Mixed-Element Decomposition Method for Three-Dimensional Grid Adaptation

Ernst Leitner and Siegfried Selberherr, *Fellow, IEEE*

***Abstract*— A new method for adaptive tessellation of three-dimensional (3-D) grids is presented. A mixed-element decomposition method is introduced for local refinement of fully unstructured grids, consisting of tetrahedra and octahedra. The method preserves the shape of the elements of the initial grid and therefore the element quality. Furthermore, local anisotropies of the initial grid are preserved. The developed implementation allows efficient adaptation and is used in a finite element program for simulation of thermal diffusion processes.**

***Index Terms*— Adaptive gridding, diffusion 3-D, finite-element method, refinement, tetrahedral meshes.**

I. Introduction

IMPRESSIVE work has been done to increase the number of components of highly integrated semiconductor circuits. The continuous down-scaling has lead to submicron devices where three-dimensional (3-D) effects are becoming increasingly important. This, most obviously, leads to the demand of 3-D simulation. There is a considerable lack in accuracy of the results, however, because often, the doping profiles used as input for the device simulation are derived either from analytical models or from one- or two-dimensional process simulation by a more or less naive extension to 3-D profiles. These methods are not accurate enough, especially when the feature size is small. Therefore, true 3-D process simulation is needed.

Several process steps have already been dealt with in 3-D process simulation and resulted in successful tools for the simulation of, e.g., ion implantation [1], [2], etching, and deposition processes [3], [4]. Also several approaches for the simulation of diffusion and annealing steps have been presented recently [5], [6]. But the treatment of complex 3-D structures still suffers on difficulties in handling complicated geometries and in obtaining well-suited 3-D meshes as well as the difficulties in the verification of the computed 3-D profiles.

The grid strongly influences the necessary computational resources for solving a typical diffusion problem. On the one hand, generating an optimal 3-D grid itself needs considerable CPU time and, as the diffusion advances, several regridding steps may be required during the simulation. On the other hand, the grid quality has a substantial influence on the convergence, and the accuracy as well as the number of grid nodes affects the numerical effort for the solution.

One approach to solve this problem is to start with a coarse initial grid which is then refined during the simulation. Both the initial-grid generation and the subsequent grid refinement can be carried out efficiently. An interesting, tree-based approach is shown in [7] and [8], where the adaptive refinement starts from a rectangular bounding box. This, however, always leads to elements which are aligned to this initial bounding box. Therefore, it is hardly possible to obtain optimal grids along oblique boundaries and interfaces of general geometries.

In our approach we start the refinement based on a domain decomposition which accounts for the local geometrical properties of the structure: a coarse mixed-element grid consisting of tetrahedra and octahedra, which resolves the geometry. The generation of the initial grid may be carried out using various grid-generators (e.g., [9] or [10]), which are responsible for the quality of the initial grid. Our work focuses on the refinement done by a mixed-element decomposition preserving the essential grid properties, i.e., the grid quality and the structural anisotropies. Thereby, we can avoid the aforementioned problems effectively as long as the initial grid fulfills the required grid quality, since the proposed refinement method is basically isotropic.

In Section II, we define the demands on a grid refinement algorithm. In Section III we introduce our new mixed-element decomposition method. In Section IV we describe the numerical methods used for the solution of thermal diffusion problems. Section V shows the application of the new method to an annealing step of a MOSFET source/drain implant, and we conclude with Section VI.

II. Demands on an Adaptive Refinement Algorithm

For recursive refinement algorithms it is indispensable to preserve the grid quality. The unavoidable degradation of the grid quality has to stay within a limit which is independent of the number of refinements. To assess this degradation, some quality criteria are necessary. In, e.g., [11], several quality criteria for a finite-element approach are given. We use

$$Q^e = \frac{V_e}{h_{\max}^3} \tag{1}$$

as a measure for the element quality, where Q^e stands for the element quality, V_e is the volume of the element, and $h_{\max}$ is the maximum dimension of the element. Additionally the

Manuscript received June 5, 1996. This paper was recommended by Associate Editor Z. Yu.

The authors are with the Institute for Microelectronics, TU Vienna, Gußhausstraße 27-29, A 1040 Vienna, Austria.

Publisher Item Identifier S 0278-0070(98)05194-X.

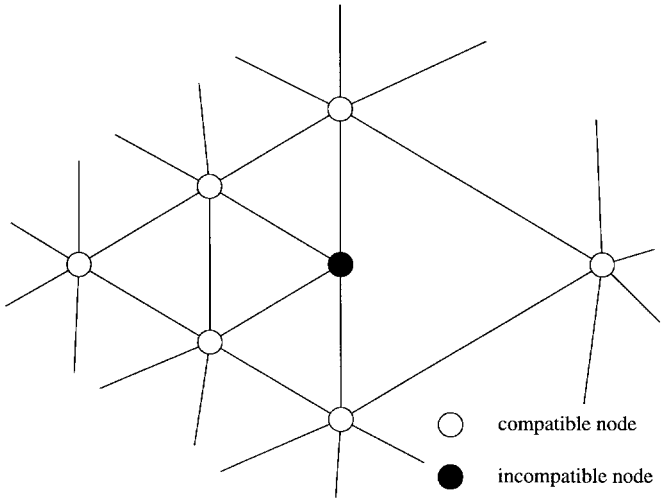


Fig. 1. Incompatible elements.

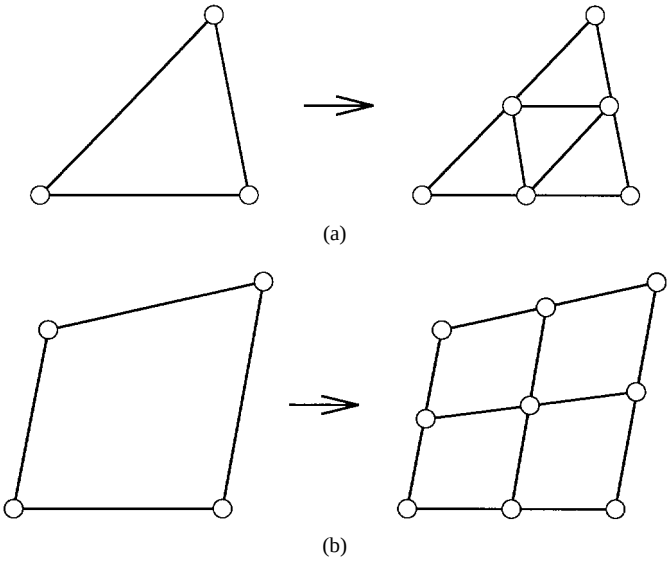


Fig. 2. Uniform refinement for triangular and quadrilateral elements.

local grid quality around one node is expressed using

$$Q^n = \frac{\min(V_i)}{\max(V_i)}, \qquad i = 1, \cdots, n_e \tag{2}$$

where Q^n is the nodal grid quality and V_i are the volumes of all n_e elements which are incident to the node. For a refinement level of $i > 1$, we have to show that the element quality is independent of the number of the level. We have to prove that there exists a constant $C_d > 0$ such that

$$Q_i^{e,n} > Q_0^{e,n} C_d, \quad C_d = \text{const.} \tag{3}$$

holds for all i, where $Q_i^{e,n}$ are the nodal or element qualities for the ith refinement level of the grid, respectively. C_d describes the quality degradation.

In process and device simulation we often have to deal with regions which show a mainly one-dimensional (1-D) behavior, e.g., like a doping profile in the channel region of a MOS transistor. In order to perform an economical simulation, the grid should reflect this property. Thus, the grid may be locally anisotropic, i.e., it has regions where it consists of long narrow elements and it is important that the refinement algorithm preserves this anisotropic character of the initial grid.

The above quality criteria (except for the nodal quality) are satisfied if the element shape is not changed by the refinement, which is the case for a uniform element decomposition, as it is used for two-dimensional (2-D) grids. Unfortunately for tetrahedral elements no uniform decomposition is available.

III. The Mixed-Element Decomposition Method

A. The Algorithm

Our method is based on unstructured finite-element grids which may consist of different element types. Additionally, we allow incompatible elements (Fig. 1), when the incompatible elements satisfy the C_0-continuity condition at the interfaces to the neighboring elements. This condition is required for convergence of the discretization scheme (see Section IV-B).

