

# ANALYTICAL PARTIAL DIFFERENTIAL EQUATION MODELING USING AMIGOS

M. RADI, E. LEITNER, E. HOLLENSTEINER and S. SELBERHERR

Institute for Microelectronics, TU-Vienna  
Gusshausstrasse 27-29, A-1040 Vienna, AUSTRIA  
Phone: +43-1-58801/5234, FAX: +43-1-505922  
e-mail: radi@iue.tuwien.ac.at

## Abstract

To accurately simulate modern semiconductor process steps, a simulation tool must include a variety of physical models and numerical methods. Increasingly complex physical formulations are required to account for effects that were not important in previous generation of technology. Thus flexibility in definition of models as well as numerical solving methods is highly desirable. An object-oriented approach has been applied to implementing a dimension independent solver which uses an analytical input interface in the manner of MathCad, Mathematica, Matlab, etc. but highly optimized for high performance semiconductor modeling.

## 1 INTRODUCTION

During the last years it became increasingly difficult to meet all requirements in semiconductor simulation since a lot of new more complex physical models were developed. The transposition to computer science turned out to be very time consuming and resulted in a profusion of different simulators which can handle more or less a special but very restricted field of physical behavior especially in three dimensions. At the same time the demand for single process simulation decreased rapidly and instead of simple singular process steps the industry requires more and more a complete simulation flow control which can handle all necessary simulation steps to describe the behavior of any semiconductor device and its manufacturing process. However, the development of new complex physical models and its practical realization requires a large expenditure of time. To reduce this amount of developing time we analyzed the requirements for a general solver carefully and separated the common features such as modeling for physical definitions, parameter and model library, grid-adaptation, numerical solver, simple front-end controller as well as geometry and boundary definition. As a result we developed AMIGOS (Analytical Model Interface & General Object-Oriented Solver) that translates a mathemat-

ical formulation of any nonlinear discretized coupled partial differential equation system into a highly optimized model which will then be passed to a nonlinear numerical solver that can handle different physical models on various grids as well as interfaces and boundaries.

## 2 AMIGOS (ANALYTICAL MODEL INTERFACE & GENERAL OBJECT- ORIENTED SOLVER)

AMIGOS (Figure 1) is a problem independent simulation system which can handle any nonlinear partial differential equation system in time and space in either one, two or three dimension(s). It is designed to automatically generate optimized numerical models from a simple mathematical input language, so that no significant speed loss in comparison to 'hand coded' standard simulation tools occurs. In difference to similar algorithms working with the so called 'operator on demand' concept [1], AMIGOS is completely independent of the kind of discretization since the model developer can formulate any discretization of choice. There are no restrictions whether using scalar, field or even tensor quantities within a model, and, if desired, any derived field quantity can be calculated, too. Furthermore, the user can influence the numerical behavior of the differential equation system by complete control of the residual vector and its derivative (e.g. punishing terms, damping terms, etc.). Even interpolation and grid-adaptation formulations can be used within a developed model and can thus be very well fitted to the special problem.

AMIGOS also provides several layers of access to the variety of users. For example, a process engineer (user-mode) may need only to select a model appropriate for the process step to calculate (e.g. field dependent diffusion), modify the process parameters (e.g. duration, temperature, material characteristics, etc.)

