Doctoral Thesis

A hybrid finite element method to solve the stationary semiconductor equations including galvanometric effects

Author(s):
Sartoris, Guido Ettore

Publication Date:
1993

Permanent Link:

https://doi.org/10.3929/ethz-a-000904770 →

DISS. ETH Nr. 10216

A HYBRID FINITE ELEMENT METHOD TO SOLVE THE STATIONARY SEMICONDUCTOR EQUATIONS INCLUDING GALVANOMETRIC EFFECTS

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH
for the degree of
Doctor of Technical Sciences

Presented by
Guido Ettore Sartoris
Dipl.Phys.(ETH)
born on 10 January 1963
Citizen of Switzerland

Accepted on the recommendation of
Prof. Dr. E. Anderheggen, Examinator
Prof. Dr. H. Melchior, Co-Examiner
Prof. Dr. G. Gonnet, Co-Examiner

Zürich, 1993

Zusammenfassung

Nach der Erfindung des bipolaren Transistors durch Bardeen und Brattain im Jahre 1947 und der Entwicklung des MOS-Transistors in den Sechzigerjahren, stieg der Bedarf an numerischen Modellen zur Simulation des physikalischen Verhaltens von Halbleiterkomponenten. Heute, nach mehr als dreissig Jahren weltweiter, intensiver Bemühungen um aussagekräftige Computermodelle, wird in diesem Forschungsgebiet immer noch aktiv geforscht. Die Modellierungsaufgaben sind komplex. Eine Vielfalt von Methoden sind bis heute vorgestellt worden, jede mit gewissen Vor- und Nachteilen, alle mit dem primären Ziel, eine genaue und zuverlässige Modellierung zu ermöglichen. Zweck der Modellierung ist dabei die Charakterisierung und Optimierung des elektrischen Verhaltens von elektronischen Komponenten, was letztlich im Vergleich zur rein experimentellen Entwicklung, zu kleineren Entwicklungskosten führt. Daneben sei auch auf die Bedeutung von numerischen Modellen für die Ausbildung hingewiesen. Unter vielen physikalischen Modellen ist das "Drift-Diffusion"-Modell eines der einfachsten und gleichzeitig von grosser Bedeutung. Es geht auf van Roosbroeck zurück und wurde in den Fünfzigerjahren vorgestellt. Dieses Modell besteht aus einem nichtlinearen System von drei partiellen Differentialgleichungen zweiter Ordnung, dessen Lösung analytisch und numerisch problematisch ist: Analytische Lösungen existieren nur in den aller einfachsten Fällen. Gängige Diskretisierungs-methoden für numerische Lösungen sind ungeeignet, ausserdem ist der anfallende Rechenaufwand zur Erzeugung von, betreffend Genauigkeit akzeptablen Lösungen, enorm gross. Verschiedene Diskretisierungs-methoden mit kleinem Rechenaufwand für adequate Resultate sind bis heute vorgeschlagen worden. Der kontinuierliche Miniaturisierungs-prozess in der Halbleiterindustrie lässt den Bedarf an dreidimensionalen Simulationen ansteigen. Damit steigt zusätzlich die Nachfrage nach wenig rechenaufwendigen Diskretisierungsformeln, denn die drei-dimensionalen numerischen Simulationen von Halbleiterbauelementen eine enorm grosse Rechenleistung verlangen, die die Limiten der heutigen Superrechner übersteigen können.

Die vorliegende Dissertation befasst sich mit einer neuen finiten Elementmethode zur Diskretisierung des 2D und 3D "Drift-Diffusion"-Modells, sowie mit den Fragen der Entwicklung eines entsprechenden Computerprogramms. Ausgehend von der Boltzmann Gleichung wird zuerst das "Drift-Diffusion"-Modell hergeleitet, dies unter Berücksichtigung von galvanometrischen Effekten. Die hybride finite Elemente Methode zur Lösung von partiellen Differentialgleichungen zweiter Ordnung wird vorgestellt und für den Fall des "Drift-Diffusion"-Modells angewendet. Andere zur Simulation notwendige Algorithmen werden bezüglich Rechenaufwand und Speicherbedarf diskutiert. Die Programm- und Datenstruktur werden vorgestellt. Numerische Resultate werden am Schluss präsentiert.

