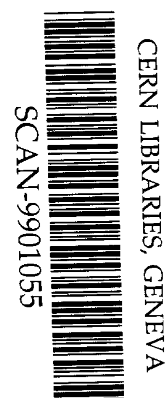
**Technical Report**
RAL-TR-1998-082




# The EVEREST Solver Module: Version 4.0

**J V Ashby R F Fowler and C Greenough**

6th January 1999

COUNCIL FOR THE CENTRAL LABORATORY OF THE RESEARCH COUNCILS

**© Council for the Central Laboratory of the Research Councils 1997**

Enquiries about copyright, reproduction and requests for additional copies of this report should be addressed to:

The Central Laboratory of the Research Councils
Library and Information Services
Rutherford Appleton Laboratory
Chilton
Didcot
Oxfordshire
OX11 0QX
Tel: 01235 445384     Fax: 01235 446403
E-mail library@rl.ac.uk

**ISSN 1358-6254**

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

# The EVEREST Solver Module: Version 4.0

JV Ashby, RF Fowler and C Greenough

December 1998

**Abstract**

In this report we describe Version 4.0 of the EVEREST Solver Module which forms part of the EVEREST semiconductor device modelling suite of programs. The solution module solves the semiconductor device equations in steady state or transient mode in three dimensions. The continuous equations are spatially discretised by the control region method on a mixed mesh of tetrahedral and hexahedral elements. The temporal discretisation of the continuity equations is performed by the variable order, variable time step Gear method. The nonlinear and the linear systems of equations are solved using damped Newton with correction transformation and incomplete Choleski conjugate gradient and conjugate gradient squared methods with Eisenstat's preconditioning. For physical models, the user has the choice of several mobility models namely; lattice, impurity concentration dependent and field dependent. Recombination rate can be either zero or a combination of Shockley-Read-Hall and Auger. The effects of bandgap narrowing due to heavy doping may also be modelled. Surface charge layers may be specified within the device. Transient simulation allows a choice of functions to define the bias variation on the contacts.

The EVEREST suite is one of the products of the ESPRIT project EVEREST (ESPRIT 962E-17, *Three-Dimensional Algorithms for a Robust and Efficient Semiconductor Simulator with Parameter Extraction*). The original authors of the Solver Module were D. Gunasekera of University College, Swansea and R.F. Fowler and C. Greenough of the Rutherford Appleton Laboratory.

A copy of this report can be found at the Department's web site (*http://www.dci.clrc.ac.uk/*) under page *Group.asp?DCICSEMSW* or anonymous ftp server *www.inf.rl.ac.uk* under the directory *pub/mathsoft/publications*

---

Mathematical Software Group
Department for Computation and Information
Rutherford Appleton Laboratory
Chilton, DIDCOT
Oxfordshire OX11 0QX

# Contents


**1 Introduction** . . . . . **1**

**2 Computational Methods** . . . . . **2**
- 2.1 The Governing Equations . . . . . 2
- 2.2 Physical Models . . . . . 3
- 2.3 Spatial Discretisation . . . . . 6
- 2.4 Temporal Discretisation . . . . . 6
- 2.5 Nonlinear Solution Process . . . . . 7
- 2.6 Initialisation . . . . . 7
- 2.7 Linear Solution Process . . . . . 7
- 2.8 Adaption . . . . . 7

**3 Using the Program** . . . . . **10**
- 3.1 Interface Files . . . . . 10
- 3.2 User Interface . . . . . 11
- 3.3 Command Reference . . . . . 12

**4 Examples and results** . . . . . **14**
- 4.1 Diode containing one dimensional effects . . . . . 14
- 4.2 Diode containing three dimensional effects . . . . . 14
- 4.3 Corner diode with oxide overlay . . . . . 17
- 4.4 Diode under transient applied bias . . . . . 21
- 4.5 Mesh refinement for current flow around an oxide corner . . . . . 21

**A Internal Commands** . . . . . **27**
- A.1 MORE - to display the contents a file . . . . . 28
- A.2 CHANGE - to change working directory . . . . . 28
- A.3 RENAME - to rename a file . . . . . 28
- A.4 COPY - to copy a file . . . . . 29
- A.5 RM - to delete (remove) a file . . . . . 29
- A.6 LIST - to provide directory listing . . . . . 29
- A.7 WRITE - to provide monitoring of a session . . . . . 30
- A.8 READ - to specify a command input file . . . . . 30
- A.9 SYNTAX - to provide the syntax of a command . . . . . 31
- A.10 HELP - to access HELP system . . . . . 32

**B Application Commands** . . . . . **34**
- B.1 BIASING - to specify contact potentials . . . . . 35
- B.2 CATALOGUE - to specify the catalogue neutral file name . . . . . 36
- B.3 CHARGE - to specify surface charge . . . . . 36
- B.4 COMMENT - to INSERT comments in the command file . . . . . 37
- B.5 CUEVENT - to include a charge upset event . . . . . 37
- B.6 DOPING - to specify doping neutral file name . . . . . 38
- B.7 END - to end the program run . . . . . 39
- B.8 GEOMETRY - to specify the geometry neutral file name . . . . . 39

