SIMULATION OF SEMICONDUCTOR DEVICES AND PROCESSES Vol. 3
Edited by G. Baccarani, M. Rudan - Bologna (Italy) September 26-28, 1988 - Tecnoprint

A NOVEL APPROACH TO ADAPTIVE MESHING FOR THE SEMICONDUCTOR PROBLEM

W.H.A. Schilders

Nederlandse Philipsbedrijven BV, Corp. CAD-Centre/MSW
Eindhoven, The Netherlands

SUMMARY

In this paper we describe an adaptive meshing technique which is specifically aimed at accurately solving semiconductor device problems for a sequence of bias conditions. The crucial step is the iterative generation of an initial grid, which is used as the basis for further refinement. Another feature of the method is that it performs additional refinement according to a pre-described quantity.

INTRODUCTION

In the past decade, many programme packages for the analysis of arbitrary semiconductor devices have been developed (Baccarani et al., 1985; Cottrell and Buturla, 1979; Pinto et al., 1979; Polak et al., 1987; Selberherr et al., 1980). Most of these packages require the user to specify a mesh on which the problem is to be solved. For an accurate solution, this approach requires the user to carefully place mesh elements. In fact, this means that there has to be a priori knowledge on the behaviour of the solution. Furthermore, when a sequence of problems is solved, the specified mesh has to take the changing behaviour of the solution into account. Often this leads to an excessive number of mesh elements.
In order to avoid the problems described above, interactive mesh editors have been designed which allow the user to adapt the mesh to characteristics of the solution (see, for example, Armstrong et al., 1985). In practice, however, these mesh editors are only used as a means of providing automatic mesh generation to model a specific doping profile. An alternative approach is to allow the user to change the mesh during the solution process. In this case, the mesh is refined according to the behaviour of the solution calculated so far (cf. Pinto et al., 1984). Both approaches suffer from the disadvantage that the mesh is not based on the solution being calculated. In addition, most algorithms do not allow removal of

unneccessary mesh elements.

Recently, we have developed a method which specifically aims at providing adaptively generated meshes for an arbitrary semiconductor device problem which has to be solved for a sequence of bias conditions. For each bias condition a new starting grid, which already reflects the behaviour of the new solution, is determined iteratively. Thus, the new mesh will in general not be a refinement of the old mesh. Another feature of the method is that, in addition to the accurate solution of the problem, one or more quantities can be specified which will also be resolved accurately by performing some additional refinement. Examples of such quantities are depletion capacitances and recombination currents.

The algorithm for the method described above consists of several phases. During the first phase, a coarse rectangular initial grid is generated iteratively. The resulting grid does already reflect the behaviour of the solution to be calculated, and will in general not be a refinement of the mesh used for the previously calculated solution. Furthermore, its determination is relatively cheap since only a few iterations are needed for convergence. In the second phase, the initial grid is refined in order to obtain an accurate solution. The discontinuities in the electric displacement and the current densities are used as criteria for refinement. At the moment, we only refine the mesh by adding mesh lines. In the future, local refinement will also be incorporated. In the last phase, we allow the possibility of a more accurate determination of specified quantities. The reasoning behind this is that these quantities may vary rapidly over small regions in the device, whereas the criteria used for accurately determining the potentials and carrier concentrations do not give rise to sufficient refinement in these regions. Therefore, during this phase, some additional refinement will take place.

In the following sections, we will describe the method in more detail, and discuss several aspects, such as convergence of the iterative process for the initial grid, interpolation of the solution from one grid to another, and the use of stopping criteria in order to guarantee a reasonably efficient solution. Finally, we present several applications of the method to practical problems.

ITERATIVE DETERMINATION OF AN INITIAL GRID

The first stage of the new algorithm consists of adaptively constructing a suitable initial grid for a given bias condition. This grid should be coarse, but at the same time it should contain enough mesh elements to already reflect the behaviour of the solution (which is still to be calculated). A very simple way of achieving this is to iteratively construct a rectangular N_1 x N_2 grid (or, in 3-d, an N_1 x N_2 x N_3 grid), where N_1 and N_2 are small numbers (usually in the range from 10 to 30). The method for generating such grids is essentially one-dimensional in nature, and will be described in the

following (it was suggested by Hemker, 1987).
Let N > 0 be a given number. Assume we have a grid $\{x_1,\dots,x_N\}$ in 1-d and corresponding solution values $\{u_1,\dots,u_N\}$ (here, "solution" refers to the solution of a discretised one-dimensional problem). Let f be a quantity depending on the solution, and $\{f_1,\dots,f_N\}$ the values of f at the nodes. In the example problems, f_i is the jump of the electric displacement at the node x_i. The function f will be termed the criteria function.
The approach is now to subdivide (into N-1 equal areas) the area under the piecewise linear function with nodal values the f_i, i=1,..,N. The total area under the curve is equal to (for simplicity we assume the f_i to be non-negative):