Conclusions

The first part of the development of SESES was characterized by software engineering aspects. At the same time, we developed a primary SESES kernel and the graphics program, both requiring a judicious choice of data structures and algorithms to avoid possible rewriting of code in a later phase. When the kernel was able to treat uniform 2D and 3D meshes, we attached a linear solver and programmed the simple Laplace equation for brick elements. This simple linear problem already exhibited all difficulties related to 3D modeling. We were confronted with performance problems when evaluating the element matrices so that an accurate analysis was required in order to optimize the code. Also the choice of the current field shape functions was not so obvious at the beginning, which somewhat delayed our time-table. At this stage, the kernel was able to treat in a non-optimized way a refined mesh and the handling of constrained node values, which allowed us to get the first convincing results in 3D for the Laplace equation. The development continued, the graphics interface became more sophisticated, the kernel assumed its definitive form and non-brick elements were implemented together with the *Reduced* model. At this stage the software engineering work was terminated, all interfaces were clearly defined and we started with the development of the *Full* model. We knew already at the beginning that this would require the greatest effort and *surprises* of any kind were expected. The uncoupled algorithm was first implemented with Gauss quadrature to perform the integration. But what a disappointing result, even a simple abrupt 1D diode required so many elements to guarantee a good convergence. In trying to speed-up the computations we thus implemented the polynomial and rational extrapolation of the start solutions up to the fifth order with the result that in many cases of interest the simplest extrapolation, i.e. linear extrapolation, is the best choice for maximal speed-up. For forward biased diodes the uncoupled algorithm converged slowly, so that we also implemented different versions of the Δ^2-Aitken acceleration algorithm, but without getting satisfying results. Later by choosing a better approximation for the

charge we notably reduced the constraint on the mesh size. At this stage it was clear that Gauss quadrature was unable to reproduce the exact solution available in 1D of the carrier continuity equation for a vanishing recombination and we implemented the *ExpLin-Lagrange* integration. Some changes were now required in the program because the new integration algorithm able to integrate correctly all 1D integrals was no longer based on evaluating the functions on some sample points, but worked with monomial coefficients. Many 1D examples now worked pretty well, even with very coarse meshes. For 2D and 3D the integration algorithm needed further improvement, because even with relatively fine meshes negative values for the carrier densities were obtained. We always hoped to have for 2D and 3D an exact integration algorithm, but the complexity and time requirements for its implementation seemed to be far beyond our possibilities. Therefore, we limited ourselves to simply computing better approximations, which automatically involve more stringent constraints on the mesh size than with an exact integration algorithm. We completed the implementation of the coupled algorithm but something was wrong with the Carrier variables: they simply did not work. This was an odd situation because the carrier continuity equations in the Slotboom and Carrier variables are both in 1D and for a vanishing recombination completely equivalent. Both are linear and use only the carrier field values at the element boundaries which in 1D reduce to the node values only, so that they must exhibit exactly the same convergence behaviour. The reason for the discrepancy was quickly discovered, and in fact when we forced the 1D solutions computed on a 2D mesh to be symmetric towards the symmetry axis, i.e. after each iteration we reset the small non-symmetries of the order of the machine precision, then both variables were completely equivalent. The problem in using the Carrier variables was the discontinuity of the argument when evaluating the matrix G and vector $\vec{E}$ and the high sensitivity of the algorithm using the classical field approximation for the carrier densities. This problem has been solved by choosing the same field approximation used by the Slotboom variables which also makes the behaviour of both variable sets for the uncoupled algorithm similar.

It has not been long since we returned to the question of evaluating the integrals for 2D and 3D exactly. For the 2D case after a new analysis we derived a feasible algorithm. We directly started the implementation by focusing our attention on the execution speed. After tuning of the algorithm we obtained very good results. The 2D program version was now a fast and robust tool to model 2D semiconductor devices.

SESES can refine and unrefine the mesh very fast. To the aim of speedup the solution process is therefore possible to first compute a solution on a coarse meshes, refine the mesh locally on regions where the numerical error is large, interpolate the solution onto new mesh and compute a more precise solution. This process can be repeated recursively. Towards this topic, however, some more efforts most be done. First, the interpolated solution is generally a bad solution, many iterations are requested to compute a solution for the new mesh so that a better interpolation scheme is requested. Secondly we would like to have a correct error estimator to perform local refinement, actually we use *ad hoc* methods.

Almost all effort of this work is done on the numerical side, how to discretize the *Full* model equations assuring convergence of the solution even with the use of a coarse mesh. Little time has been spent in describing physical models and only the simplest and most commonly used models are actually implemented. However, the time required to include more sophisticated models is negligible and can be done at any time in order to fulfil specific user requirements. The code has been so organized that this would be a simple task and if the user has some knowledge of the programming language *C* he can do this work by himself (or herself) so that we do not have to worry about implementing the most general models.