A Common Mesh Implementation for Both Static and Moving Boundary Process Simulations

Tao Chen, Daniel W. Yergeau, Robert W. Dutton

CISX 305
Integrated Circuits Laboratory
Dept. of Electrical Engineering
Stanford University
Stanford, CA 94305

Abstract

This paper illustrates a common mesh implementation for use in both static and moving boundary process simulations. By using a single mesh server to support the different requirements of those two types of process simulations, it eliminates many interfaces between different simulators and simplifies the simulation process flow. Each simulation module only needs to communicate directly with the mesh program through a well defined common procedure interfaces. By providing a persistent and consistent storage of mesh and field data, the mesh server also greatly reduces the possibility of data loss when transported between simulation steps.

1. Introduction

Various tree based mesh generation algorithms have been used for process and device simulations on fixed geometries for some time [3], [4]. By subdividing the octants/quadrants, they provide flexible implementation for mesh refinement and derefinement, with the added benefit of minimizing interpolation errors since meshes share many common nodes.

Process steps such as etching, deposition, and oxidation involve moving boundaries and require dynamic geometry handling capabilities that are tightly coupled with mesh generation. Rule-based approaches that move individual mesh nodes become very difficult to implement, especially in 3D [5]. Recent work [1], [2], [6] has demonstrated promise using the levelset method, representing the geometry by a field defined on an Eulerian reference grid and simulating the motion of the boundary via the evolution of this field.

Since semiconductor processes involve many different steps, it's advantageous to have a unified mesh/field representation that provides functionalities for both static geometry and moving boundary problems. Field information (e.g. doping profiles) can be lost between different simulation steps when they are transported from one simulator to another. For example, a typical etching simulator only requires and outputs geometry information, thus discarding other information which may be needed by later simulation steps. Writing application specific translators to interpolate and map such field information is usually uneconomic. A unified capability is demonstrated using an octree/quadtree mesh server that provides functionalities for both types of process simulations. It also supports persistent and consistent storage of field information so that various field data can be supplied to different simulation modules upon request.

2. Algorithm

To provide the unified capability, the mesh server implemented can generate both Eulerian reference grids and Lagrangian volume meshes. During the initialization stage, an octree grid is generated by the mesh server. The data structure is very similar to a tree structure and such a grid can act as a Eulerian reference grid. The grid density in different regions is automatically adapted to user controlled specifications. Such a grid can be used directly by a moving boundary process simulations that performs topographic simulations using the levelset method. Each octant vertex has a set of levelset function values, corresponding to the distances from the nearest surfaces of geometry volumes. Each mesh node (not just octant vertices) has an array of fields to store field information. During a moving boundary step, the mesh server receives the velocities calculated by a simulator and updates the levelset function values, thus moving the boundary implicitly.

When a Lagrangian volume mesh is needed, tetrahedralization is performed in each leaf octant to generate the mesh. Several postprocessing routines including local Delaunay transformations and vertices moving can also be performed in addition to the normal tetrahedralizations. Detailed tetrahedralization algorithms are used to match geometry boundary and ensure mesh conformity. When the boundary is moved after some steps of topographical simulations, by calculating where the implicit boundary crosses the octants, new tetrahedralization can then be performed on the octants, generating a new volume mesh. The previous volume mesh is also maintained so that if any new mesh nodes are created, the field values can be interpolated using the older mesh (if applicable).

The mesh/field server and different simulation modules communicate using a common procedure interface as illustrated in Fig. 1. Such an organization allows plug & play of both mesh/field servers and simulators. Included in the interfaces are routines that provide the simulator to write any nodal field values in the mesh server. To provide persistent and consistent field data storage, the mesh server has built-in routines to automatically create, remove, interpolate and map field values when necessary.