$$A = \sum_{i=1}^{N-1} (x_{i+1} - x_i)(f_{i+1} + f_i)/2$$

Thus, we have to find nodes $\hat{x}_1,\dots,\hat{x}_N$ such that the area under the piecewise linear function between nodes $\hat{x}_k$ and $\hat{x}_{k+1}$ is equal to A/(N-1). To do this, we proceed as follows. First we define $A_1,\dots,A_N$ by:

$$A_1 = 0$$

$$A_i = A_{i-1} + (x_{i+1} - x_i)(f_{i+1} + f_i)/2 \quad , \quad i = 2,\dots,N$$

Remark that $A_N = A$, and that A_i is the area under the piecewise linear function from x_1 to x_i. Now we find the new nodes. Of course, $\hat{x}_1 = x_1$. Now suppose we have already found $\hat{x}_1,\dots,\hat{x}_i$, and we want to determine $\hat{x}_{i+1}$. We know that

$$A_k \le \frac{i-1}{N-1} A < A_{k+1}$$

for a certain k. Now there can be two cases:

1) $\dfrac{i}{N-1} A < A_{k+1}$

2) $\dfrac{i}{N-1} A \ge A_{k+1}$

In the first case, we have that the new node $\hat{x}_{i+1}$ will be in the interval $[x_k, x_{k+1}]$ and has to satisfy:

$$(1) \qquad \frac{A}{M-1} = \frac{1}{2} (\hat{x}_{i+1} - \hat{x}_i) [u(\hat{x}_{i+1}) + u(\hat{x}_i)]$$

where f is the piecewise linear polynomial with nodal values $f_1, \ldots, f_N$. This is a quadratic equation for $\hat{x}_{i+1}$, which can be readily solved.

In the second case we know that $\hat{x}_{i+1}$ will be situated to the right of x_{k+1}. Thus we replace k by k+1 and repeat the process until we have found a new value for k such that we are in case 1). Then we proceed as described above, replacing the left hand side of equation (1) by

$$\frac{A}{M-1} - A_k$$

which is the remaining area, and replacing $\hat{x}_i$ by x_k in the right hand side.

The above process is repeated until we have found $\hat{x}_1, \ldots, \hat{x}_N$. Of course, we must have that $\hat{x}_N = x_N$. Thus we have constructed a new mesh.

The construction in one dimension can be easily generalised to two or three dimensions. In two dimensions, we perform the construction twice, once for the x-direction and once for the y-direction. For example, when designing a new mesh in the x-direction, the value f_i is taken to be the maximum of the nodal values $f_{i,j}$ over all j. In this way, a new rectangular grid with the required number of mesh lines is constructed.

In the above we have described the procedure for generating rectangular grids with a given number of mesh lines. If a new grid has been constructed, we can solve the corresponding discrete problem to obtain the solution on this grid. Thus, we have a new set of nodal values and using these we can again generate a new grid. In this way, an iterative process can be set up to find an optimal rectangular grid with the required number of mesh lines. It should be remarked that, in practice, it is sometimes required that certain lines (for example those defining the geometry of the device) are always mesh lines. Such lines are termed coarse mesh lines. This can be treated in a very simple way by subtracting the number of coarse mesh lines in one direction from the required total number of mesh lines in that direction. Then we determine a mesh distribution with the smaller number of nodes, and add the coarse mesh points afterwards. In this way we again obtain the required number of nodes for that direction.

The algorithm described above has been designed to be easily implemented in existing codes. In fact, the original solution process only has to be interrupted to change the mesh and related data. Once a new mesh has been constructed, the solution process can be continued. The only problem here is to transfer the data from the old to the new mesh. For geometrical data, this is straightforward, for solution-dependent data some kind of interpolation is needed. Although sophisticated interpolation (e.g. making use of the exponential behaviour of the carrier densities) is a possibility, we suggest to use straightforward linear (bi-linear, tri-linear) interpolation, in combination with a simple damping procedure. The latter is important, since it is wellknown that the subsequent Newton process may converge either very slowly or not at all (cf. Polak et al., 1987). Finally, it is clear that the Newton processes for the intermediate grids should only be allowed to do a few iterations. Only when we have reached convergence of the grids, the Newton process should be performed to the desired tolerances.

