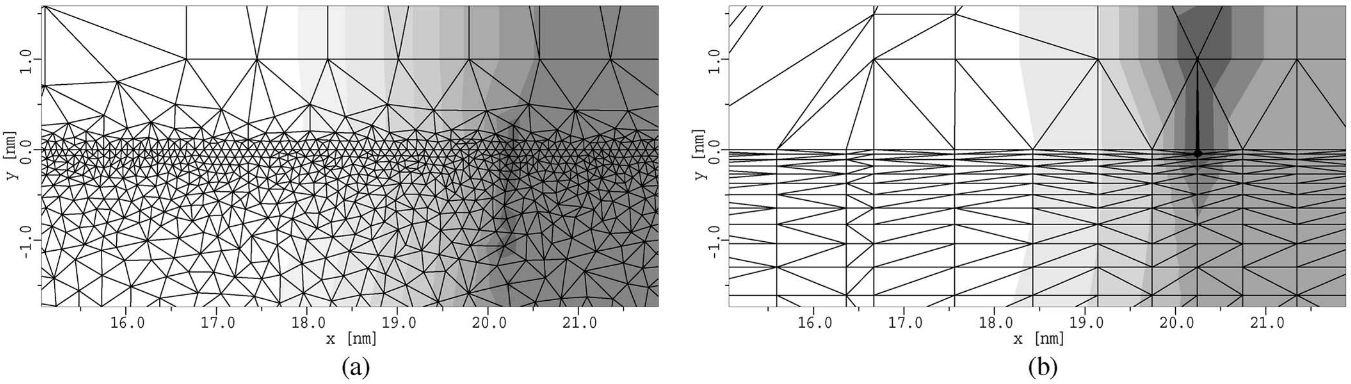


Fig. 11. Detail of MOS structure. (a) Mesh refinement by maximum allowable triangle area. (b) PDE-based mesh generation.

TABLE I
COMPARISON OF SYSTEM MATRIX SIZE AND EXECUTION TIME REQUIRED TO SIMULATE OUTPUT CHARACTERISTIC OF TRANSISTOR WITH 49 OPERATING POINTS ON AMD-64 3000+ FOR BOTH METHODS

	System matrix		Execution time	
	rank	relative	seconds	relative
Maximum area refinement	20,262	5.97	1,407.4	12.13
PDE based mesh generation	3,392	1.00	116.0	1.00

IX. THREE-DIMENSIONAL EXAMPLES

A. FinFET

To overcome some of the red-brick walls that conventional CMOS scaling faces today, for instance, FinFET structures have been advocated [17]. A thin silicon fin connects the two large contact pads, known as the source and the drain. The whole structure is placed, shielded by a thick oxide layer, on top of a wafer. An example structure is shown in Fig. 13. Over the channel region, a polysilicon gate contact is deposited, which is shielded from the fin by an oxide layer. Thus, the carrier flow can be controlled in the domain under the polysilicon area by the gate potential.

Because of the planar and simple segment geometries, the mesh-generation procedure should, in principle, be straight forward. However, due to the three-dimensional nature of the problem, care has to be taken. Usually, the three-dimensional simulations based on the unstructured meshes are problematic due to the misaligned directions of the tetrahedrons in the fin. The different carrier concentrations, which are achieved from the grid points in the fin, result in instabilities of the equation solver and the simulations often fail to converge.

Better results can be achieved by the ortho grids. Because of the planarity of the boundaries and alignment to the coordinate axes, the grid-generation process works well. Unfortunately, as shown in Fig. 14(a), all the grid lines required in one segment propagate through all other segments and result in a large number of unnecessary grid points.

With tetrahedral meshes, where the fin is meshed by the PDE-based method, excellent results have been obtained. The finally used mesh, which produces the same device simulation results as the ortho-product grids, is shown in Fig. 14(b). Along the fin and the contact pads, the tetrahedrons are aligned in the direction of the current flow. In addition, it can be clearly seen that the high mesh density in the fin stays only local.

