A 3D Mesh Generation Method for the Simulation of Semiconductor Processes and Devices

Klas Lilja, Victor Moroz, and Dan Wake

Avant! Corporation, TCAD Business Unit, 46871 Bayside Parkway, Fremont CA 94538
Phone: 1-510-413-8111 Fax:1-510-413-7743
E-mail : klas_lilja@avanticorp.com

ABSTRACT

We present an extended quadtree-octree mesh generation method which is well suited for semiconductor process and device simulation. The method can handle complicated geometries and moving boundaries. In order to describe boundaries and trace boundary movement, we apply the level set method in combination with a local transformation of the grid. The grid is modified only in the vicinity of the boundary, which keeps the computational work involved in the grid generation low, and avoids interpolation of solutions in most of the structure. Topological changes in the structure are easily handled, and the mesh quality and density are maintained throughout a simulation.

INTRODUCTION

The generation and adaption of the computational grid is a crucial part of the numerical simulation procedure. The quality of the grid determines the accuracy of the numerical solution, and the number of mesh points directly dictates the calculation speed. For 3D problems in particular, an efficient mesh generation algorithm which generates a high quality grid with as few mesh points as possible, is very desirable in order to keep the calculation times low. General qualities of a good mesh algorithm are the capability to resolve complicated geometries and to refine and un-refine the grid without severely distorting the elements.

The simulation of semiconductor processes and devices imposes several special requirements on the grid. The abrupt variations and layers that occur in the solution require that the mesh density can vary abruptly. Furthermore, these layers are often planar, and hence the refinement strategy should allow for anisotropic refinement. In fact, since all semiconductor devices start out from a flat wafer, many of these layers in the solutions are parallel to the original wafer surface.

The mesh requirements for semiconductor simulation favor the quadtree-octree mesh generation method [1], in which the mesh is created by successive subdivision of rectangles/bricks. These types of meshes allow for a very easy and natural anisotropic refinement and un-refinement, and any type of searches in the mesh are fast and algorithmically easy. Quadtree-octree mesh generation methods have been applied successfully to fixed grid problems in 3D device simulation [2] and diffusion simulation [3]. Despite these successful applications to the semiconductor simulation area, several problems remain to be solved.

When applying the octree method to non-planar geometries the basic algorithm must be extended. In the modified quadtree-octree method [1] this is done by the introduction of new nodes at the intersection points of the geometrical boundary with the quadtree-octree elements. The intersected elements are then tessellated into triangles/tetrahedra respecting the geometrical boundary. Variants of this method have later been applied to the simulation of semiconductors [2][3]. The method appears to have some inherent shortcomings, though, as was pointed out in [4].

Moving boundary problems, e.g. oxidation and silicidation, pose additional problems for the mesh generation. For 2D problems moving boundaries have traditionally been handled by the string algorithm, and 3D extensions to this algorithm have been presented [5]. However, for problems which do need a boundary conforming bulk mesh (oxidation, silicidation) the string algorithm requires the mesh to be unstructured, which prevents (or severely complicates) un-refinement and an-isotropic refinement. In addition the problem of "collapsing elements" (de-looping) in the string algorithm is considerably harder in 3D than in 2D.

An attractive alternative to the string method for moving boundary problems, is the level set method [6]. In this method the boundary is represented by a solution field (i.e. the zeroth level set of this field), and the grid is not required to be boundary conforming in principle. The method gives an elegant and stable boundary tracking, which naturally handles topological breaking and merging - problems which are hard to handle for any string algorithm. As will be discussed below the level set representation of the geometry also opens new possibilities for the use of a quadtree-octree mesh in complicated geometries and for moving boundaries.

QUADTREE-OCTREE ALGORITHM

The basic quadtree-octree algorithm successively subdivides rectangles or bricks into smaller elements. The version we have chosen subdivides an element into two identical elements along a plane given by the refinement criteria. These subdivisions introduce unconnected ("green") nodes at the edges of adjacent elements, which are resolved by subdividing these elements into triangles or tetrahedra, pyramids, and prisms. The basic algorithm works if any edge has a maximum of one unconnected node (1-irregular mesh). Fortunately, by balancing the mesh tree during construction [7], it can be ensured that only 1-irregular mesh trees are constructed, and the time consuming task of converting a mesh to be 1-irregular can be avoided.