From a mathematical point of view, the method described above is related to the moving finite element method (Miller and Miller, 1981), which has also been used (in a modified form) for process simulation (Baines et al., 1986). The difference is that we do not set up and solve equations for the coordinates. Although this approach gives a better convergence behaviour of the grids (cf. next section), it is much more time consuming and will thus drastically reduce the practical value of the method. Furthermore, we use the generated grid as an initial grid for further refinement, and not as the final grid for the specified bias. Finally, the moving finite element method does not seem to be easily implementable into existing codes.

CONVERGENCE OF THE GRIDS

An important matter concerning the iterative procedure described in the previous section is its convergence. It can be shown that, for smooth criteria functions f, the grids converge to a final grid with the property that the solution values at the nodes are such that the area under the piecewise polynomial is equi-distributed between the nodes. Furthermore, this final grid is independent of the starting grid. From these two observations it follows that the method is robust. However, for the method to be of practical value (especially for 3-d problems), it is important that the number of iterations is minimal. We have investigated the performance of the method on several 1-d and 2-d problems, showing that the number of iterations is rather low (typically 4 to 8). In the following we will present two of these examples and discuss how convergence of the grids can be measured.

The first example is an essentially 1-dimensional bipolar transistor with an artificial base in the interior of the device. A sequence of grids was generated, starting from a uniform grid, with 65 mesh points. It should be remarked that quite a large number of nodes were taken in order to be able to better investigate the convergence behaviour. The convergence can be measured in a simple way by calculating the maximum shift of the mesh points, i.e.:

$$s = \max_{i} |x_i - \hat{x}_i|$$

In Table 1 we give the results for this problem:

Table 1. Convergence of grids for 1-d transistor

	s (in µm)
grid 0 → grid 1	3.030
grid 1 → grid 2	0.400
grid 2 → grid 3	0.160
grid 3 → grid 4	0.065

(the total width of the device is 8 µm). This way of measuring can also be performed in two or three dimensions.
The second example is a 2-d bipolar transistor, for which we perform the iterative process constructing 25 x 20 grids. The applied bias was taken to be 1 Volt at the collector, 0 Volt at the base and -0.7 Volt at the emitter. In Figure 1 we display some of the grids that were generated for this bias condition. The starting grid is minimal in the sense that it contains just enough mesh lines to specify the geometry of the device. Despite this coarse starting grid, the process converges quite rapidly.

ADAPTIVE REFINEMENT OF THE INITIAL GRID

In the previous sections we have described how, for a new bias condition, an initial grid can be generated which is coarse but at the same time reflects to some extent the behaviour of the solution to be calculated. Therefore this initial grid provides a useful starting grid for a refinement procedure.
The latter can be performed in many ways. Thus far, we have only experimented with a refinement procedure which adds only entire mesh lines, but in the near future we will be repeating the experiments with local refinement.
An important matter in the adaptive refinement process is the criterion used. For semiconductor device modelling using a box discretisation method, a criterion based on the jump in the electric displacement and the current densities between neighbouring mesh elements has proven to perform well. Therefore, we also use this as the criterion in our refinement procedure. However, we do not consider the final grid of this