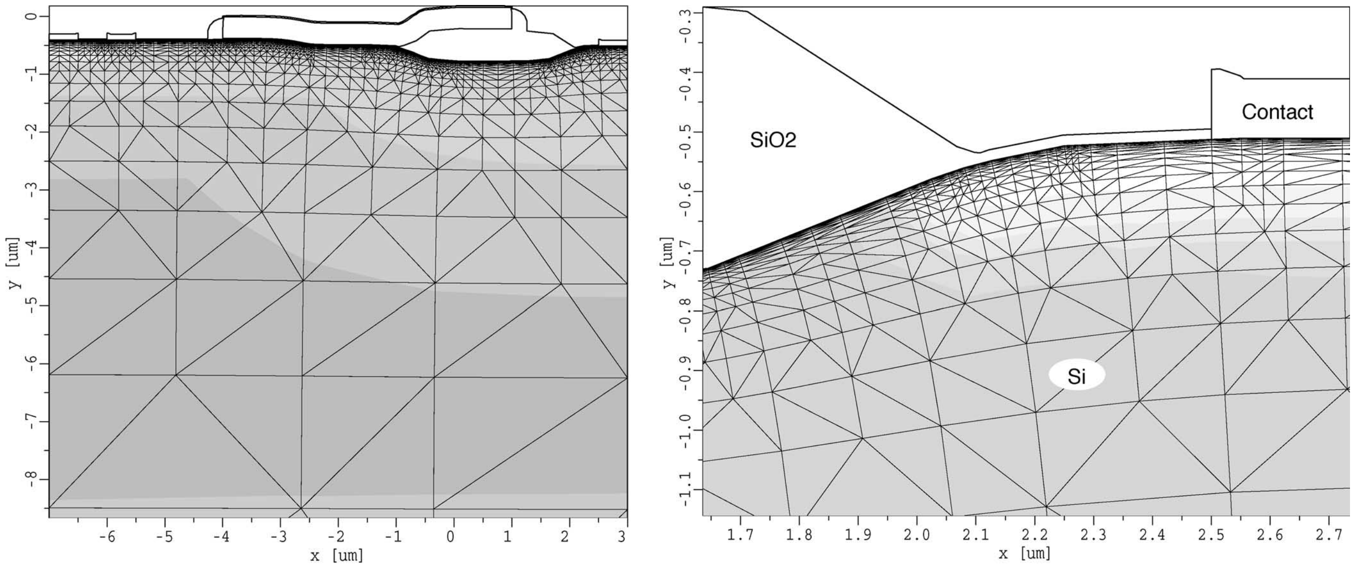
A comparison of the methods is given in Table II. The output characteristics of the device, based on 7 × 7 operating points, had been calculated using the above ortho grid and two types of the PDE methods with different mesh densities. For verification, if the PDE-based mesh is sufficiently dense, a second calculation with a much denser mesh in the fin region, compared to the first calculation, was performed, and no significant differences in the output characteristics were detected.

B. Electronically Erasable Programmable Read-Only Memory (EEPROM) Memory Cell

A three-dimensional example of an EEPROM memory cell is shown in Fig. 15. This structure shows a quarter of the entire memory cell. Here, a full three-dimensional manufacturing cycle was performed. Only the first oxidation step of the silicon wafer was carried out in a two-dimensional approximation and was expanded to three dimensions by extrusion. The following deposition and etch processes of the floating gate, the insulator layer between the two gates, and the control gate can only be performed in three dimensions. The mesh for the device simulation is prepared by applying the PDE-based method for the silicon and field-oxide segment. For a detailed device simulation, including the doping-dependent nonlinearities of the silicon, the concentration profile of the dopants must be carefully resolved in the active areas. Therefore, the silicon segment is meshed by the PDE-based method. A high point density is required under the thin parts of the field oxide. Around this region, the transversal resolution can be more crude, reaching an overall coarse mesh at the bottom regions. Also in the lateral direction, where the field oxide weakens the influence of the gate voltage, the mesh density can be coarser. The cumulative number of the mesh points and tetrahedrons is about 12 000 points and 52 000 tetrahedrons, respectively.

X. CONCLUSION

With our PDE-based mesh-generation method, a flexible algorithm for highly accurate simulations of modern



(a) (b)

Fig. 12. Nonplanar MOS transistor. (a) Whole silicon segment. (b) Bird's peak.

