

# 3D Grid Generation for Semiconductor Devices Using a Fully Flexible Refinement Approach

N. Hitschfeld and W. Fichtner<sup>†</sup>

Dpto Ciencias da la Computación, Universidad de Chile  
CL-2120 Blanco Encalada Santiago, CHILE

<sup>†</sup>Integrated Systems Laboratory, ETH-Zürich  
Gloriastraße 35, CH-8092 Zürich, SWITZERLAND

## Abstract

We present a new algorithm for the generation of 3-dimensional (3-D) grids for the simulation of semiconductor devices. The fitting of the device geometry and the required mesh density is obtained by partitioning the elements at an optimal point at each refinement step. This allows the fitting of more general 3-D device geometries and the reduction of grid points in comparison with previous grid generators.

## 1. Introduction

Successful grid generators typically use iterative refinement of coarse elements to obtain a required tessellation. Iterative refinement can be implemented by either bisecting the element edges (*bisection* based approach) or dividing the element edges at an arbitrary position (*intersection* based approach). In 2-D, both approaches have been already used. Algorithms based on the *intersection* based approach generate better grids because they choose the *best* refinement point at each refinement step. To our knowledge, this *intersection* based approach still has not be used in 3-D because of its inherent complexity.

This paper presents the main aspects of a grid generator using an *intersection* based approach and compares it, as far as it has been implemented, with its ancestor  $\Omega_{me}(\Omega$  with mixed elements) [1].

## 2. Previous work

Currently,  $\Omega_{me}$  is the most powerful 3-D mesh generator for semiconductor devices to generate grids for box method discretizations.  $\Omega_{me}$  is implemented using mixed element trees, a generalization of modified octrees[2]. Instead of only bricks, 3-D elements of different shapes are used to fit the device geometry. The main algorithm looks as follows:

1. Generate a grid that fits the geometry of the modeled device exactly. This initial grid consists of bricks, rectangular prisms and rectangular pyramids. The grid is handled as a forest where each element is the root of a tree.
2. Refine due to variations in the impurity profile and certain geometrical parameters. In order to fit physical and other geometrical parameters, an irregular grid is generated by refining each element independently of the others.
3. Generate a proper Delaunay mesh. A finite element grid is obtained after tessellating all the irregular elements into tetrahedra, pyramids, prisms and bricks if the Voronoi cross sections inside of each irregular element can be properly computed.

The most serious problem of  $\Omega_{me}$  originates from the algorithm that fits the original device geometry. This algorithm generates an initial (tensor product based) grid that is a complete partition of the device (i.e. grid elements have no green points). A complete partition is required in order to be able to use a bisection based approach. During the generation of the initial grid, small geometry features are propagated (by inserting planes) to the boundaries of the device (see Fig. 1). Therefore, the initial grid contains a high number of unnecessary elements with a very bad aspect ratio. In addition, the repetitive generation of new points due to intersections between the inserted lines (planes in 3-D) and some boundary or material interfaces makes it impossible to fit several device geometries (see Fig. 1).

### 3. New Algorithm

The same consecutive steps are used to generate a grid using an *intersection*-based approach but each step is focused in a different way.

The geometry is fitted by refining the grid elements at the best possible point (see Fig. 2). The best point—the one whose associated refinement generates sons with the smallest aspect ratio—is chosen from the available green points and intersection points. Elements are bisected for generated sons with bad aspect ratio.

After the geometry is completely fitted, the irregular macro grid is further refined until the density requirements are fulfilled. Elements are partitioned in the required direction according to the best located green point.

The grid is made 1-irregular before looking for proper tessellations. Subsequently, the algorithm checks the *splittable* condition, i.e., a condition that guarantees the existence of a proper tessellation. If an element is non splittable, proper points are inserted by looking at the neighbors.

Once all elements are splittable, each local tessellation is computed using an algorithm to compute Delaunay tessellations inside of a convex-hull. We are currently exploring several different possibilities for this algorithm.

### 4. Comparison of the new algorithm with its ancestor

The algorithms to fit the device geometry and to fulfil the density requirements have been already implemented [3].