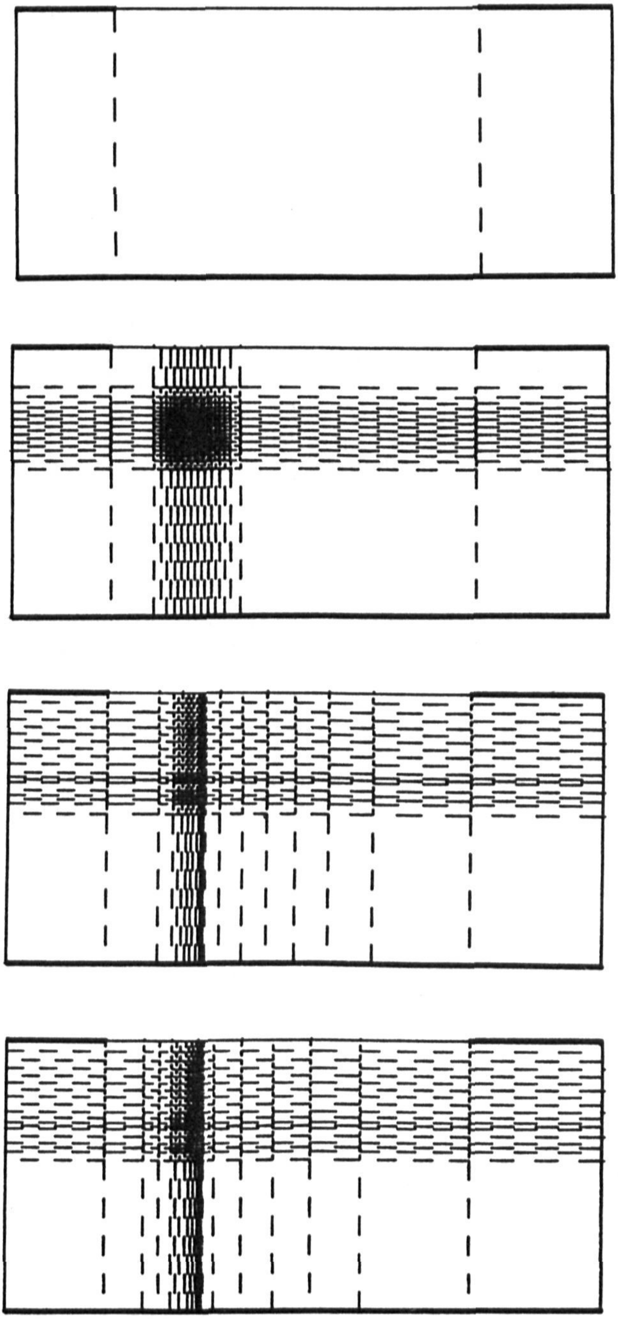

Fig. 1. Sequence of grids for
2-d bipolar transistor

refinement process as the final grid of the solution for the given bias condition. Instead, we allow for yet another adaptive procedure. The reason for this is that, in practice, one is often interested in the behaviour of certain so-called secondary quantities. Examples of these are recombination currents and depletion capacitances. Often these quantities vary rapidly in small regions within the device, and are not represented accurately enough on the grid found so far. Thus, it seems reasonable to perform some extra refinement in areas where this is necessary for the given quantity to be determined adequately.

Summarizing, we thus have a refinement procedure which starts from the iteratively generated initial grid and consists of two stages. The first stage consists of the usual approach of refining according to the jump in electric displacement and the current densities. In the second stage we refine the mesh where a pre-described secondary quantity is not yet represented accurately enough. In the lecture, we will present an example of the full procedure applied to a practical problem.

CONCLUDING REMARKS

We have presented an adaptive refinement procedure which is aimed specifically at solving semiconductor device problems for a sequence of bias conditions. The method is easily implemented in existing codes and can be applied to 1-d, 2-d and 3-d problems. It is attractive from a computational point of view since the initial grids for a given bias condition will (in general) not be refinements of the final grid for the previous bias condition. For semiconductor problems in 3 dimensions this is a very important feature.

We have also described how the commonly used refinement procedure, based on inter-element jumps in the electric displacement and the current densities, can be followed by a refinement process which produces accurate representations of certain pre-described quantities. This is rather important when simulations are used for investigations into the behaviour of such quantities.

REFERENCES

G.A. Armstrong, R.S. Ferguson and J.G. Flynn (1985). An Interactive Finite Element Mesh Editor. In J.J.H. Miller (Ed.), Proceedings NASECODE IV Conference, Boole Press, Dublin, pp. 152-157.

G. Baccarani, R. Guerrieri, P. Ciampolini and M. Rudan (1985). HFIELDS: a Highly Flexible 2-D Semiconductor-Device Analysis Program. In J.J.H. Miller (Ed.), Proceedings NASECODE IV Conference, Boole Press, Dublin, pp. 3-10.

M.J. Baines, C.P. Please and P.K. Sweby (1986). Numerical Solution of Dopant Diffusion Equations. In K. Board and D.R.J. Owen (Eds.), Proceedings SISDEP-2 Conference, Pineridge Press, Swansea, U.K., pp. 271-286.

P.E. Cottrell and E.M. Buturla (1979). Two-Dimensional Static and Transient Simulation of Mobile Carrier Transport in a Semiconductor. In B.T. Browne and J.J.H. Miller (Eds.), Proceedings NASECODE I Conference, Boole Press, Dublin, pp. 31-64.

P.W. Hemker, 1987. Private communication.

K. Miller and R.N. Miller (1981). Moving Finite Elements, Parts I and II. SIAM Journal on Numerical Analysis, Vol. 18, pp. 1019ff and pp. 1033ff.

M.R. Pinto, C.S. Rafferty and R.W. Dutton (1984). PISCES-II User's Manual. Stanford Electronics Lab., Dept. of Electrical Engineering, Stanford University.

S.J. Polak, C. den Heijer, W.H.A. Schilders and P.A. Markowich (1987). Semiconductor Device Modelling from the Numerical L Point of View. International Journal for Numerical Methods in Engineering, Vol. 24, no. 4, pp. 763-838.

S. Selberherr, A. Schuetz and H.W. Poetzl (1980). MINIMOS - a Two-Dimensional MOS Transistor Analyzer. IEEE Transactions on Electron Devices, Vol. ED-27, pp. 1540-1550.