5-1 (Invited)

New Developments and Old Problems in Grid Generation and Adaptation for TCAD Applications

W. Fichtner, J. Krause, B. Schmithüsen, L. Villablanca
Integrated Systems Laboratory, ETH Zürich
Gloriastr. 35, CH-8092 Zürich, Switzerland

TCAD applications such as multi-dimensional process and device simulations or electromagnetic field calculations rely to a large part on a stable and accurate solution of the underlying physical equations. Well-known examples are the set of six equations that govern hydrodynamic transport in deep submicron devices, or the ensemble of equations for coupled pairdiffusion.

The quality and sometimes even existence of a numerical solution depends to an overwhelming part on the spatial discretization used in a particular calculation. While mesh quality is an old and well-known problem, it is very often ignored by the user.

Driven by the need to simulate increasingly complex semiconductor processes and devices, meshing has certainly moved to the forefront of interest in the TCAD community.

Here we want to present a review of new developments in multi-dimensional mesh generation for TCAD applications and concentrate on several specific areas such as promising new algorithms for device mesh generation and grid adaptation for process and device applications.

Moving Boundaries in Process Simulation

An interesting problem in 3D process simulation is related to the geometrical and topological changes which affect the structure during some simulation steps. On a deposition step, for instance, a new layer of a certain material is placed on top of an existing one, the material is then grown until it reaches a certain thickness. The reverse effect can be observed when a layer consumed by an etching step, leaving a hole in the layer, which sometimes can go all the way through the the underlying material. Mesh generation for 3D process simulation incorporates then a new issue, namely the dynamic nature of the mesh. The problem is currently being attacked using several techniques, such as the string algorithm and the level set algorithm. The string algorithm is an extension of a two-dimensional technique, which uses a surface representation to perform the deformation steps. In two dimensions, the surface can be seen as a set of closed loops or strings which can be deformed at will. If the string deformation causes the formation of loops, they are cut out from the string and the algorithm continues. In three dimensions however, the algorithms for detecting and removing surface loops are much more difficult to implement, existing practically no reliable software implementation of them.

The level set algorithm [1] is used to describe the position in time of the surface of a simple domain. It has the advantage that it is relatively simple, robust and naturally handles changes in the topology of the domain. Unfortunately, it fails to handle certain structures if the domain composed of multiple regions. Besides, the current implementations of this algorithm need a lot of memory if they want to describe small features, such as thin layers in a large domain.

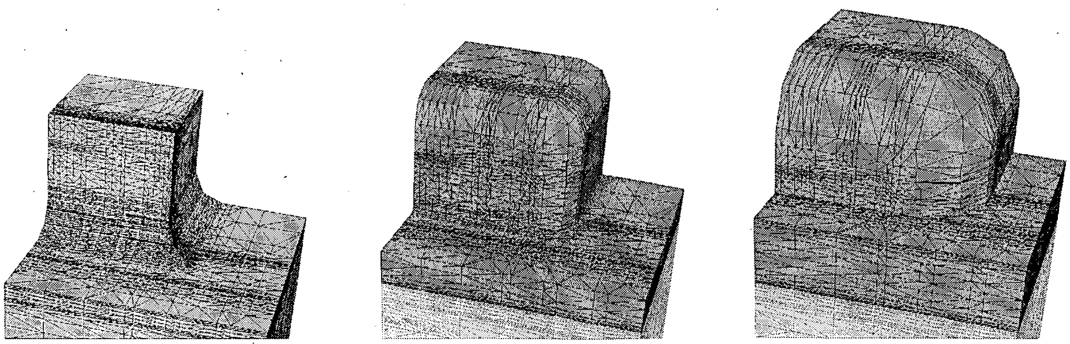

Figure 1: 3D numerical process simulation, evolution of grid during deposition step

The algorithm presented here [2] makes use of the volumes of the elements inside a given domain. The surface of the mesh is moved until some mesh elements become flat (zero volume), the flat element is corrected, and then the surface continues its trajectory (Fig. 1). The advantage of this approach is that collisions between surfaces are easily detected and corrected. Once a flat element separating two regions is solved, the two regions get immediately merged, making topological changes a trivial matter.