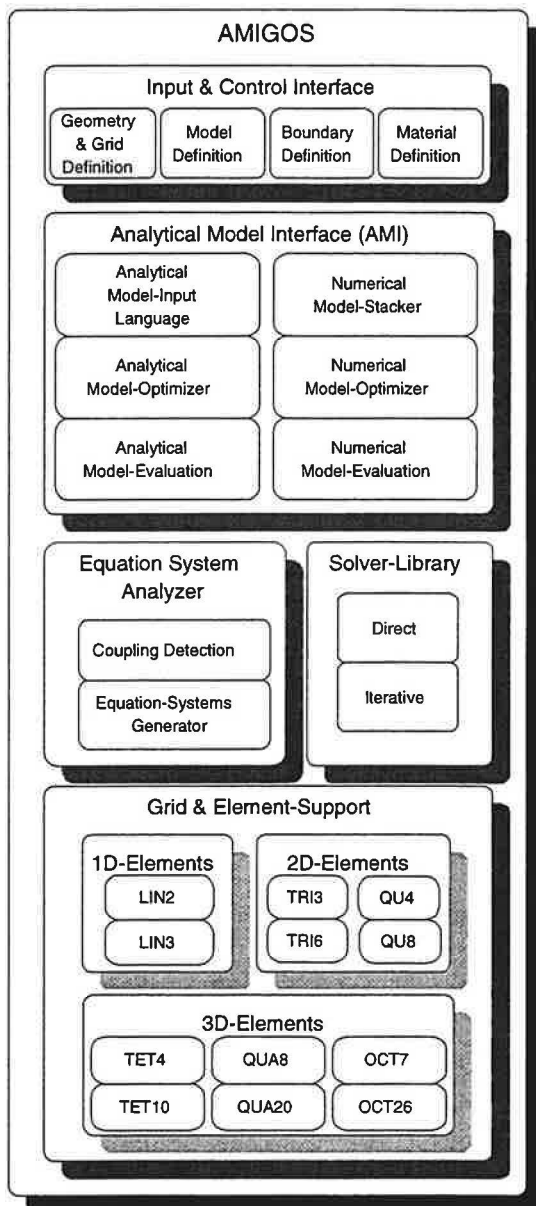


Figure 1: BLOCK STRUCTURE OF AMIGOS

and defines the geometries and boundaries the process should be calculated on. On the other hand, at a lower level (developer-mode) a model developer may need to modify existing equations by adding parameters, mathematical terms or equations, or even develop a complete new model. In contrast with previous generations of software none of the described modes requires access to and modification of the source code. Even during model development the analytical user input will be interpreted, optimized, transformed and solved on any complex simulation domain at once without the necessity of time consuming recompilations (one-pass concept) supporting a variety of several testing and debugging features. After finishing the test and calibration phase the user can switch to the two-pass concept

where all modifications are translated to C-code and are linked to a model library for high performance calculations on large simulation domains in the standard user-mode.

## 2.1 THE INPUT & CONTROL INTERFACE

The Input & Control Interface is mainly for describing the complete simulation flow process. The user has to define which physical models and boundary conditions should be used, where to read the geometry from and which grid to use for discretization. Furthermore the kind of interpolation as well as the numerical solving methods are chosen within the Input & Control Interface. This information represents the basic input needed to solve a differential equation with AMIGOS. All of the remaining tasks are done automatically by the simulation system without any further interactions by the user. It detects whether to use moving grids in case of e.g. mechanical deformations as well as the dimension of the problem. It prepares all boundaries and interface grids and checks for existing interconnections between several areas. It initializes all models, its quantities and derivatives and allocates all necessary memory. After analyzing the partial differential equations to be solved it prepares the global system matrices and residual vectors and starts solving. If necessary it starts a newton iteration for nonlinear problems and adapts the grid where the error-level is greater than a given epsilon area. After finishing it writes the results on a file either for later reuse or just for visualization tools (Figure 2).

## 2.2 THE ANALYTICAL MODEL INTERFACE

The Analytical Model Interface (AMI) is an interpreter for translating mathematical expressions describing a physical model into a numerical one, highly optimized for numerical simulation. Within this tool the models are defined and therefore it is just used in developer mode. The model description language (Figure 3) is similar to well known systems like Math-CAD, Mathematica or Matlab, on the one hand because of the physical and mathematical background and on the other hand to reduce the break-in period for scientists who are not experienced in high level programming languages like C, C++ or Fortran. The user has to set up the quantities to be solved for, can select any kind of discretization (finite elements, finite differences or finite boxes) and has just to define the discretized residual vector and its derivative. To simplify the interactive process in user mode, several parameters and derivative quantities can be defined, too. This special quantities and parameters are detected by AMIGOS and are treated in a special manner so that the per-