3. Example

Fig. 2 illustrates the unified capability of the mesh server. Fig. 2(a) shows a volume mesh with an initial doping profile. After a set of diffusion simulations, Fig. 2(b) shows the final profile stored in the mesh server. Then an etching operation removes part of the geometry, resulting a changed geometry as shown in Fig. 2(c). Finally a new material is deposited onto the trench as shown in Fig. 2(d). During all these steps in support of the adaptations performed on both static and dynamic geometry, the mesh server keeps a consistent storage of both the field and geometry information. It can thereby supply necessary information to other tools upon request.

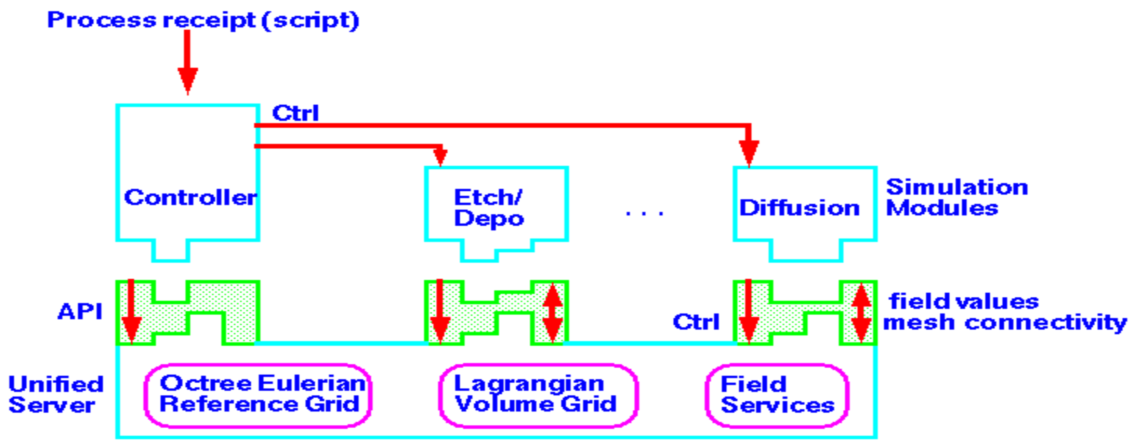

Figure 1: The organization of various simulation modules and the mesh/field server.

4. Conclusion

A common mesh implementation that can handle both static mesh generation and moving geometry is discussed in this paper. An example is shown to illustrate its use in various process simulation steps. The mesh functionalities required for both static and moving problems are provided. The implementation also enables a streamlined simulation process flow and provide easy and better support for persistent and consistent mesh and field data maintenance.

5. Acknowledgement

The authors gratefully acknowledge support from DARPA (ITO) under contract DABT63-95-C-0090.

References

[1] D. Adalsteinsson, J. Sethian, "A Fast Level Set Method for Propagating Interfaces,", *J. Comp. Phys., vol. 118, 1995.*

[2] Z. Hsiau, E. Kan, P. McVittie, R. Dutton, "Robust, Stable, and Accurate Boundary Movement for Physical Etching and Deposition Simulation", *IEEE Trans. Electron Devices, vol. 44, p.1375, September 1997.*

[3] P. Conti, M. Tomizawa, A. Yoshi, "Generation of Oriented Three-Dimensional Delaunay Grids Suitable for the Control Volume Integration method,", *Int. J. Numer. Methods. Eng., vol. 37, p.3211, 1994.*

[4] T. Chen, D. Yergeau, R. Dutton, "Efficient 3D Mesh Adaptation in Diffusion Simulation", *Proc. SISPAD 1996.*

[5] DEPICT-2, *Technology Modeling Associates, 1990.*

[6] V. Rao, T. Hughes, E. Kan, R. Dutton, "A New Numerical Formulation for Thermal Oxidation", *Proc. SISPAD 1997.*