For the adaptation of such a grid, we split the elements into smaller ones and repeat the splitting as many times as necessary to satisfy the desired discretization error. Regarding a two-dimensional grid, it is well known that a triangle can be split into four smaller ones with same shape by adding one node at the center of each line and connecting them with lines. For quadrilaterals the situation is similar (Fig. 2). This refinement method preserves the element shape, and thus, the element quality is preserved as well.

Unfortunately, for tetrahedral elements no splitting method is known, which preserves the shape of the element. It is possible, however, to define a two-level splitting method, which preserves the element shape during multiple refinement. In contrast to a uniform decomposition of an element into elements of the same topology, we perform a mixed-element decomposition, where the original element is split into parts of different topology. The mixed-element decomposition is defined by the following rules.

Rule 1: We divide a tetrahedron into four tetrahedra of same shape and one octahedron. The child tetrahedra are located at the parents' corners, and the remaining part has octahedral shape (Fig. 3).

Rule 2: We divide an octahedron into six octahedra of same shape and eight tetrahedra. The child octahedra are located at the parents' corners and the remaining parts have tetrahedral shape (Fig. 4).

Rule 3: The new nodes at the surface are located in the center of the respective edges, and the new inner nodes are located at the center of the parent element.

It is important to note that the element shapes remain unchanged during further refinement steps. To illustrate this, we look at the elements, which are generated by a second refinement step of a tetrahedron (Fig. 5). The tetrahedra come from the refinement of either a tetrahedral or an octahedral child of the original tetrahedron. Clearly, the tetrahedra with tetrahedral parents have the same shape as the original tetra-

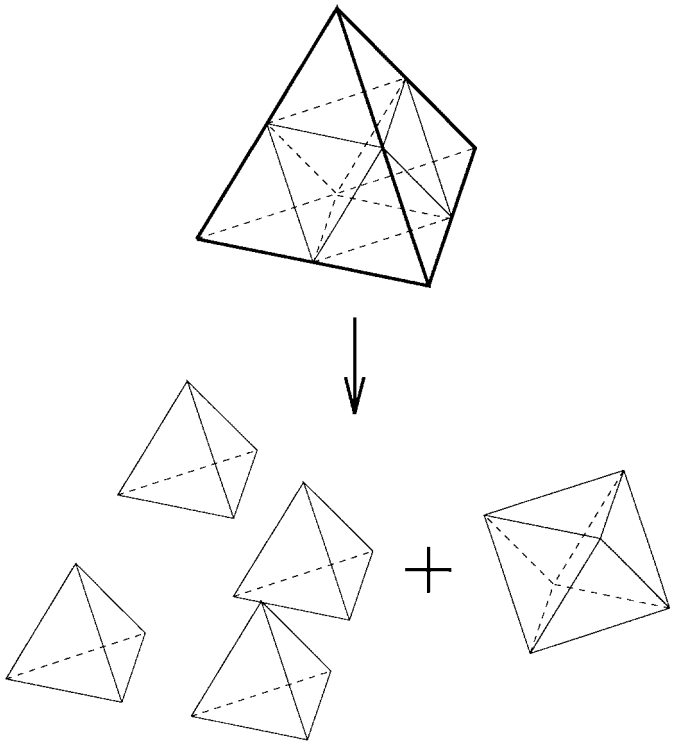


Fig. 3. Refinement for tetrahedron.

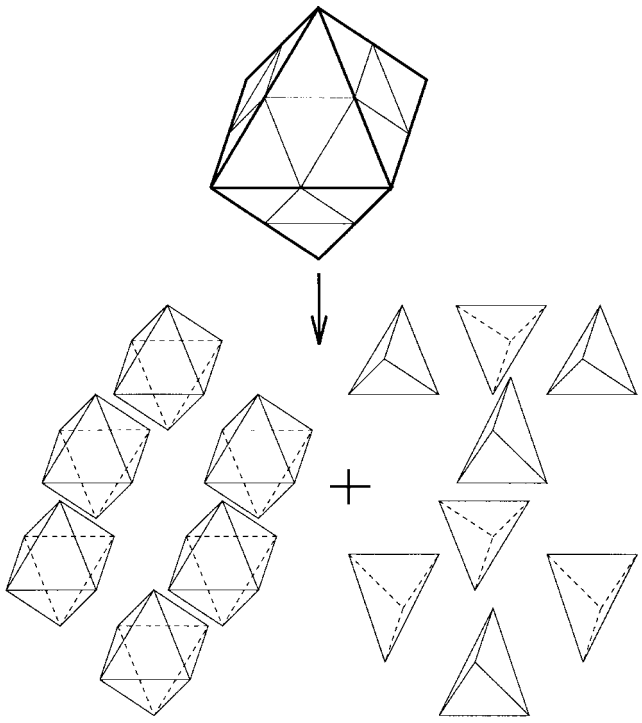


Fig. 4. Refinement for octahedron.

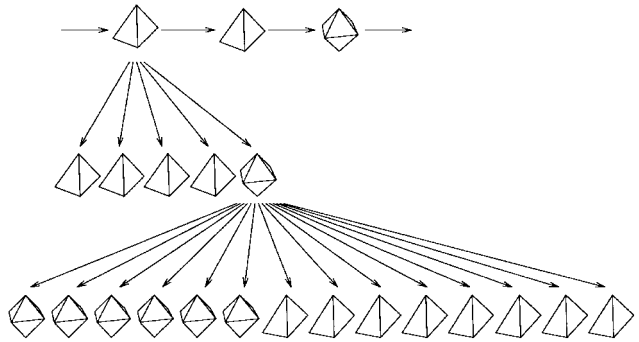


Fig. 5. Mixed element tree.

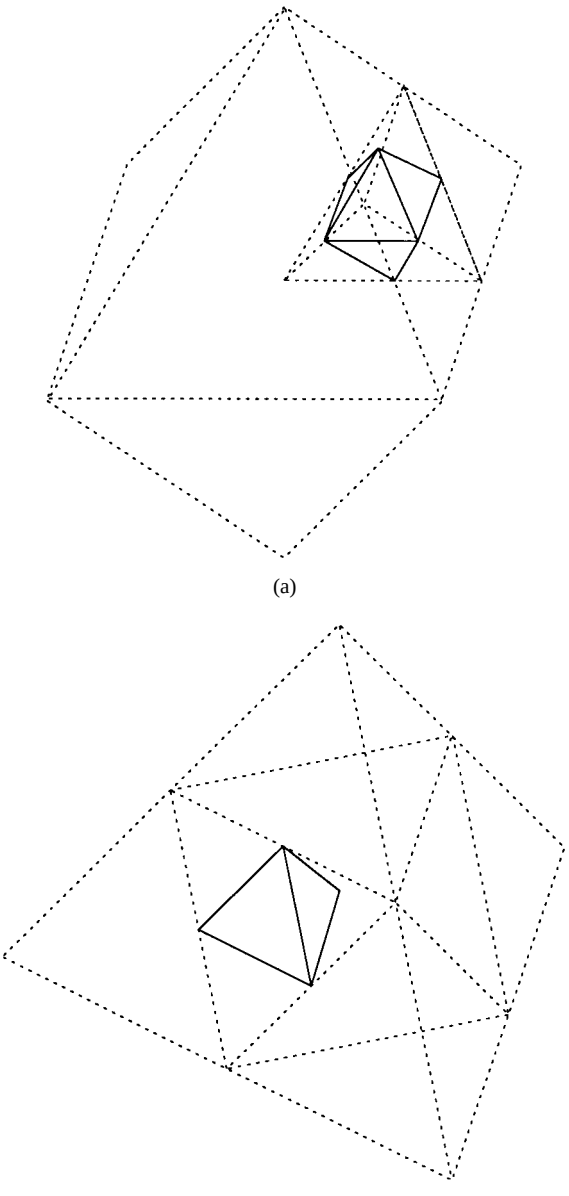


Fig. 6. Shape preservation at multiple refinement.

hedron. But also the tetrahedra derived from the octahedral child have the same shape as the original tetrahedron (Fig. 6). This shape preservation property holds for octahedra created by multiple refinement, too.