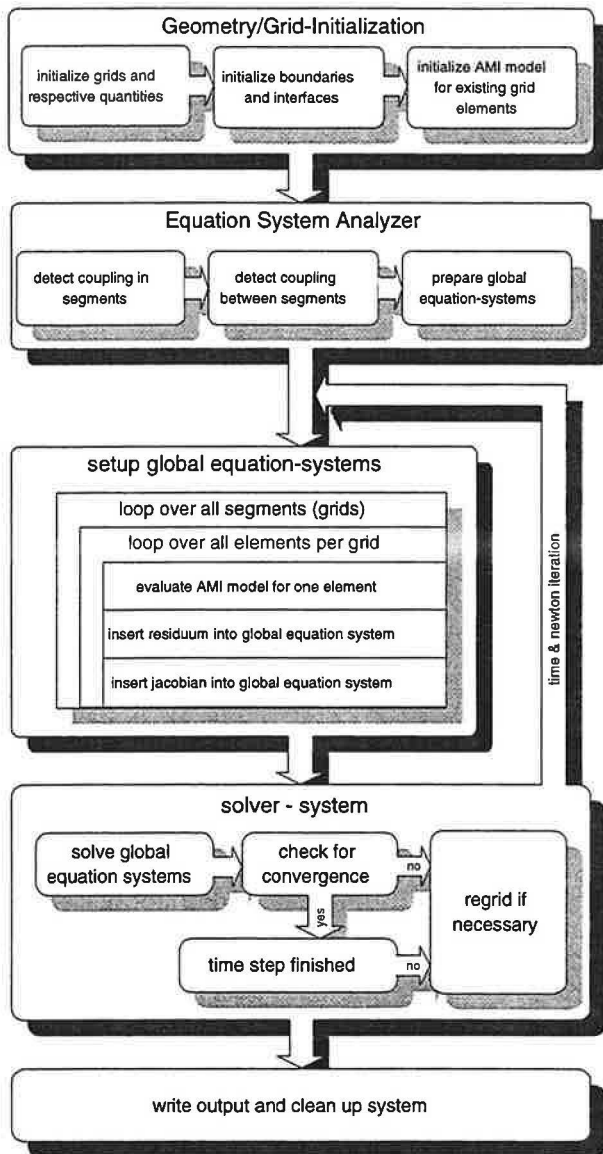


Figure 2: AMIGOS'S FLOW DIAGRAM OF A COMPLETE SIMULATION PROCESS

formance of the system is as high as possible but can be changed even in user-mode. Therefore AMI is also equipped with an analytical as well as numerical optimizer which is tailor-made for matrix operations and standard mathematical expressions (e.g. +, -, /, \*, sin, cos, atanh, ...) to minimize the number of operations for evaluation of a defined model. Every calculated mathematical term is stored and is reused whenever a similar term is detected (taking commutative law, associative law, etc. into account). Operations like multiplying by zero or one etc. are detected and immediately eliminated which reduces the amount of operations drastically especially when using matrices. Therefore the number of mathematical operations of a complete model can internally be reduced considerably (about two third) and leads to a very high performance despite its high flexibility.

```

MODEL ModelName = [x1,x2,...,xn];

# MODEL is a predefined keyword
# ModelName: the name the model should get
# X = [x1,x2,x3,...] the solution-vector
{
  # Begin Model Description

  # any user defined mathematical expression
  var1 = x1+5*x2;
  var2 = var1+2*x3;
  var3(var1,var2) = var1 - var2;
  ...
  # running index-variable from 1 to n
  # just defined for example --->
  # ---> and there are a lot of more such features
  i = 1..n;
  func[i] = 2*X[i];

  # Definition of residual and its derivative to
  # get the numerical form:
  #      [jacobian] * [X] = [residual]

  residual = [[f1(x1,x2,...,xn)]
              [f2(x1,x2,...,xn)]
              [ ... ]
              [fn(x1,x2,...,xn)]];
  jacobian = [[df1/dx1] [df1/dx2] ... [df1/dxn]
              [df2/dx1] [df2/dx2] ... [df2/dxn]
              ...
              [dfn/dx1] [dfn/dx2] ... [dfn/dxn]];
}
  
```

Figure 3: GENERAL MODEL DESCRIPTION LANGUAGE

## 2.3 THE EQUATION SYSTEM ANALYZER

AMIGOS is equipped with a so called Equation System Analyzer which detects all couplings among different quantities within a volume model as well as connections between several different models because of boundary conditions. As a result, the Equation System Analyzer distributes independent quantities to separate global matrices so that the amount of memory and evaluation time can be kept to a minimum. Furthermore, it prepares all necessary global stiffness matrices for the integrated numerical solver which can either be used as an iterative solver or, if necessary, as a direct Gaussian solver.

## 2.4 THE GRID- & ELEMENT-MANAGER

At least AMIGOS has also got an integrated Grid- & Element-Manager [2] which automatically detects the kind of elements defined on a given grid as well as its dimension. It is equipped with an adaptive-hierarchical grid algorithm thus the local discretization can be well fitted to the given physical problem.

### 3 SOLUTION OF A NONLINEAR FIELD-DEPENDENT DIFFUSION PROCESS OF TWO CHARGED DOPANTS

Diffusion is one of the basic steps in process simulation and because of the inaccessible measurement methods of dopant concentrations within silicon the prediction of simulation results is very important in micro-electronic device development. To demonstrate the capabilities of AMIGOS we present a field dependent coupled diffusion process of two charged dopants (Arsenic and Boron).