The current implementation takes an input mesh together with the displacements for every point in the mesh. The surface displacements are pre-processed to avoid unnecessary collisions or twists. Once the surface is pre-processed, every point is moved at the same rate. A fraction of the full displacement is calculated so that the first cluster of elements becomes flat. The flat elements are supposed to lie all on the same plane and can then be eliminated one by one.

After every point has reached its final destination, there is a post process which is in charge of removing small features from the mesh. Every small edge, facet or element is removed using the same moving mesh algorithm. For a small edge, for instance, a destination is assigned to one end point, such that it coincides with the current position of the other end point. Then the point is moved step-wise, solving the flat elements found on the way until the final destination is reached and the small edge is gone.

Grid Generation for Device Simulation

In the simulation of semiconductor processes and devices it can be necessary to generate surface parallel meshes. One important example occurs in MOS transistors where the electron current flows along the silicon surface underneath a gate. It is desired beneficial in terms of accuracy to have rather long mesh edges parallel to the interface and rather small edges orthogonal to those currents. For most of the devices quadtree techniques have been used with big success [3]. If the interface is not axis aligned, however, a quadtree based approach does not generate meshes of this quality, resulting in a larger numerical error or in convergence problems during equation solution.

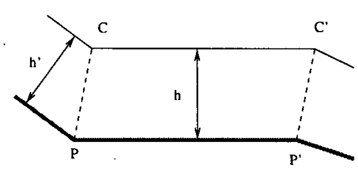

Figure 2: Quadrilateral creation for normal offsetting

As a new development we present a modified advancing front grid generator that inserts surface parallel mesh lines; the interior of the region is filled with layers of nearly rectangular quadrilaterals (Fig. 2), and not triangles as in conventional advancing front generators. Here we follow reference [4], but we use a different point location scheme, in

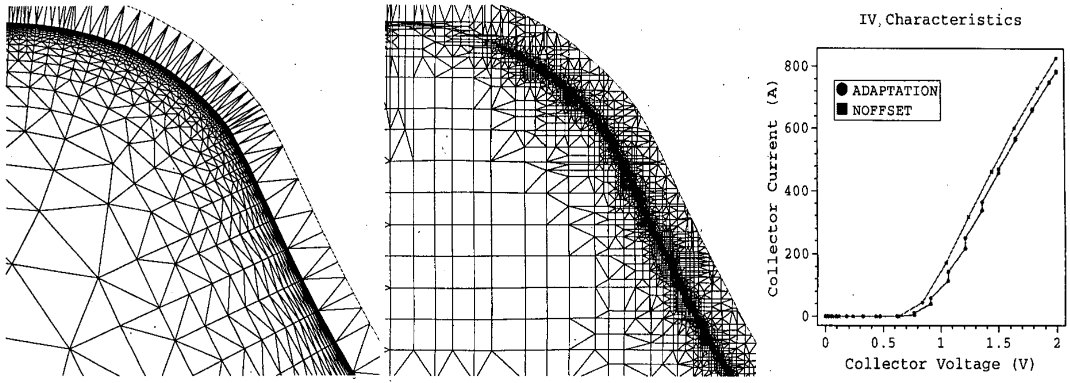

Figure 3: Grids from both algorithms and IV-characteristics

the sense that the opposite edge of the quadrilateral is kept parallel to the interface if possible. The price that has to be paid for these boundary conforming meshes is the computational expensive global intersection test. It is necessary to detect when the front collides with the opposite front; we use an Alternating Digital Tree (ADT) for geometrical searching that has a computational complexity of $O(n \log n)$.

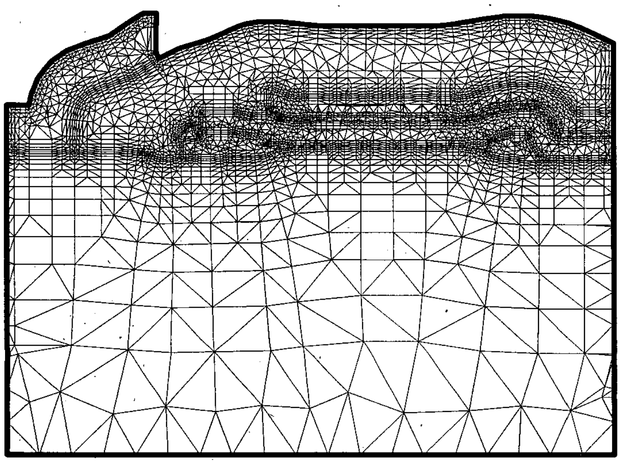
Figure 4: Complex geometry meshed with normal offsetting

Another problem for these type of generators is the quality degradation of the element when the front evolves. Here we relax the requirement to build quadrilateral and we insert triangles.