For the purpose of discretization an octahedron is split into eight tetrahedra, each of which has one face of the octahedron as ground plane and the octahedral center as opposite node (Fig. 7). Thus, only tetrahedra are treated during the finite element assembly. At the first refinement step of an initial element (octahedron or tetrahedron), a new element shape

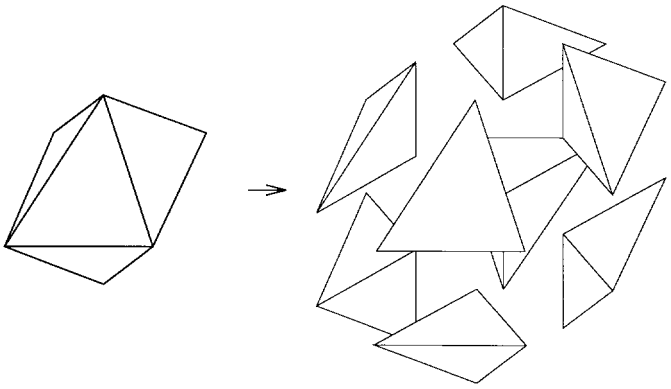


Fig. 7. Discretization for octahedron.

is introduced. All following refinement steps preserve the element shapes exactly, and therefore the quality degradation is independent of the number of grid refinements.

B. Verification of the Method

To assess the efficiency of the method it is of interest how much the grid quality is decreased through the different element shapes generated by the first refinement step. To quantify the quality degradation, the octahedral element quality is related to the tetrahedral element quality and vice versa. Taking into account the discretization of the octahedra permits a reasonable comparison of the octahedron and the tetrahedron, when we compare the tetrahedra used for discretization of the octahedron (Fig. 7) with the tetrahedral parent.

According to the refinement method, the element volume of the octahedral child is related to the volume of the parent tetrahedron by

$$V_{\text{oct}} = \frac{V_{\text{tet}}}{2} \tag{4}$$

where V_{oct} is the octahedral and V_{tet} is the tetrahedral volume. The volume of a discretization tetrahedron is given by

$$V_{\text{tet,disc}} = \frac{V_{\text{oct}}}{8} = \frac{V_{\text{tet,parent}}}{16} \tag{5}$$

where $V_{\text{tet,disc}}$ is the volume of the discretization tetrahedron, V_{oct} is the volume of the octahedron, and $V_{\text{tet,parent}}$ is the volume of the parent tetrahedron. The maximum size of the discretization tetrahedra is

$$h_{\text{tet,disc}} \leq \frac{h_{\text{tet,parent}}}{2} \tag{6}$$

where $h_{\text{tet,disc}}$ is the overall size of the discretization tetrahedron and $h_{\text{tet,parent}}$ is the overall size of the parent tetrahedron. Properties (4)–(6) hold for all tetrahedral shapes because of the special node inserting method according to Rule 3.

Using (5) and (6) we find that the degradation of the element quality due to refinement of a tetrahedron into four tetrahedra and one octahedron is limited by

$$Q_{\text{tet,disc}} = \frac{V_{\text{tet,disc}}}{h^3_{\text{tet,disc}}} \geq \frac{\frac{V_{\text{tet,parent}}}{16}}{\left(\frac{h_{\text{tet,parent}}}{2}\right)^3} = \frac{V_{\text{tet,parent}}}{2(h_{\text{tet,parent}})^3} = \frac{1}{2} Q_{\text{tet,parent}} \tag{7}$$

where $Q_{\text{tet,disc}}$ is the quality of the discretizing tetrahedron and $Q_{\text{tet,parent}}$ is the quality of the parent tetrahedron.

A similar calculation can be done to estimate the maximum degradation of the element quality due refinement of an octahedron. In this case the element qualities of the discretization tetrahedra and the child tetrahedra are compared. The result shows that

$$Q_{\text{tet,child}} \geq \tfrac{1}{4} Q_{\text{tet,disc}} \tag{8}$$

where $Q_{\text{tet,disc}}$ is the quality of the discretizing tetrahedron and $Q_{\text{tet,child}}$ is the quality of the child tetrahedron.

In conclusion the maximum element quality degradation is given by a factor of 1/2 for the tetrahedron and by a factor of 1/4 for the octahedron.

Usually grid refinement is done only locally. This results in the fact that refined elements are adjacent to unrefined ones. These neighboring elements are called incompatible elements, and we define the order of incompatibility as the difference of the refinement levels of two adjacent elements (Fig 1). Considering a compatible point of an incompatible element (Fig. 1) we can determine the effect of the incompatibility on the nodal quality

$$Q^n_{\text{incomp}} \geq \left(\frac{V_{\text{child}}}{V_{\text{parent}}}\right)^k Q^n_{\text{comp}} \tag{9}$$

where Q^n_{comp} is the nodal quality without refinement, Q^n_{incomp} is the nodal quality after refinement, and k is the order of incompatibility.

Equation (9) shows that the amount of quality degradation strongly depends on the order of incompatibility. To limit the quality degradation to a small finite value, we restrict the order of incompatibility to one (also called one-irregularity condition), which means that elements which are incident to an incompatible node are not allowed to get refined again. For mixed-element decomposition with tetrahedra and octahedra it can be shown that $\min(V_{\text{child}}/V_{\text{parent}}) = 1/8$, which is also the limit for the degradation of the nodal quality.

Thus, the degradation of the element quality is limited to 1/4, and the degradation of the nodal quality is limited to 1/8. Additionally, from the shape preserving property of the algorithm it follows that our algorithm preserves the boundaries and interfaces and the structural anisotropy of the initial grid. Although the element quality degradation is limited, the refinement algorithm cannot assure other quality criteria like the Delaunay condition or the M-matrix property.

C. Algorithmic Details of the Implementation

For the implementation of the method we developed algorithms and data structures which allow local refinement and unrefinement (coarsening) as well as the control of the

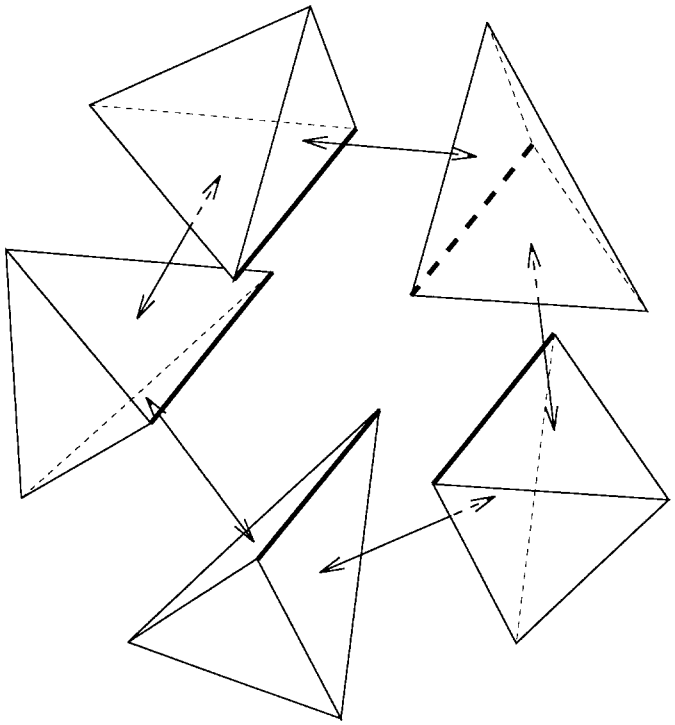


Fig. 8. Storage of the neighbor relations.