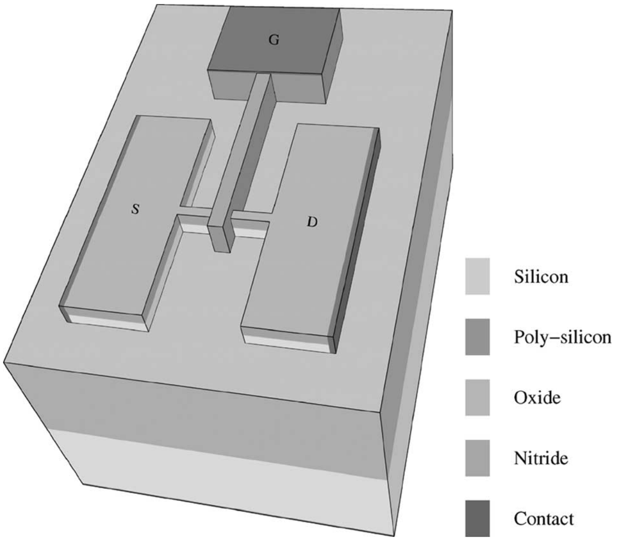


Fig. 13. Geometry of simulated FinFET structure.

semiconductor devices is available. For the two-dimensional simulations, in combination with the use of a terminating-line algorithm, the number of mesh points is drastically reduced without the loss of accuracy. Therefore, calculation times are highly reduced. Especially for the calculation of multiple operating points combined with frequent device optimization loops, the savings in time is enormous.

For the three-dimensional simulations, good results have been achieved by the PDE method as well. Naturally, the complexity for the three-dimensional geometries and their data representation is much higher than that for two dimensions. Especially for the applicability of the PDE-based method, care has to be taken concerning boundary smoothing for the prevention of the input- and geometry-induced point generations and, therefore, also distortion of the orthogonality close to the boundaries.

Appendix
Contour and Field Lines

Equation (1) can be interpreted as the result of a two-dimensional electric-field calculation [19] with

$$\nabla \times \mathbf{E} = \mathbf{0} \tag{13}$$

$$\nabla \mathbf{E} = 0. \tag{14}$$

Usually, a permittivity appears in (14). However, in the case of a constant scalar permittivity, it can be skipped in the divergence equation. With the ansatz of a scalar potential u

$$\mathbf{E} = -\nabla u. \tag{15}$$

Equation (13) is implicitly satisfied, and insertion into (14) delivers the initial (1).

A second kind of solution of (13) and (14) is derived by an electric vector potential $\mathbf{A}$, defined as

$$\mathbf{E} = \nabla \times \mathbf{A} = \nabla \times (v\,\mathbf{e}_z) = (\nabla v) \times \mathbf{e}_z \tag{16}$$

which satisfies (14) implicitly. Because of the two dimensionality of the problem, this potential shows only a z-component. Insertion into (13) delivers

$$0 = \nabla \times \mathbf{E} = \nabla \times [(\nabla v) \times \mathbf{e}_z] = -\mathbf{e}_z\,(\nabla\nabla v) \tag{17}$$

and finally for the nonzero component only

$$\Delta v = 0 \tag{18}$$

which is the same as (2).