The advantage of a mesh tree, over an unstructured grid, for refinement and un-refinement purposes is apparent. The tree structure allows us to easily remove or add leaves for appropriate un-refinement or refinement.

BOUNDARY TRANSFORMATION

One of the problems with the modified quadtree-octree method used to handle non-planar geometries in [1][2][3] is that it introduces new nodes at irregular positions in the grid. The tessellation patterns of the elements intersected by a boundary can become very complicated and the tessellation needs to be re-done every time further refinement, un-refinement, or boundary movement is performed. Problems also occur when the intersection points are close to a node in the tree, which can lead to reduced mesh quality. Instead of adding new nodes, we have chosen to transform the mesh by moving the nodes to new positions.

The algorithm we use to transform the grid uses a level set representation of the boundary. A boundary (T) is represented by the zero level set of a function $\phi(x)$ from R^n to R. $\phi(x) = d$, where d is the distance from the point x to T if the point is inside of T, and $\phi(x) = -d$ if the point is outside of T. This boundary description also allow for refinement of the geometry based on the level set function, and can allow for boundary movement through the grid (below).

A description of the geometry in the form of a level set function would in principle allow for a mathematical coordinate transformation similar to conformal mapping. Such a method preserves the mesh connectivity, but cannot handle topological changes in the mesh, and involves interpolation (or advective terms in the equations) globally. We have instead chosen to perform a local geometrical transformation of the mesh to make it conform to the boundary. Figure 1 gives a simple illustration of the procedure – the positions of the octree nodes are transformed to make the mesh conform to the boundary. This local geometrical transformation allows us to handle topological changes, maintains proper element shapes and mesh density, and is local, i.e., only the mesh in the vicinity of the surface is affected.

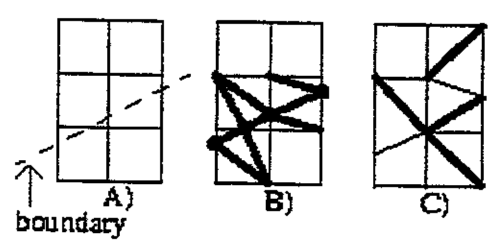

Figure 1. When boundary crosses the regular quadtree mesh (A), nodes can be added (B), or the mesh can be transformed (C). The thicker edges are the newly introduced ones.

The basic steps required to create a boundary conforming mesh via a local geometrical transformation are simple in principle:

a) Identify edges intersected by the boundary

b) Move the nodes closest to the boundary to a location on the boundary

c) Tessellate all elements affected by the above operations, while respecting the region boundaries

Note that in c) the tessellation patterns required are the same ones as for the regular quadtree-octree, except that any diagonal edges along the geometrical boundary must be maintained.

Steps b) and c) naturally must follow a set of rules in order to create a mesh with well shaped elements, e.g., a face with un-connected ("green") nodes may not cross a geometrical boundary, as well as geometrical criteria for a well shaped mesh.

MOVING BOUNDARY

Using the level set formalism allows us to move the boundary through a structure solving the level set equation:

$$\frac{\partial \varphi}{\partial t} + F|\nabla \varphi| = 0 \qquad (1)$$

or

$$\frac{\partial \varphi}{\partial t} + \bar{v} \cdot \nabla \varphi = 0 \qquad (2)$$

where F is the normal surface speed, and v the vector speed of the surface movement.

Several non trivial issues are involved in the moving boundary simulation using the level set equation, including the numerical discretization and the handling of triple lines and points. Our solution and implementation of these issues will be described elsewhere.

EQUATION SOLUTION

The mesh generation tool is an integral part of a semiconductor process and device modeling program, which solves a great variety of partial differential equations from the semiconductor modeling area [8]. For use with quadtree-octree meshes the most efficient way to discretize the partial differential equations is the finite volume discretization method, which can handle all the types of elements occurring in the mesh (triangles, rectangles, tetrahedra, prisms, and bricks) without further subdivisions.