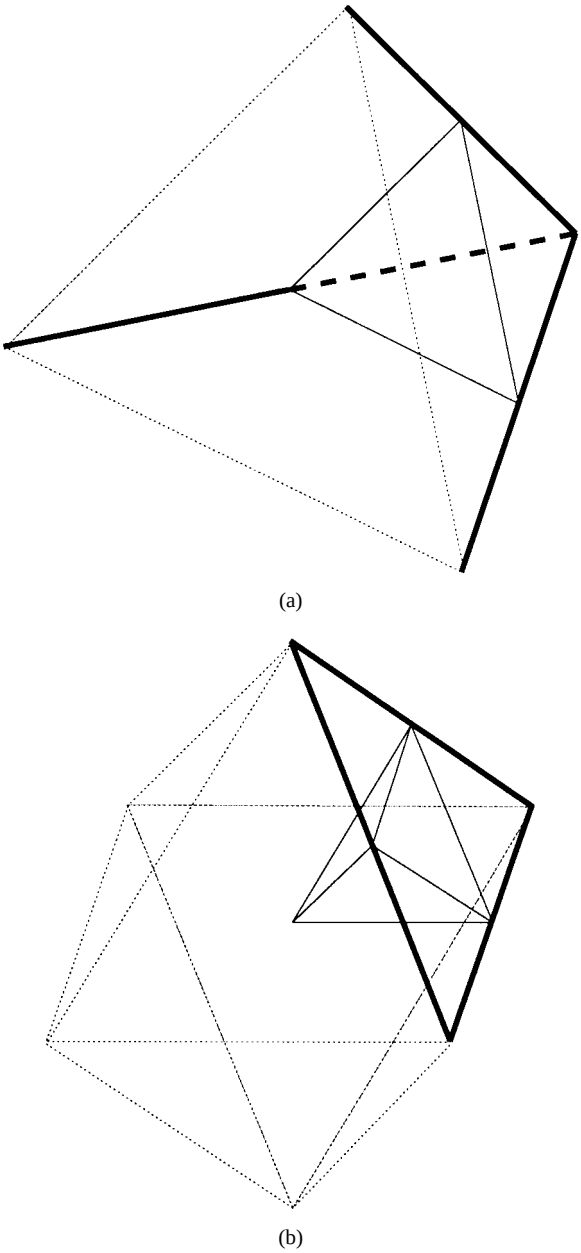
one-irregularity condition. We store the whole element tree, where the parent elements reference their children and each child references its parent (Fig. 5). Furthermore, each element references its neighbors opposite its faces (Fig. 8).

The storage of the neighboring relations allows direct determination of all neighboring elements which are incident to a face (face-neighbors). Besides referencing the neighboring element for each face, we additionally store the index of the opposite face within the neighboring element and the relative position of the two incident faces. Utilizing an element-based lookup-table which holds the two face-indexes sharing one line direct determination of all neighbors incident to a certain line (line-neighbors) is implemented without any searching algorithm. Fig. 8 shows an example for line-neighbors. The common line is drawn enhanced.

On splitting a line of an element a grid node has to be inserted at this line. This node, however, may exist already in one of the elements incident to this line. Before generating a new node we query all line-neighbors, whether their corresponding line is split and the node already exists or not. In this way we ensure, that each grid node exists exactly once. In turn, before we remove a node we first have to ensure that no other line-neighbor or face-neighbor references that point anymore.

The neighboring relations together with the parent–child relations are used to preserve the one-irregularity condition. This condition means, that two neighboring elements (line-neighbors as well as face-neighbors) have a difference in the number of subsequent refinements of at most one.

This condition is satisfied on refinement of a certain element, when we refine all "relevant neighbors" of the parent element. In Fig. 9 we show a tetrahedron with a tetrahedral parent and a tetrahedron with an octahedral parent, respectively. The "relevant neighbors" of the parent are the neighbors on those lines, which are (partially) common to the (in this case tetrahedral) element to refine. These "relevant" lines are drawn enhanced in Fig. 9.



(a)

(b)

Fig. 9. Examples for the relevant lines (enhanced) of parents.

The refinement for the relevant neighbors is done by means of a recursive call of the refinement function for them. Hence, the recursion depth is limited to the number of subsequent parents of the considered element, usually a small number.

On local grid coarsening we replace all the children of an element by the element itself, and therefore all children must be coarseable, i.e., the local discretization error is small enough. Furthermore, we have to check all neighbors of all children, whether they have children to meet the one-irregularity condition. In this case, we have to check these neighbors for possible coarsening, once again by means of a recursion. Here the recursion depth is limited to the number of subsequent refinements within the neighboring elements.

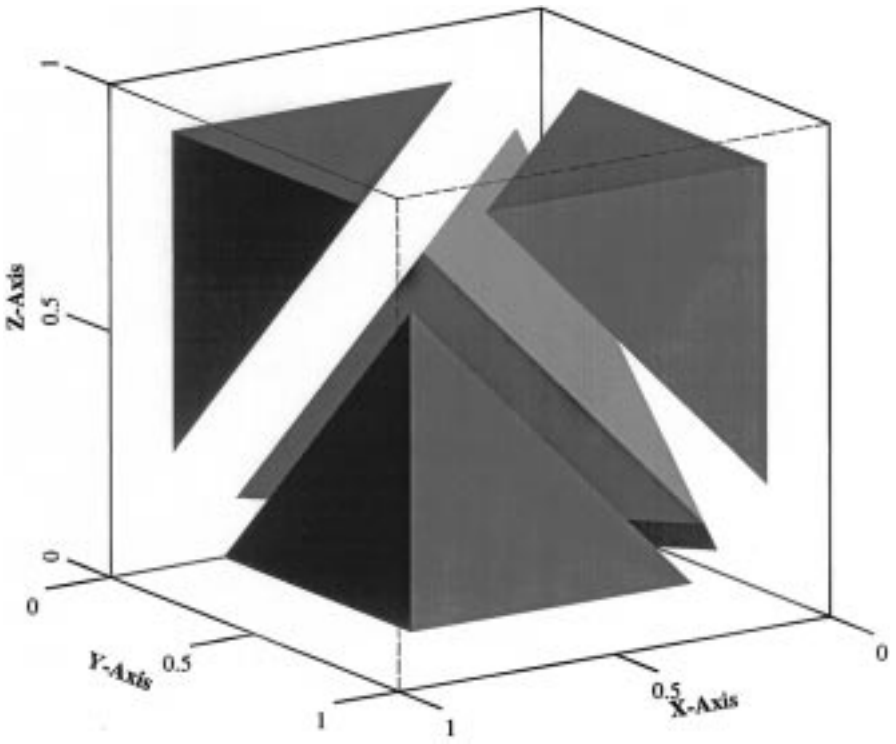


Fig. 10. Initial grid for cube.

The use of recursions for refinement and coarsening allows for the adaption of to the whole grid with one single sweep over just those elements which are marked for refinement and coarsening, respectively. Therefore, the effort for grid adaptation is approximately proportional to the number of changed elements. Without the recursions the one-irregularity condition would require sweeping over all grid elements several times until no more changes occur, which is n times in the worst case, when n denotes the number of subsequent refinement levels. Thus, the effort for our adaptation algorithm depends linearly on the number of changed elements, which are usually just a few percent of the total number of elements in the grid.

The element topologies are represented by lookup tables which hold, e.g., the relevant lines of a child or how many children and which kind of children a certain element type has, etc. These lookup tables are generated automatically based on simple element descriptions, which allow an easy extension of the refinement and coarsening algorithms for all relevant kinds of 1-D, 2-D, and 3-D elements.

All operations used within the adaptation algorithm are based on pure topological operations. Therefore, the algorithm does not rely on the geographical coordinates of the nodes and permits moving grid methods as well. Clearly, in such cases the element quality is determined by the moving grid algorithm itself [12]. Using adequate equations, our adaptation algorithm is also suitable for moving grid problems arising from process simulation.

D. An Illustrating Example

The mechanisms described are illustrated by means of two successive refinement steps of a cubic domain consisting of five tetrahedra as initial grid (Fig. 10). The grid elements are plotted shrunk, which allows to have some insight into the grid. For the first refinement step we trigger the refinement only for the right-uppermost tetrahedron (Fig. 11). This tetrahedron is just replaced by its children: four small tetrahedra and one octahedron. The one-irregularity condition has not been violated, thus no propagation of refinement occurs. For the second refinement step, we tag the foremost child of the just refined tetrahedron. The result shows the propagation of refinement due to the one-irregularity condition and verifies the correctness of the above algorithms (Fig. 12). Furthermore, the shape preserving property becomes obvious for the octahedra and tetrahedra which are irregular in this case.