The Dirichlet boundary conditions of u

$$\partial B_{u0} \ \to u = 0 \tag{19}$$

and

$$\partial B_{u1} \ \to u = 1 \tag{20}$$

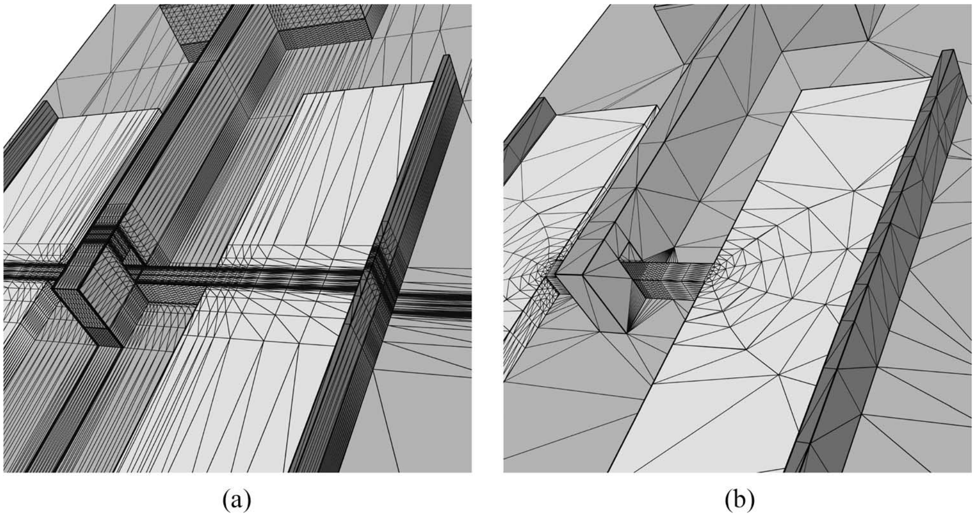


Fig. 14. Meshed FinFET structure. Oxide and nitride layers around fin have been removed for visualization purposes. (a) Ortho-product simulation grid. Grid elements are split into tetrahedrons for visualization purposes only. (b) Tetrahedral simulation mesh. PDE method is applied to silicon segment as well as for cuboidal segments.

TABLE II
COMPARISON OF DIFFERENT GRIDS. RANK DESCRIBES RANK OF SYSTEM MATRIX OF EQUATION SYSTEM ASSEMBLED BY MINIMOS-NT [18], WHICH WAS USED FOR ELECTRICAL SIMULATION

Grid type	Points	Elem.	Elem. type	Rank	Sim. time
Ortho grid	48,480	42,840	Cuboids	83,998	124m30s
PDE mesh 1	11,186	32,666	Tetrahedra	16,238	38m39s
PDE mesh 2	42,404	128,819	Tetrahedra	65,132	390m10s

describe a vanishing tangential component of the electric field

$$0 = E_t = \mathbf{t} \cdot \mathbf{E} = \mathbf{t} \cdot [(\nabla v) \times \mathbf{e}_z] \qquad (21)$$

$$= \nabla v \cdot (\mathbf{e}_z \times \mathbf{t}) = -\nabla v \cdot \mathbf{n} \qquad (22)$$

$$\Rightarrow \partial_n v = 0 \qquad (23)$$

and result in the homogenous Neumann conditions for v. Alternatively, the homogenous Neumann conditions for u

$$\partial B_{v0,1} \rightarrow \partial_n u = 0 \qquad (24)$$

with a zero normal component of the electric field result in

$$0 = E_n = \mathbf{n} \cdot \mathbf{E} = \mathbf{n} \cdot [(\nabla v) \times \mathbf{e}_z] \qquad (25)$$

$$= \nabla v \cdot (\mathbf{e}_z \times \mathbf{n}) = -\nabla v \cdot \mathbf{t} \qquad (26)$$

$$\Rightarrow v = \text{const} \qquad (27)$$

which are the Dirichlet conditions for v. One of the Dirichlet boundary values $v_0 = v = \text{const}$ can be chosen arbitrarily, and the second value v_1 has to be evaluated by integration from one Dirichlet boundary to the opposite, for instance along the boundary ∂B_{u0}

$$\Lambda = \int\limits_{\partial B_{u0}} E_n \, ds = \int\limits_{\partial B_{u0}} -\partial_n u \, ds$$

$$\Lambda = \int\limits_{\partial B_{u0}} E_n \, ds = \int\limits_{\partial B_{u0}} \nabla v \cdot \mathbf{t} ds = \int\limits_{\partial B_{u0}} dv = v_1 - v_0. \qquad (28)$$