In presence of an electric field, the dopants are driven by two forces, the concentration gradient, and the electric field which acts on all charged particles. Thus the differential equation for diffusion has to be extended by an electrostatic potential  $\psi$  which accounts for the dopant's self induced flux including field enhancement as well as the flux induced by other dopants. This fact can be modeled with the following partial differential equation [3],

$$\frac{\partial C_i}{\partial t} = \text{div} \left[ D_i(C_i) \left( \text{grad}(C_i) + Z_i \frac{C_i}{U_T} \text{grad}(\psi) \right) \right]$$

where  $C_i$  represents the dopant concentration,  $D_i$  its diffusivity,  $Z_i$  its charge state and  $U_T = \frac{kT}{q}$  the thermal voltage. Furthermore, the coupling electrostatic potential  $\psi$  is calculated by

$$\text{grad}(\psi) = \frac{U_T}{\sqrt{4n_i^2 + C_{net}^2}} \sum Z_j \text{grad}(C_j)$$

assuming Boltzmann statistics for the electron and hole concentration.

The simulation itself has been started from an unsymmetrical Gaussian distribution of arsenic (Figure 4) with a peak of  $10^{20} \text{cm}^{-3}$  and boron of a constant concentration of  $10^{15} \text{cm}^{-3}$  as a background doping. The typical simulation result (Figure 5) shows the equilibrium state of the background doping and its deviation from the initial constant distribution just because of the electrical field caused by the different charge state of the dopants.

### 4 CONCLUSION

The consequent progress in semiconductor device fabrication makes it necessary that simulation tools have to become increasingly flexible to keep up the pace of advance. With AMIGOS we have developed a powerful model development tool which supports the major interests of numerical engineering.

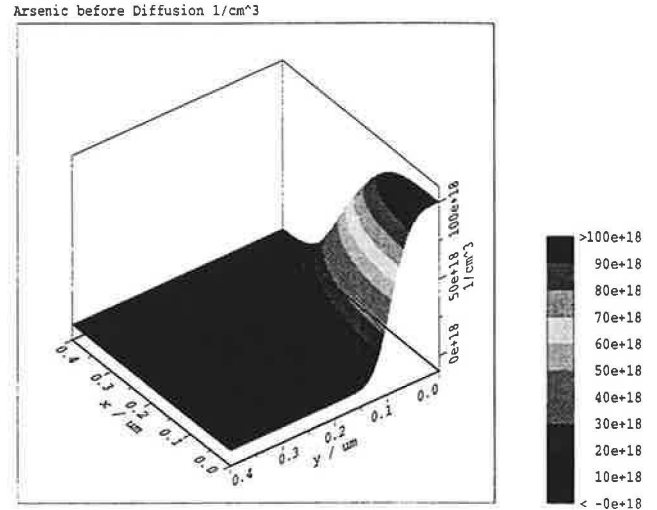


Figure 4: INITIAL DISTRIBUTION OF ARSENIC

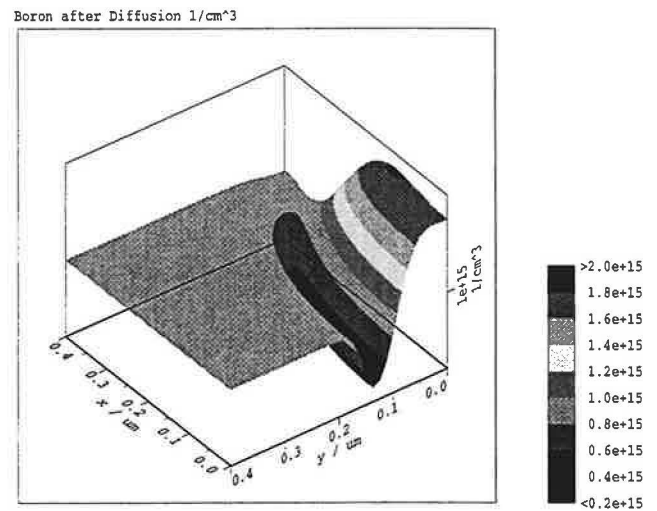


Figure 5: TYPICAL FIELD DEPENDENT DISTRIBUTION OF THE BACKGROUND-DOPING (BORON) AFTER DIFFUSION PROCESS

### References

- [1] D.W. Yergeau, E.C. Kan, M.J. GANDer, and R.W. Dutton. Alamode: A layered model development environment. In Ryssel and Pichler [4], pages 66–69.
- [2] E. Leitner and S. Selberherr. Three-dimensional grid adaptation using a mixed-element decomposition method. In Ryssel and Pichler [4], pages 464–467.
- [3] S.Wolf. *Silicon Processing for VLSI Era – The Sub-micron MOSFET*, volume 3. Lattice Press, 1995.
- [4] H. Ryssel and P. Pichler, editors. *Simulation of Semiconductor Devices and Processes*, volume 6, Wien, 1995. Springer.