IV. The Thermal Diffusion Problem

A. Governing Physical Equations

The set of diffusion equations (10) and (11) and the boundary conditions (12) allow the implementation of almost arbitrary physical models

$$\frac{\partial C_k}{\partial t} = \operatorname{div} \vec{J}_k + \sum_{i=1}^{n} \gamma_{ki}, \qquad k = 1, \cdots, n \tag{10}$$

$$\vec{J}_k = \sum_{i=1}^{n} \alpha_{ki} \operatorname{grad} C_i, \qquad k = 1, \cdots, n \tag{11}$$

$$\sum_{i=1}^{n} \beta_{ki} (\vec{J}_i \cdot \vec{n}) + \Phi_k = 0, \qquad k = 1, \cdots, n \tag{12}$$

where C_k are the concentrations (dependent variables) and n is the number of quantities (equations). The coefficients γ_{ki} allow modeling of recombination/generation terms. In

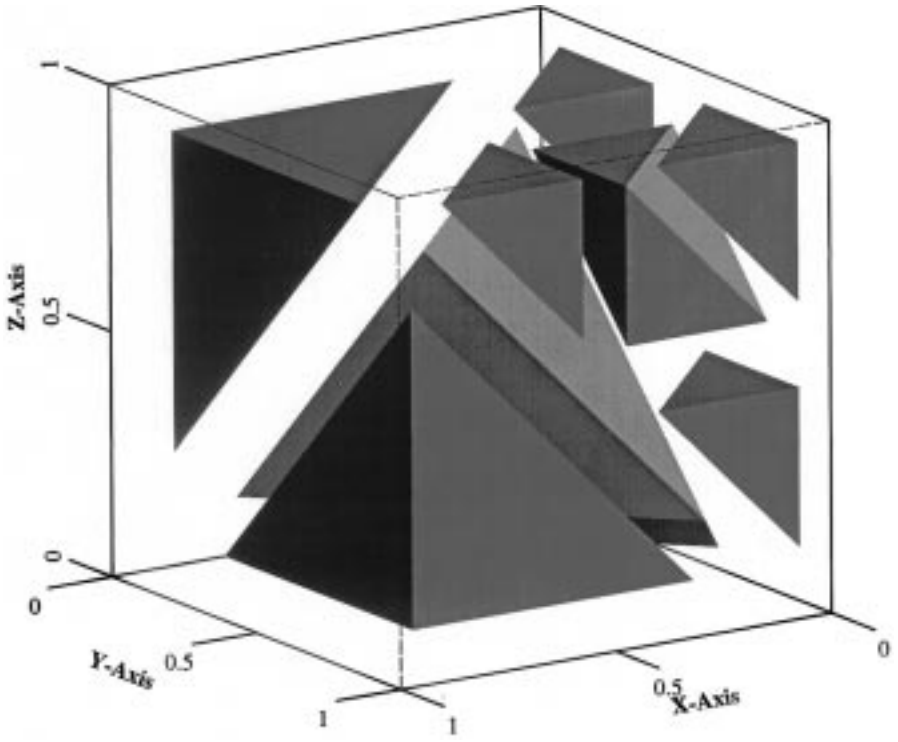


Fig. 11. Grid after first refinement step.

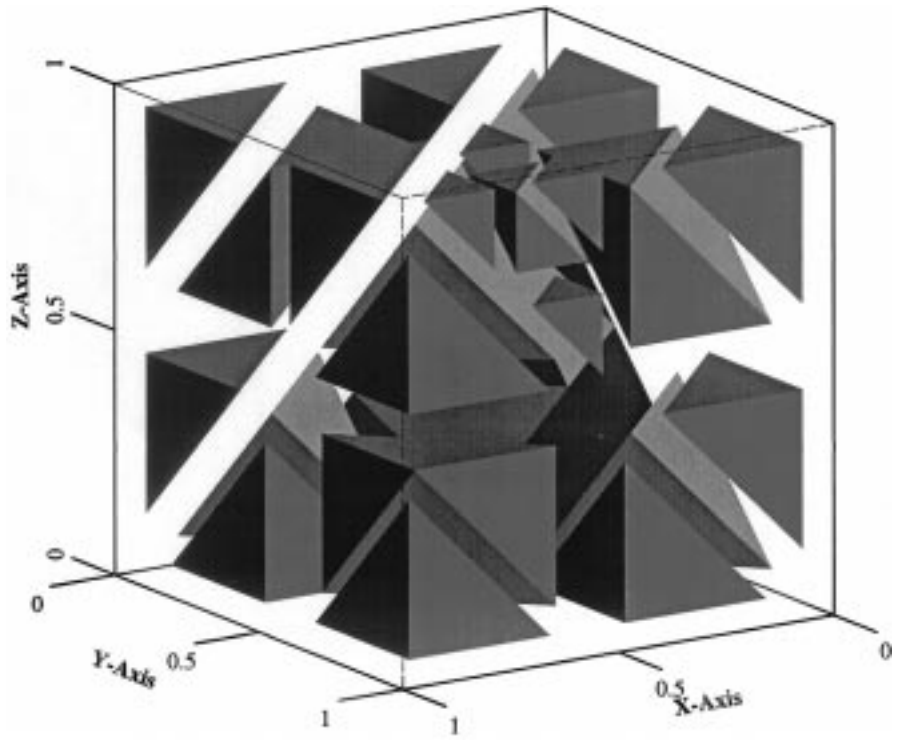


Fig. 12. Grid after second refinement step.

the flux definition (11) the flux $\vec{J}_k$ may depend on the gradient of all other quantities according the coefficients α_{ki}. All coefficients $(\alpha_{ki}, \beta_{ki}, \gamma_{ki}, \Phi_k)$ may be functions of temperature, time, and the spatial coordinates. All coefficients but β_{ki} may be functions of the dependent variables C_k. In the boundary conditions (12) $\vec{J}_i \cdot \vec{n}$ denotes the flux component perpendicular to the boundary for the quantity C_i. Thus, all practically relevant boundary conditions can be treated.

B. Spatial Discretization

For discretization of the above equations, we use the finite element method. The tetrahedron with linear shape functions

is chosen as discretizing element. We build a weak form of the above equations by means of the Galerkin weighted residuals method. The numerical computation of the weak form is done by Gaussian integration at one integration point in the center of the tetrahedron for (10) and (11) and at one Gaussian point in the center of the boundary triangles for (12). The Gaussian integration methods requires an at least piecewise continuous integrand. Otherwise the numerical integration would not converge to the exact value on reducing the element size toward zero. As the weak form contains first-order spatial derivatives of the shape functions, the shape functions have to fulfill the C_0-continuity condition. This leads to a special treatment for the incompatible nodes occurring at transitions between refined and unrefined neighboring elements. The solution values at such nodes have to be interpolated using the parents' shape functions, which enforces the C_0-continuity condition. These values depend on the solution values of the parents' nodes and are not free variables. Therefore, the corresponding equations values are pre-eliminated from the system matrix before solving the global linear system. This effort is needed in addition to the standard finite element assembly.

The application of the mass lumping technique [13] for the integration of the time derivatives as well as for generation/recombination terms improves the convergence rate of the iterative solver considerably and additionally reduces the discretization errors due to error cancellation effects [13].

C. Estimation of Discretization Errors

Utilization of the grid adaptation algorithm requires a suitable error estimation for determining the regions of refinement and unrefinement, respectively. We use an error-estimator which is originally mentioned in [14] and provides a simple and effective basis for adaptive refinement algorithms.

Although the overall integral of the weighted residuals of a solution of the weak form is zero, locally the residuum is nonzero. By locally smoothing the solution these local residuals are reduced, and therefore, the smooth approximation is (at least locally) more accurate than the original solution, which also holds for the gradient of the solution. This fact is used for estimation of the discretization error, which is computed from the difference between the original solution and the smoothed approximation. A proof for the stability of this error estimator is given in [15].

In our approach we use a piecewise linear finite element approximation with piecewise constant gradients. We implemented a gradient smoothing by means of a least-square fit of a piecewise linear gradient $(\nabla C)_{\rm lin}$ to a piecewise constant gradient $(\nabla C)_{\rm const}$ by minimizing

$$\Pi = \int_V [(\nabla C)_{\rm lin} - (\nabla C)_{\rm const}]^2 \, dV. \tag{13}$$

This allows the use of the same type of finite elements for gradient smoothing as for the finite element solution itself. To obtain the piecewise linear gradient approximation, the mass-matrix of the finite element formulation has to be inverted. Using the mass-lumping technique, this matrix degenerates to a diagonal matrix and the effort of obtaining a higher order approximation is kept low as the effort is approximately proportional to the number of elements.