EXAMPLES

We present a few typical application examples from the semiconductor process simulation area.

Rounded STI trench corners

Trench technology is widely used for semiconductor devices from memory to power devices. The presented mesh generation can be used to study the effect of different rounding radii on the mechanical stresses generated during processing, as well as electrical effects in device operation.

Mesh for LOCOS oxidation

As we discussed above the mesh generation algorithm we present can be used to simulate moving boundary processes such as oxidation and silicidation. We illustrate that here with the meshes from a LOCOS process.

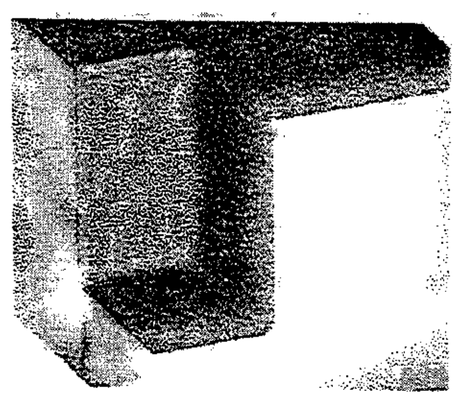

Figure 2. Trench structure. Oxide layer on top of a silicon trench with rounded corners.

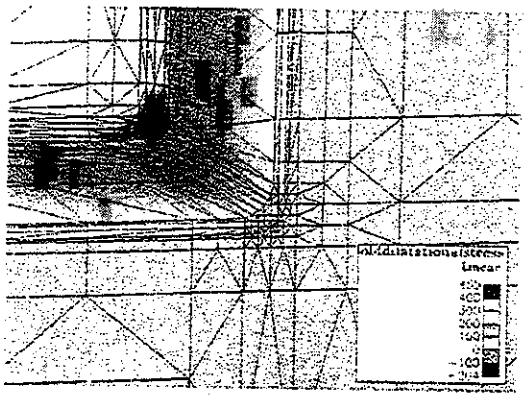

Figure 3. Mesh detail for the trench structure. The total mesh has 4310 nodes. The field shown in the figure is the dilatational component of the mechanical stress generated by a thermal ramping process. The maximum tensile stress for the rounded corners is 455Mpa, to compare with 537MPa for sharp (on the scale of this mesh) corners.

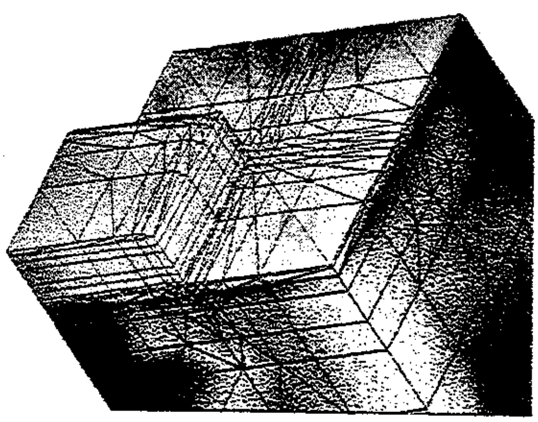

Figure 4. LOCOS mesh. Start point of the oxidation (1640 nodes).

3D MOSFET

The geometry of complex 3D device structures can be defined via process simulation, or specified via a structure definition input. Here we show a 3D MOSFET structure for studies of channel length and width effects. This structure was generated via the structure definition input. Refinement based on the source and drain doping profiles has been performed. The mesh contains 2870 mesh points.

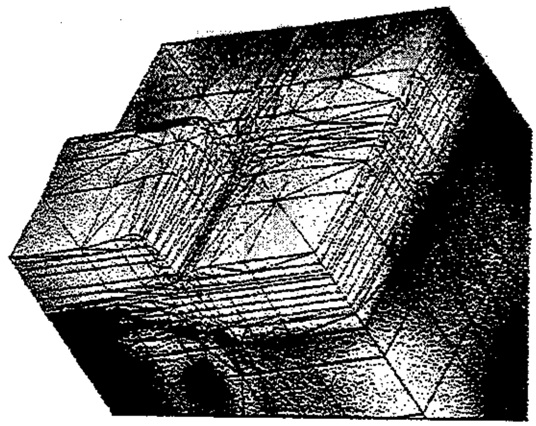

Figure 5. LOCOS mesh. Mesh after locos oxidation (2320 nodes).