Figure 3 shows two views of the silicon part of the same ECL bipolar transistor. It can be noticed that the new algorithm strongly reduces the number of unnecessary

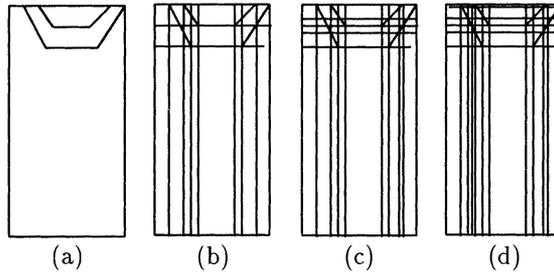


Figure 1: Fitting a 2-D device geometry using a tensor product grid. (a) Device geometry (b), (c), and (d) sequence of steps to fit the device geometry. After the step in (d) the geometry is still not optimally fitted

grid points and elements. In addition, it avoids the propagation of grid planes to the whole device and small changes in the geometry are fitted locally. Empirically, the new algorithm fits device geometries of complex devices much faster than the old one.

Because of the better fitting of the device geometry, the new algorithm also fulfils the required point density using fewer points (Fig. 4). The rest can be only empirically compared after the new algorithm to generate the final grid is implemented.

The main improvement of the new algorithm is the fitting of more realistic device geometries. In addition, its implementation shows theoretically a more predictable behavior because it does not use loops neither to fit the device geometry nor to look for proper tessellations.

## Acknowledgments

This work was partially supported by ESPRIT-6075 (DESSIS) project in 1992 and by FONDECYT project No. 1930765 in 1993.

## References

- [1] N. Hitschfeld, P. Conti, and W. Fichtner, *Mixed elements trees: A generalization of modified octrees for the generation of meshes for the simulation of complex 3-d semiconductor devices*, Accepted for publication in IEEE Trans. on CAD/ICAS, 1993.
- [2] M. A. Yerry and S. Shephard, *Automatic Three-dimensional Mesh Generation by the Modified-Octree Technique*, Int. J. Numer. Methods Eng. p. 1965–1990, vol. 20, 1984.
- [3] N. Hitschfeld, *Grid Generation for Non-Rectangular Semiconductor Devices*, PhD thesis publish. by Hartung-Gorre Konstanz, Series in Microelectronics, Vol. 21, 1993.

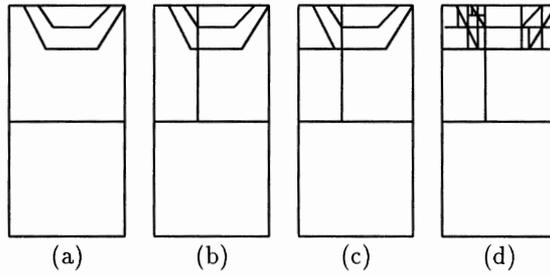


Figure 2: Fitting a 2-D device geometry using the *intersection* based approach. (a), (b), and (c) first steps to fit the device geometry (d) final initial grid.

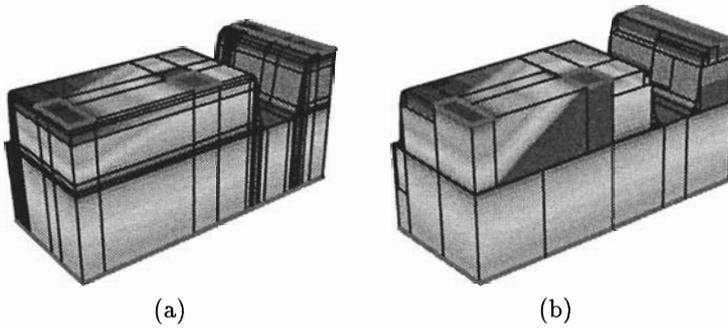


Figure 3: Fitting the device geometry for the bipolar transistor (a) tensor product grid: 2365 pts (b) *intersection* based approach: 881 pts

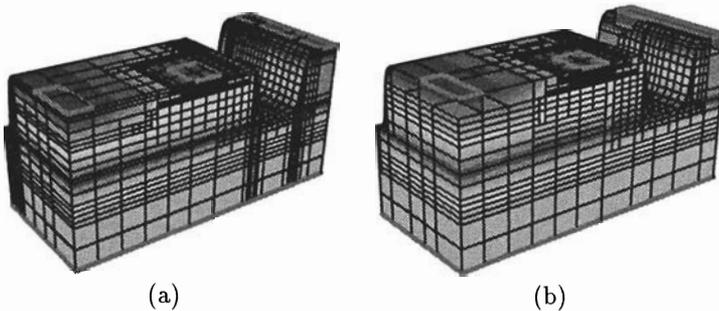


Figure 4: Achieving the desired mesh density for the bipolar transistor (a) *bisection* based approach: 10,376pts (b) *intersection* based approach: 7,097pts