Finally, the gradient error can be estimated with good accuracy as

$$e = ((\nabla C)_{\rm lin} - (\nabla C)_{\rm const}) \tag{14}$$

which is used as refinement and coarsening criterion.

D. Time Discretization and Time Step Control

The discretization in time is done by the standard finite differences method for a backward-Euler integration scheme. Error estimation is used to test the accuracy of the previous time step as well as to predict the size of the next time step.

Similar to the error estimation in space, we use a higher order approximation in time, too. The piecewise linear evolution of the solution is approximated parabolically, and the difference is used for error estimation. When the estimated discretization error is small enough, the time step is accepted. Otherwise we reduce the size of the time step and compute the solution again.

Once a time step is accepted, the parabolic approximation in time is used for estimation of the size of the next time step. We assume that the evolution of the solution follows the extrapolation of the parabolic approximation. According to the extrapolated discretization error, the next time step is chosen. Additionally, this parabolic extrapolation is used as an initial condition for the solution of the next time step which reduces the average number of nonlinear iterations per time step by one.

E. Grid Adaptation Strategy

On starting a diffusion simulation, the initial grid is refined according to the initial solution, e.g., a doping profile computed by an ion implantation simulator. First the nodal values of the initial grid are interpolated from the initial profile. Then a loop is formed where we estimate the discretization error and adapt the grid locally. On new generated nodes we get the solution values again by interpolation from the initial solution. This loop terminates, when the desired accuracy is reached over the whole region.

Throughout the simulation, the grid criterion is checked after the solution of each time step and the grid becomes adapted accordingly. On local refinement new nodes are generated. The solution values on these nodes are computed by interpolation from the solution at the parent elements. We use an arithmetic interpolation as an interpolation function.

This strategy ensures that the discretization errors remain within the specified bounds during the simulation. Additional benefits stem from the fact that the profiles become smoother during the diffusion process, and lower discretization errors allow local grid coarsening, which reduces the number of grid nodes. Thus, every time step is computed with a nearly optimal grid and a well-known discretization error.

F. Solution of the Nonlinear Equation Systems

For solution of the nonlinear equation systems, we implemented a damped Newton iteration scheme for the coupled

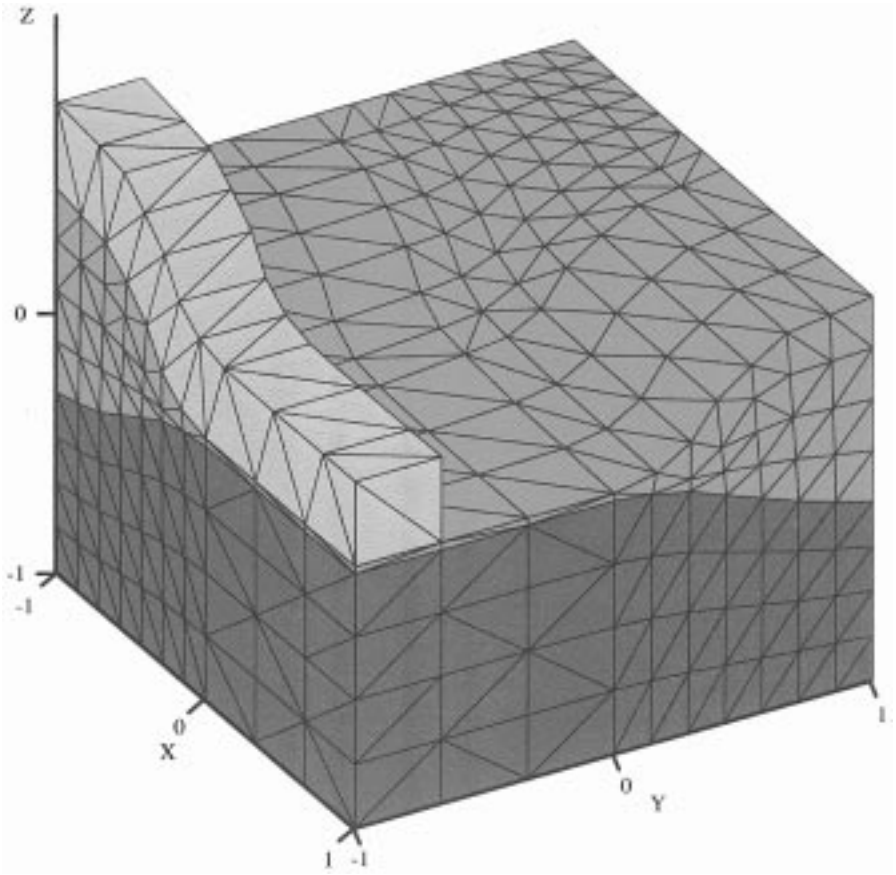


Fig. 13. Initial grid for the LOCOS-structure.

equations. The resulting linear equation systems are solved iteratively by means of a BiCGStab-solver [16] with incomplete Gauss-elimination for preconditioning [17]. For a particular time step the initial condition for the Newton iteration scheme is obtained by quadratic extrapolation of the solution of the previous time steps (cf. Section IV-D).

On computation of the Jacobian matrix for an incompatible element (cf. Section IV-B), we pre-eliminate the dependent variables of the incompatible nodes before we add this matrix to the global stiffness matrix.

V. Annealing of a MOSFET Source/Drain Implant

The formation of source and drain is an intrinsically 3-D problem in minimum size transistor design, which raises the need for full 3-D simulation.

For our example we simulated the ion implantation and the following annealing step for the lightly doped drain (LDD) LDD-implant in a conventional LOCOS-structure. Fig. 13 shows a corner of the considered example.

The Boron implantation has been computed by the Monte Carlo ion implantation module [18] of the VISTA framework [19] for an energy of 20 keV, a dose of $3 \cdot 10^{12}$ cm^{-2}, and a tilt angle of 7°. The wafer is oriented in $\langle 100 \rangle$ direction and is covered by a screening oxide of 30-nm thickness. The surface distribution of the implanted boron profile is shown in Fig. 14. The grid consists of 73 739 elements and 36 163 nodes. It has been generated by adaptive refinement of the initial grid shown in Fig. 13 according to the implanted profile, where the error limit has been set to 1% dose error.

First we show the simulation of a technologically relevant annealing step at 875°C for a time of 20 min in inert ambient. As physical model we used a concentration-dependent diffusion model with no point defects included. To save computational resources, the annealing step has been simulated just for the silicon region by assuming a zero-flux boundary condition between silicon and oxide. The computational grid for the silicon region consists of 22 549 elements and 11 898 free nodes at the beginning of the simulation (Fig. 15). As diffusion advances the profile moves deeper, and therefore at the diffusion front additional refinement becomes necessary. This results in grids consisting of up to 23 721 elements and 12 743 nodes. The simulation finished after six time steps and totally 18 Newton-iterations with a final grid consisting of 23 553 elements and 12 672 nodes (Fig. 16) and the used computational resources were 23-MB memory and 9-min CPU-time on an HP9000-735 workstation.