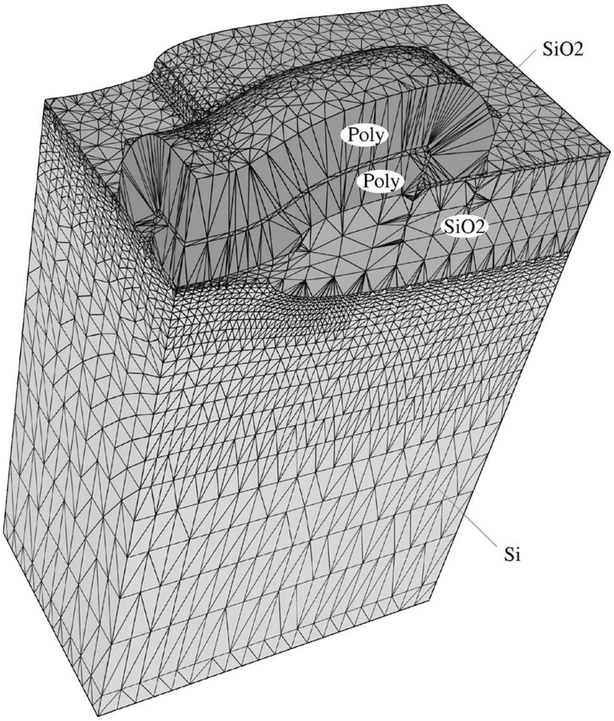


Fig. 15. Three-dimensional example of EEPROM memory cell.

Consequently, if the just described boundary conditions and (28) are satisfied, the two scalar fields u and v describe the same problem.

An interesting side effect of these scalar fields can be seen when the field lines of $\mathbf{E}(u)$ are evaluated. The differential equation for the field lines can be written as

$$0 = dy \, E_x - dx \, E_y \overset{(16)}{=\!=} dy \, \partial_y v + dx \, \partial_x v = dv \qquad (29)$$

or consequently

$$v = \text{const} \qquad (30)$$

which means that the contour lines of (2) describe the field lines of (1). However, the shape of the field lines of $\mathbf{E}$ is invariant to a constant scaling factor and an offset of v

$$\tilde{v} = \alpha\, v + \beta \tag{31}$$

$$0 = d\tilde{v} = \alpha\, dv. \tag{32}$$

Therefore, the boundary values v_0 and v_1 can be set to two different arbitrary values, and the normalization $v_0 = 0, v_1 = 1$ delivers the complete set of the boundary conditions, already shown in (3).

This duality is a significant advantage of the method. As the contour lines and the field lines stay orthogonal, the contour lines of $u = \text{const}$ and $v = \text{const}$ are orthogonal, too; therefore, the generated grid lines will stay nearly orthogonal. As a consequence of the chosen boundary conditions (3), the contour lines are also boundary conforming or orthogonal to the boundaries. A rectangular grid can always be split into a triangular Delaunay mesh; therefore, concerning the required point insertion for achieving the Delaunay properties of the mesh, the nearly satisfied orthogonality eases the generation of the Delaunay mesh as well.

ACKNOWLEDGMENT

The authors would like to thank Dr. G. Schrom, Intel, for inspiring this paper.

REFERENCES

[1] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*. New York: Springer-Verlag, 1984.

[2] P. Markowich, C. Ringhofer, and C. Schmeiser, *Semiconductor Equations*. New York: Springer-Verlag, 1990.

[3] A. Franz, G. Franz, S. Selberherr, C. Ringhofer, and P. Markowich, "Finite boxes—A generalization of the finite difference method suitable for semiconductor device simulation," *IEEE Trans. Electron Devices*, vol. ED-30, no. 9, pp. 1070–1082, Sep. 1983.

[4] M. Sever, "Delaunay partitioning in three dimensions and semiconductor models," *COMPEL*, vol. 5, no. 2, pp. 75–93, 1986.

[5] P. George and H. Borouchaki, *Delaunay Triangulation and Meshing*. Paris, France: Hermes, 1998.

[6] B. Haindl, R. Kosik, P. Fleischmann, and S. Selberherr, "Comparison of finite element and finite box discretization for three-dimensional diffusion modeling using AMIGOS," in *Proc. Simul. Semicond. Processes and Devices*, Kyoto, Japan, Sep. 1999, pp. 131–134.

[7] R. Kosik, P. Fleischmann, B. Haindl, P. Pietra, and S. Selberherr, "On the interplay between meshing and discretization in three-dimensional diffusion simulation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 11, pp. 1233–1240, Nov. 2000.