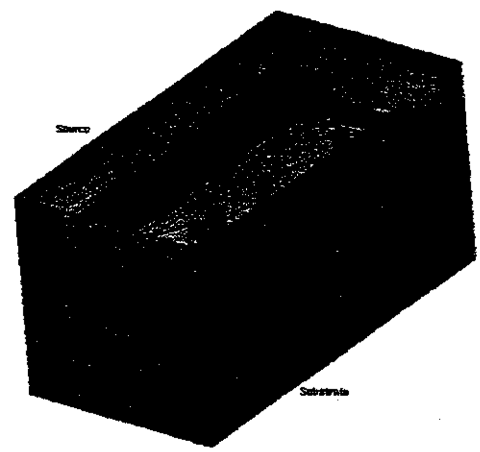

Figure 6. 3D MOSFET structure for device simulation.

Etch / Deposition

Via etch and deposition steps complicated structures can be generated and simulated. The deposition and etch steps manipulate the level set geometry description and, as described above, the mesh is generated to conform to this geometry description.

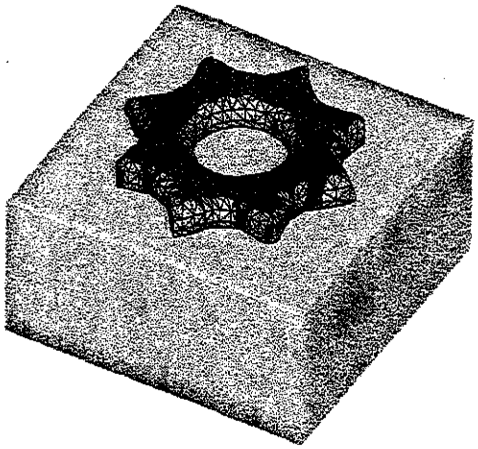

Figure 7. "Gear" structure generated via deposition and etch steps.

CONCLUSION

A modified quadtree-octree algorithm is presented, which can handle complicated geometries and moving boundaries. This type of mesh generation is well suited to the requirements of semiconductor simulation. It allows for anisotropic refinement and un-refinement and rapid changes in mesh density. The method is an integral part of a multidimensional, commercially available, semiconductor process and device simulation program called Taurus [8].

REFERENCES

[1] M. Yerry and M. Shephard, "Automatic Three-Dimensional Mesh Generation by the Modified-Octree Technique", Int. J. Numer. Methods Eng., 20, pp. 1965-1990, 1984

[2] P. Conti, W. Fichtner, "Automatic Grid Generation for 3D Device Simulation", In Proc. of SISDEP 3 Conf., pp. 497-505,1988

[3] T. Chen, D. W. Yergeau, R. W. Dutton, "Efficient 3D Mesh Adaption in Diffusion Simulation", Proc. SISPAD'96, pp. 171-172, 1996

[4] P. Fleischmann, R. Sabelka, A. Stach, R. Strasser, and S. Selberherr, "Grid Generation for Three-Dimensional Process and Device Simulation", Proc. SISPAD'96, pp. 161-166, 1996

[5] S. Cea, M. Law, "Multidimensional Nonlinear Viscoelastic Oxidation Modeling", in Simulation of Semiconductor Devices and Processes, Vol. 6 (Proc. SISDEP 95), eds. H. Ryssel, P. Pichler, Springer Verlag, 1995

[6] J. A. Sethian, D. Adalsteinson, "An Overview of Level Set Methods for Etching, Deposition, and Lithography Development, IEEE Trans. Manufacturing, 1996

[7] Dan Yang, Kincho Law, Robert Dutton, "An Automated Mesh Refinement Scheme Based on the Level-Control Function", Proc. NUPAD'92, pp 181-186

[8] For information see, e.g., http://www.tmai.com/taurus