For this simulation the diffusion length is small and the profile does not change its shape substantially. Therefore, less grid adaptation is required. To give a demonstration of the capabilities of the automatic grid adaptation, we performed a simulation for the same structure and initial profile but with 1050°C and 30 min as annealing conditions. In this case substantial diffusion occurs and more grid adaptation was necessary. The largest grid consisted of 24 030 elements

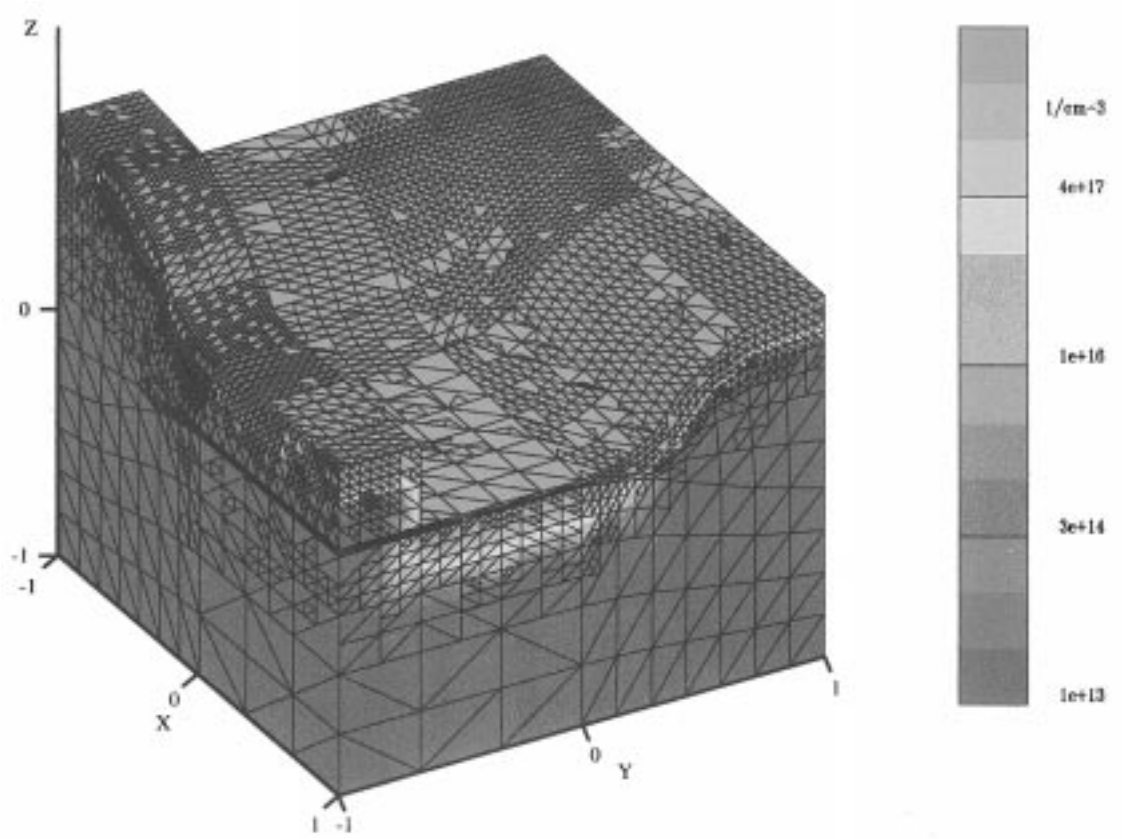


Fig. 14. Surface concentration of the boron profile after ion implantation.

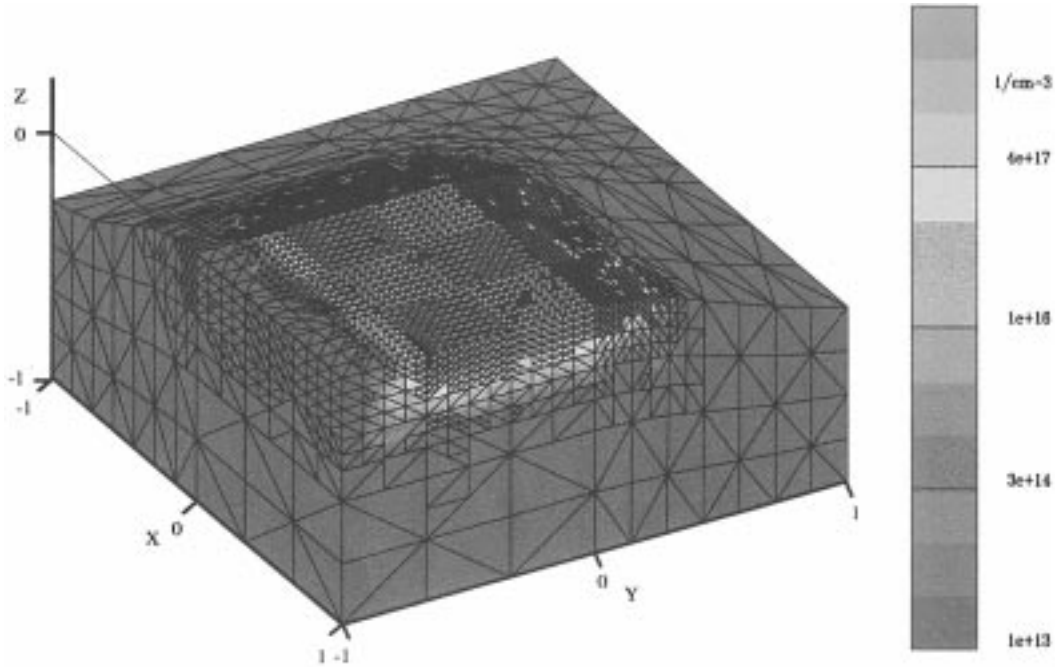


Fig. 15. Grid for simulation region after initial adaptation to implanted boron profile.

and 12 892 nodes due to additional refinement at the moving diffusion front. The further diffusion reduced the profile gradients and the grid could be reduced to 18 485 elements and 10 437 nodes at the end of the simulation after 33 time steps and totally 71 Newton-iterations (Fig. 17). In this case the used computational resources were 25-MB memory and 30-min CPU-time.

This example shows one major benefit of fully adaptive gridding: there is no need for the user to take care about the grid density except to specify the desired accuracy.

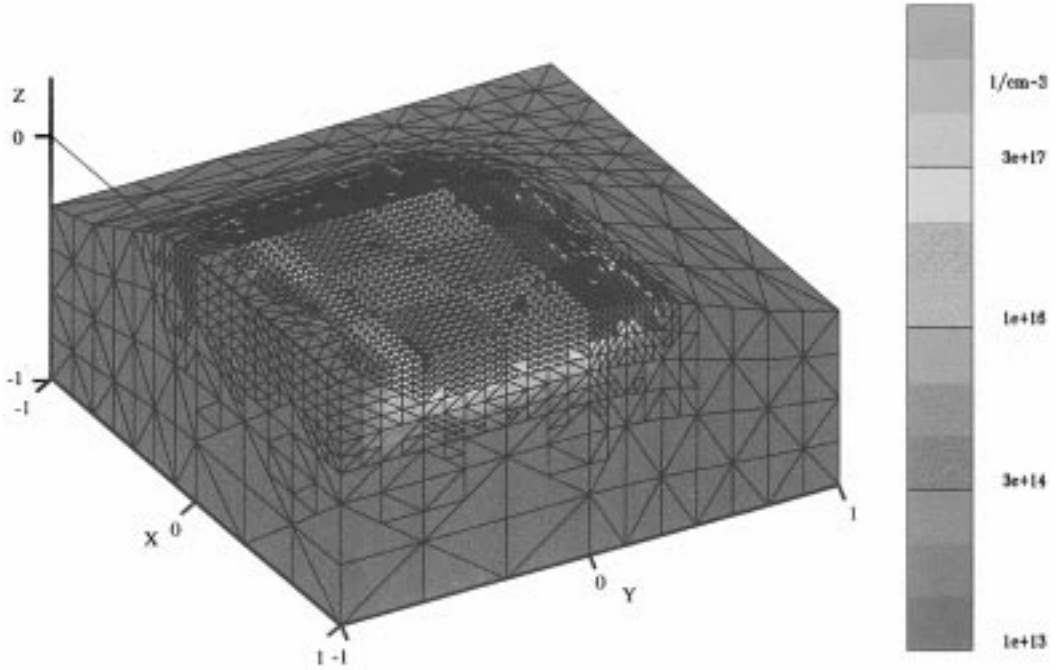


Fig. 16. Boron profile in the silicon block after annealing at 875°C, 20′.