In a post-processing step the remaining polygon and the quadrilaterals that are not exactly rectangular are triangulated. Additionally the final quality is enhanced by smoothing and delaunizing the grid.

The normal offsetting algorithm is capable of meshing even complex geometries like in Fig. 4. An impressive example is the IGB transistor: the current flow under the interface substrate to gate oxide can be simulated more precisely with a mesh which follows the curved interface.

Grid Adaptation for Device Simulation

In contrast to process simulation, where the simulation domain changes according to the changing shape and material fronts, for many applications in device simulation the material boundaries are regarded as fixed, neglecting mechanical and other effects. Nevertheless grid adaptation in device simulation is of growing interest with the increasing complexity of the devices and for simulations over a wide range of operation conditions. Though grid adaptation is common practice in other fields of application and in the last years an intensive field of research (e.g. [5]), only a few approaches have been made for the device equations. The main difficulties stem from the singularly perturbed character of the underlying system of equations, exhibiting strong boundary and internal layers.

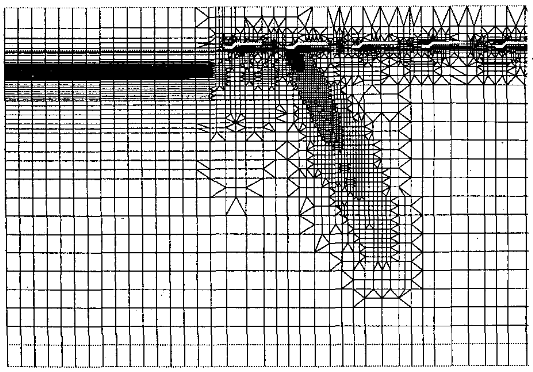

Figure 5: Adaptively generated mesh of a power diode structure at avalanche breakdown

Practical simulations on reasonable grids are mainly possible due to the well-known Scharfetter-Gummel box method discretization, guaranteeing a stable solution even on coarse grids, what is extensively used in practice. Standard adaptation techniques with respect to the (local) discretization error, developed for finite element discretizations of scalar elliptic PDEs, are not feasible in practice, because almost all refinements would be spent in the layers. Furthermore red and green refinements, normally used for finite element meshes, in general don't preserve the constrained Delaunay property required by the box methods.

A novel approach utilizes the dissipation rate of the system as the main adaptation instrument [6], guided by its close relationship to the integral terminal currents, which are in many cases the quantities of interest. Based on an octree technique, the approach allows anisotropic refinements using gradient information (though limited to the coordinate axis), and introduces a natural and physical weighting of the constituting equations (Fig. 5).

Both approaches were applied to an IGBT structure with a curved oxide interface. The octree based adaptation refines the channel region correctly according to the dissipation rate, but needs much more grid points compared with the normal offsetting mesh. On the other hand the normal offsetting approach can save these point by introducing interface parallel mesh lines, which are in this case also parallel to the current flow. The simulated collector currents are depicted in figure 3.

References

[1] J. A. Sethian, "Level Set Methods and Fast Marching Methods", Cambridge University Press, 1999.

[2] L. Villablanca "Mesh Generation for Three Dimensional Semiconductor Process Simulation", Ph.D. Thesis, Integrated Systems Laboratory, ETH Zürich, to be published.

[3] G. Garretón, L. Villablanca, N. Strecker, W. Fichtner; "Unified Grid Generation and Adaptation for Device Simulation"; *Proceedings of SISDEP'95, Erlangen, Germany*, pp 468 – 471, 1995.

[4] B. P. Johnston, J. M. Sullivan; "Fully Automatic two dimensional mesh generation using normal offsetting", *International Journal for Numerical Methods in Engineering* Vol 33, pp 425 – 442, 1992.

[5] R. Verfürth, "A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques," Wiley, Teubner, 1996.

[6] B. Schmithüsen, K. Gärtner, W. Fichtner, "Grid Adaptation for Device Simulation According to the Dissipation Rate," *Proceedings of SISPAD'98*, pp 197 – 200, 1998.