[8] D. Arnold, A. Mukherjee, and L. Pouly, "Locally adapted tetrahedral meshes using bisection," *SIAM J. Sci. Comput.*, vol. 22, no. 2, pp. 431–448, 2001.

[9] W. Wessner, C. Heitzinger, A. Hössinger, and S. Selberherr, "Error estimated driven anisotropic mesh refinement for three-dimensional diffusion simulation," in *Proc. Int. Conf. Simul. Semicond. Processes and Devices*, 2003, pp. 109–112.

[10] M. Protter and H. Weinberger, *Maximum Principles in Differential Equations*. New York: Springer-Verlag, 1999.

[11] J. Thompson, Z. Warsi, and C. Mastin, *Numerical Grid Generation*. Amsterdam, The Netherlands: North Holland, 1985.

[12] J. F. Thompson, B. K. Soni, and N. P. Weatherill, *Handbook of Grid Generation*. Boca Raton, FL: CRC, 1999.

[13] S. P. Spekreijse, "Elliptic grid generation based on laplace equations and algebraic transformations," *J. Comput. Phys.*, vol. 118, no. 1, pp. 38–61, Apr. 1995.

[14] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and delaunay triangulator," in *Proc. 1st ACM Workshop Appl. Comput. Geom.: Towards Geom. Eng.,* ser. Lecture Notes in Computer Science, M. C. Lin and D. Manocha, Eds. New York: Springer-Verlag, May 1996, vol. 1148, pp. 203–222.

[15] P. Fleischmann, "Mesh generation for technology CAD in three dimensions," Ph.D. dissertation, Technische Universität Wien, Vienna, Austria, 2000. [Online]. Available: http://www.iue.tuwien.ac.at/phd/fleischmann

[16] J. R. Shewchuk, "Delaunay refinement mesh generation," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, May 1997. [Online]. Available: http://www.cs.berkeley.edu/~jrs

[17] E. J. Nowak, "Maintaining the benefits of CMOS scaling when scaling bogs down," *IBM J. Res. Develop.*, vol. 46, no, 2/3, pp. 169–180, Mar.–May 2002. [Online]. Available: http://www.research.ibm.com/journal/rd/462/nowak.html; http://www.research.ibm.com/journal/rd/462/nowak.pdf

[18] IμE, *MINIMOS-NT 2.1 User's Guide*, 2004, Vienna, Austria: Institut für Mikroelektronik, Technische Universität Wien. [Online]. Available: http://www.iue.tuwien.ac.at/software/minimos-nt

[19] J. D. Jackson, *Classical Electrodynamics*, vol. 2005. Hoboken, NJ: Wiley, 1998.

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.

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.

Elaf Al-Ani was born in Baghdad, Iraq, in 1976. He received the Diplom-Ingenieur degree in electrical engineering from Vienna University of Technology, Vienna, Austria, in 2004. He is currently working toward the Ph.D. degree at the same university.

He joined the Institute for Microelectronics, Vienna University of Technology, in May 2004. His research interests include process simulation with special emphasis on three-dimensional applications.

Tibor Grasser (SM'05) was born in Vienna, Austria, in 1970. He received the Diplom-Ingenieur degree in communications engineering, the Ph.D. degree in technical sciences, and the Venia Docendi degree in microelectronics, all from Vienna University of Technology, Vienna, Austria, in 1995, 1999, and 2002, respectively.

Since 1997, he has headed the Minimos-NT development group, working on the successor of the highly successful MiniMOS program. He was a Visiting Research Engineer with Hitachi Ltd., Tokyo, Japan, and for the Alpha Development Group, Compaq Computer Corporation, Shrewsbury. In 2003, he was an Appointed Head with the Christian Doppler Laboratory for Technology Computer Aided Design (TCAD) in Microelectronics, an industry-funded research group embedded in the Institute for Microelectronics. He is currently an Associate Professor with the Institute for Microelectronics, Vienna University of Technology. His current scientific interests include circuit and device simulations, device modeling, physical, and software aspects, in general.

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.