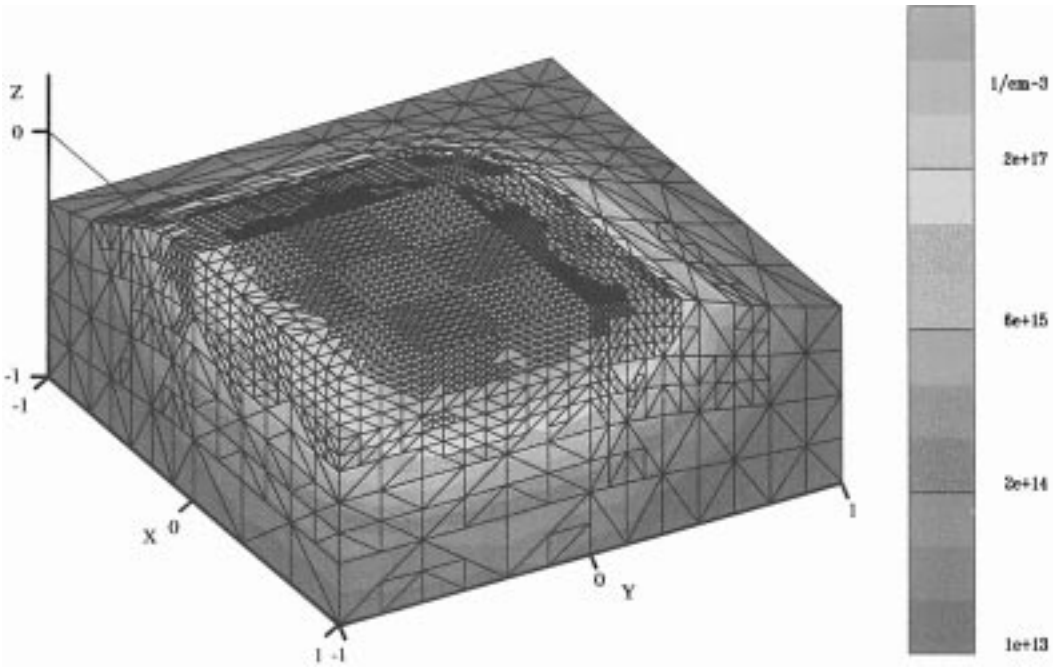


Fig. 17. Boron profile in the silicon block after annealing at 1050°C, 30′.

VI. Conclusion

A method for automatic adaptation of 3-D grids has been presented which allows to locally refine and coarsen the computational grid. We have shown that the method preserves the quality of the initial grid elements as well as local anisotropies and that boundaries and interfaces of the computational domain are respected.

The adaptation is based on an efficient estimation of discretization errors and therefore automatically produces nearly optimal grids for a certain discretization error limit. The coupling of the error estimation and local grid adaptation dispenses the user from the difficult and time-consuming task of checking the grid density.

The computational effort for checking the discretization error depends linearly on the number of grid elements, and the effort for grid adaptation depends linearly on the number of adapted elements. Thus, the presented methods and algorithms combine computational efficiency and robustness of the simulation.

ACKNOWLEDGMENT

The authors acknowledge important support by Austria Mikro Systeme AG, Unterpremstätten, Austria; Digital Equipment Corp., Hudson, USA; Hitachi Ltd., Tokyo, Japan; LSI Logic Corp., Milpitas, USA; Motorola Inc., Austin, USA; National Semiconductor Corp., Santa Clara, USA; Philips B.V., Eindhoven, The Netherlands; Siemens AG, München, Germany; and Sony Corp., Atsugi, Japan.

REFERENCES

[1] W. Bohmayr, A. Burenkov, J. Lorenz, H. Ryssel, and S. Selberherr, "Trajectory split method for Monte Carlo simulation of ion implantation," *IEEE Trans. Semiconduct. Manufact.*, vol. 8, pp. 402–407, Nov. 1995.

[2] A. M. Mazzone and G. Rocca, "Three-dimensional Monte Carlo simulations—Part I: Implanted profiles for dopants in submicron devices," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 64–71, Jan. 1984.

[3] E. Strasser and S. Selberherr, "Algorithms and models for cellular based topography simulation," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 1104–1114, Sept. 1995.

[4] E. W. Scheckler and A. R. Neureuther, "Models and algorithms for three-dimensional topography simulation with SAMPLE-3D," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 219–230, 1994.

[5] M. E. Law and S. Cea, "Three-dimensional simulation of thermal processes," in *Three-Dimensional Process Simulation*, J. Lorenz, Ed. Wein, Germany: Springer, 1995, pp. 77–94.

[6] B. Baccus, S. Bozek, V. Senez, and Z. Z. Wang, "3D process simulation at IEMN/ISEN," in *Three-Dimensional Process Simulation*, J. Lorenz, Ed. Wein, Germany: Springer, 1995, pp. 98–108.

[7] P. Conti, N. Hitschfield, and W. Fitchner, "ω—An octree-based mixed element grid allocation for adatpive 3D device simulation," in *Proc. Workshop Numerical Modeling of Processes and Devices for Integrated Circuits NUPAD III*, Honolulu, HI, 1990, pp. 25–26.

[8] N. Hitschfeld, P. Conti, and W. Fichtner, "Mixed element trees: A generalization of modified octrees for the generations of meshes for the simulation of complex 3-D semiconductor device structures," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 1714–1725, Nov. 1993.

[9] B. Joe, "Construction of three-dimensional improved-quality triangulations using local transformations," *SIAM J. Sci. Comput.*, vol. 16, no. 6, pp. 1292–1307, 1995.

[10] P. Fleischmann and S. Selberherr, "A new approach to fully unstructured three-dimensional delaunay mesh generation with improved element quality," in *Proc. Int. Conf. Simulation of Semiconductor Processes and Devices*, Tokyo, Japan, 1996, pp. 129–130.

[11] R. E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Frontiers in Applied Mathematics*, vol. 7. Philadelphia, PA: SIAM, 1990.

[12] R. E. Bank and R. K. Smith, "Mesh smoothing using *a posteriori* error estimates," *SIAM J. Numer. Anal.*, vol. 34, no. 3, pp. 979–997, 1997.

[13] O. C. Zienkiewicz and R. L. Taylor, *Solid and Fluid Mechanics, Dynamics and Non-Linearity*, 4th ed. (The Finite Element Method, vol. 2). London, U.K.: McGraw-Hill, 1991.

[14] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineering analysis," *Int. J. Numer. Meth. Eng.*, vol. 24, pp. 337–357, 1987.

[15] M. Ainsworth, J. Z. Zhu, A. W. Craig, and O. C. Zienkiewicz, "Analysis of the Zienkiewicz-Zhu *a posteriori* error estimator in the finite element method," *Int. J. Number. Meth. Eng.*, vol. 28, pp. 2161–2174, 1989.

[16] H. A. van der Vorst, "BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 12, pp. 631–644, 1992.

[17] O. Heinreichsberger, M. Thurner, and S. Selberherr, "Practical use of a hierarchical linear solver concept for 3D MOS device simulation," in *Simulation of Semiconductor Devices and Processes*, S. Selberherr, H. Stippel, and E. Strasser, Eds. Wien, Germany: Springer, 1993, vol. 5, pp. 85–88.

[18] W. Bohmayr, A. Burenkov, J. Lorenz, H. Ryssel, and S. Selberherr, "Statistical accuracy and cpu time characteristic of three trajectory split methods for Monte Carlo simulation of ion implantation," in *Simulation of Semiconductor Devices and Processes*, H. Ryssel and P. Pichler, Eds. Wien, Germany: Springer, 1995, vol. 6, pp. 492–495.

[19] S. Halama, Ch. Pichler, G. Rieger, G. Schrom, T. Simlinger, and S. Selberherr, "Vista—User interface, task level, and tool integration," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 1208–1222, Oct. 1995.

[20] J. Lorenz, Ed., *Three-Dimensional Process Simulation*. Wien, Germany: Springer, 1995.

Ernst Leitner was born in Gmunden, Austria, in 1968. He studied electrical engineering at the Technische Universität Wien. He received the degree of Diplomingenieur in 1992 from the Technische Universit at Wien and the doctoral degree in technical sciences from the Technical University Vienna in 1998.

In the autumn of 1994 he held a visiting research position at Philips, Eindhoven, The Netherlands. His work is focused on three-dimensional process simulation, adaptive gridding, and the basic physics of transport phenomena in solids.

Siegfried Selberherr (M'79–SM'84–F'93) was born in Klosterneuburg, Austria, in 1955. He received the degree of "Diplomingenieur" in electrical engineering and the doctoral degree in technical sciences from the Technical University Vienna in 1978 and 1981, respectively.

Since 1981 he has been with the Technical University Vienna as a Professor. He has been holding the "venia docendi" on Computer-Aided Design since 1984. He has been the head of the Institut für Mikroelektronik since 1988. His current research interests are modeling and simulation of problems for microelectronics engineering.