(5B-1)

Generation of Meshes for the Simulation of Power Devices*

S. Müller K. Kells W. Fichtner
Integrated Systems Laboratory, ETH–Zürich, Switzerland
Phone: +41 1 256 5746, FAX: +41 1 252 0994

With the growing complexity of semiconductor devices and the use of increasingly sophisticated device simulation packages to prototype these devices, new demands have been made on the speed and efficiency of mesh generators and the sophistication of the meshes that are created.

The simulation of power devices puts more stringent requirements on the mesh generation, because of the very large sizes and the important influence of the geometry. Also, the numerical problems increase because of the possibly large gradients in the physical parameters. This puts a higher pressure on the quality of the meshes with respect to the discretization errors made. Because we use the box method [1], we have the restriction that we should not use elements with obtuse angles or bad aspect ratios [2].

In our search of the literature to find an algorithm on which to base our program (e.g., [3, 4, 5]), we found that the algorithms presented until now have either only partially fulfilled the above criteria, produce an unnecessarily high number of elements, or are very complex or based on probabilistic methods.

Our device specification consists of several regions, each described by a polygon boundary and a material type. We cover this with a rectangular mesh, refine this where necessary and use triangles to fit the boundary and to handle *green* points (extra points introduced in the adaptive refinement). In the end we allow the user to refine or unrefine the resulting mesh interactively, because the engineer will always have a better knowledge than any program about where large changes in the solution are expected and therefore, where the mesh has to be fine.

Our major improvements as compared to other approaches are: An improved tensorproduct covering, the *fuzzy-box* refinement and an implementation with data structures that allow adaptive (un-)refinement.

The adaption of meshes to an arbitrary device geometry can cause quite large difficulties, because of the complex shapes they can have. The easiest attempt to simplify the problem, is a tensorproduct covering through all boundary vertices. We improved this attempt such that the aspect ratio of the resulting elements is always good.

To be able to fit the boundary, we take care during the refinement steps that we generate only elements that we can fit simple. Therefore, we developed the method show in fig. 1.

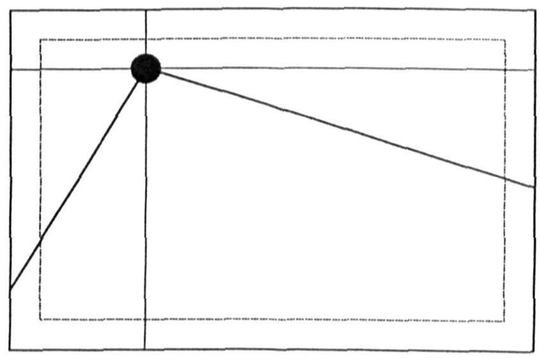

Figure 1: *Fuzzy box* refinement to a boundary vertex showing fuzzy-box for 8 : 1 maximum aspect ratio.

We define a *fuzzy box* by the criterion that the aspect ratio of the rectangles that are created during the refinement should not be larger than a user given parameter. If a critical point (e.g. a boundary vertex, a boundary intersection, or a green point) is inside the box, we can directly refine on it.

The whole program is written in C++. This object-oriented language allows us to implement the mesh in several tree structures. That gives us an easy way to refine and unrefine according to the wishes of the user.

As a representative example we show a mesh together with the doping profile for a thyristor

*Typeset on June 15, 1990.

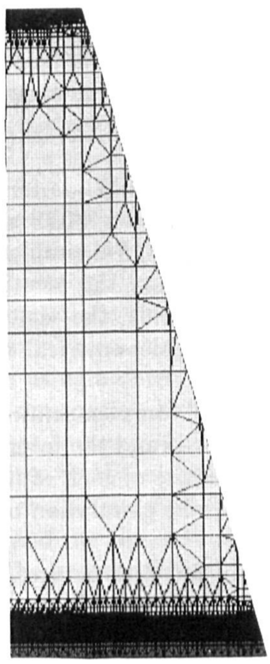

Figure 2: The whole thyristor

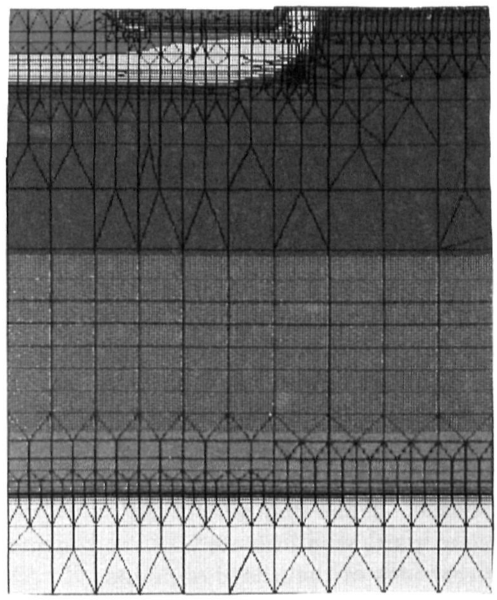

Figure 3: Enlargement of the upper 50 μm

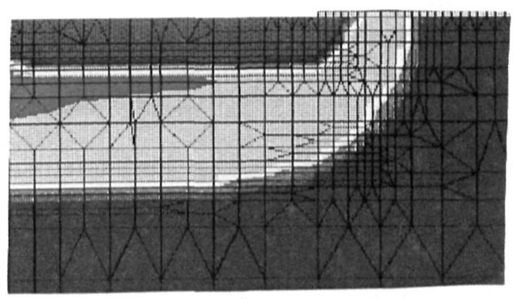

Figure 4: Enlargement of the oxide in the MOS structure

and some enlargements of it (Fig. 2, 3, 4). The height of this thyristor structure is 0.5 mm, the width is 0.2 mm. The oxide height of the MOS structure of the last figure is 0.05 μm. We use 32117 elements and 26321 vertices.

The mesh generator designed and implemented by the authors, `meshbuild`, produces meshes with no obtuse angles for even complex geometries, while at the same time reducing the overall number of mesh elements. Runtimes for complex examples are on the order of one minute on a Sun 4.

That means, to the best of our knowledge, that we are now for the first time able to model real-life power devices with typical sizes in the order of up to several millimeters.

References

[1] R. Bank, D. Rose, and W. Fichtner, "Numerical methods for semiconductor device simulation," *IEEE Transactions on Electronic Devices*, vol. ED-30, pp. 1031–1041, 1983.

[2] G. Strang and G. Fix, *An Analysis of the finite element method*, pp. 172–204. Prentice-Hall Englewood Cliffs, 1973.

[3] B. Baker, E. Grosse, and C. Rafferty, "Non obtuse triangulation of polygons," *Journal of Discrete and computational geometry*, vol. 3, pp. 147–168, 1988.

[4] P. Ciampolini, A. Forghieri, A. Pierantoni, A. Gnudi, M. Rudan, and G. Baccarani, "Adaptive mesh generation preserving the quality of the initial grid," *IEEE Transactions on CAD*, vol. 8, pp. 490–500, 1989.

[5] B. Wördenweber, "Finite-element mesh generation from geometric models," *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 2, no. 1, pp. 23–33, 1983.