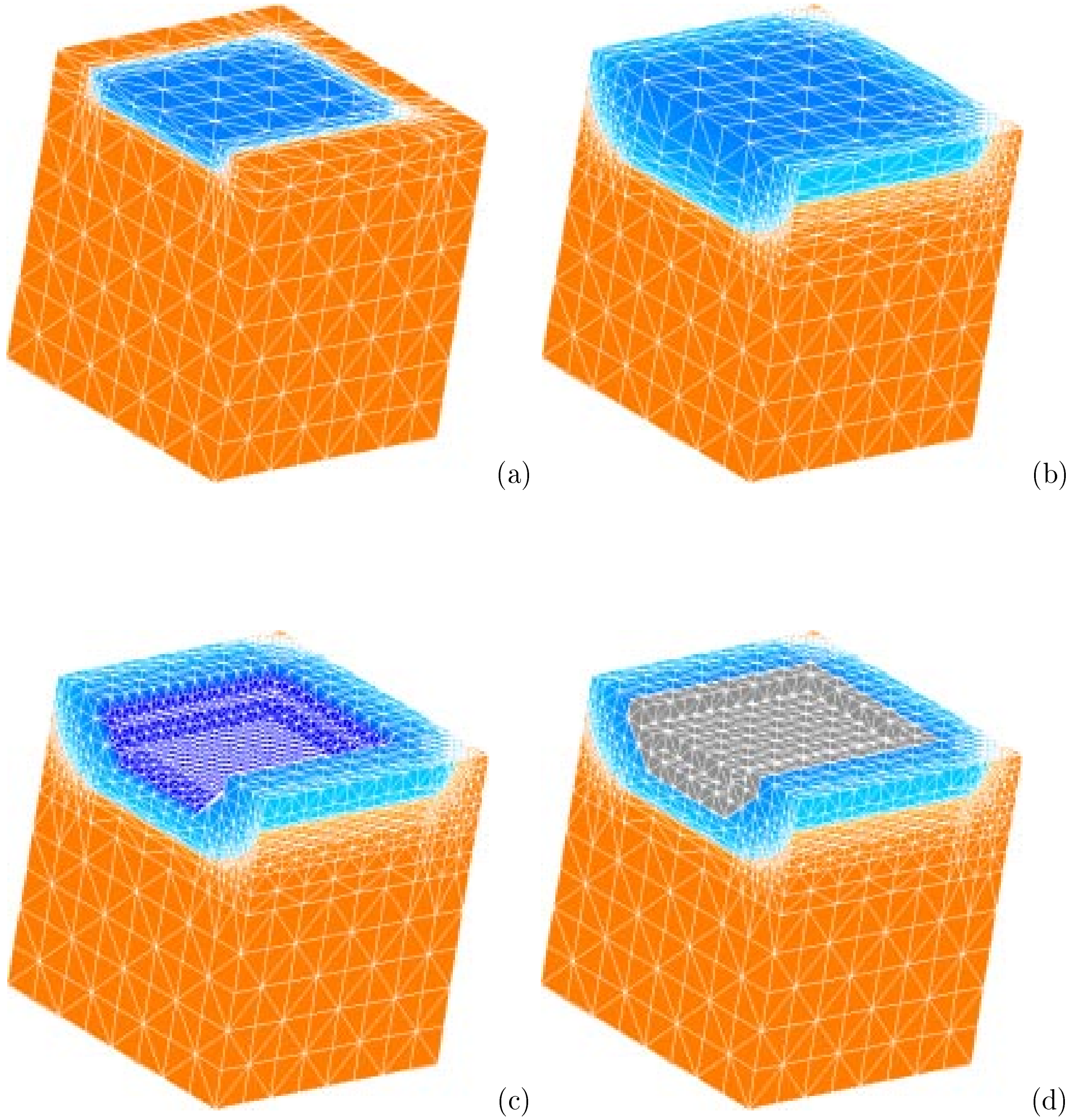

Figure 2: Using a common mesh server to handle mesh adaptation and field storage for several simulation modules. (a) initial profile. (b) after a set of diffusion simulation steps. (c) after another set of etching steps. (d) after trench deposition steps.