B.9 INTEGRATE - to specify contact to integrate $J_p$, $J_n$ . . . . . . . . . . . . . . . . . . 40
B.10 MATERIAL - to change material properties . . . . . . . . . . . . . . . . . . . . . . . . 41
B.11 MESH - to specify mesh neutral file name . . . . . . . . . . . . . . . . . . . . . . . . . . 45
B.12 MODELS - to select physical models . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
B.13 OPTIONS - to specify the run-time options . . . . . . . . . . . . . . . . . . . . . . . . 47
B.14 OUTPUT - to specify results neutral file name . . . . . . . . . . . . . . . . . . . . . 48
B.15 PHYSICS - to specify physics neutral file name . . . . . . . . . . . . . . . . . . . . . 49
B.16 SOLVE - to start solution sequence . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
B.17 STEPS - to specify initial bias fractions . . . . . . . . . . . . . . . . . . . . . . . . . . 54
B.18 TIMES - to specify transient output times . . . . . . . . . . . . . . . . . . . . . . . . 55
B.19 TITLE - to identify a program run . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
B.20 TRANSIENT - to specify transient biasing . . . . . . . . . . . . . . . . . . . . . . . . 56

# 1 Introduction

This manual describes the solution module of the EVEREST semiconductor device modelling software. This constitutes the fourth release of the solver module with solution of the full 'on-state' transient semiconductor problem with adaptive meshing for steady state simulations. This version also includes charge generation events for modelling the effect of X-ray and particle strikes within semiconductor devices.

The EVEREST suite consists of five stand-alone modules, of which the solver module performs the solution of the semiconductor device equations. The other modules are the pre-processor [2], the doping generator [3], the post-processor [4] and inimsh, the mesh generator for adaption. Three types of simulations are allowed by the solver. They are, steady state with a fixed mesh, transient with a fixed mesh and steady state with adaptive meshing. The geometry definition and the generation of meshes for regular geometries are performed by the pre-processor, more general geometries are meshed by the automatic mesh generator, the doping profile is generated by the doping module and the results are viewed using the post-processor. The information from the pre-processor and the doping module is supplied to the solver module in a set of formatted files with extensions .GEO, .MSH and .DOP, known as *neutral files*. Likewise, the results from the solver module are transmitted to the post-processor in another set of neutral files with extensions .RES, .CAT and .PHY. Figure 1 is a schematic representation of the information flow. In a simulation involving adaptive meshing additional information on the mesh is passed to the solver module via a mesh database file. The solver will generate new mesh and doping files describing the refined mesh.

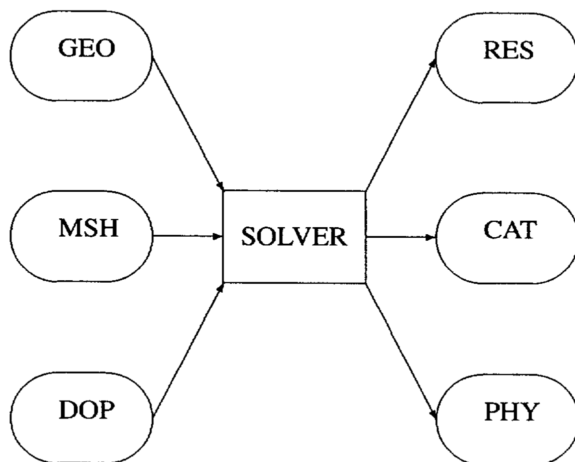



Figure 1: Program Environment

The following section outlines the device equations, the physical models and the algorithms used. Section 3 describes how to use the program, giving details of the interface files, command environment and control of the solution procedure. The final section gives some examples of the use of the solver, and the Appendices contains details of all the commands.

# 2 Computational Methods

This section details the semiconductor device equations, lists the physical models used and gives an outline of the numerical techniques employed in EVEREST. In the current version, the solution module approximately solves the semiconductor and insulator equations in steady state or transient mode in three dimensions. The continuous equations are spatially discretised by the control region method on a mixed mesh of tetrahedral and hexahedral elements. The temporal discretisation of the continuity equations is performed by the variable order variable time step method of Gear. The nonlinear and the linear systems of equations are solved using damped Newton with correction transformation and incomplete Choleski conjugate gradient and conjugate gradient squared methods with Eisenstat's preconditioning. As for physical models, the user has the choice of several mobility models namely; lattice, impurity concentration dependent and field dependent. Recombination rate can be either zero or a combination of Shockley-Read-Hall and Auger. A future release will include impact ionization and surface recombination. The effects of bandgap narrowing due to heavy doping may also be modelled. Surface charge layers may be specified within the device. In a transient simulation you are given a choice of functions to define the bias variation on any contact.

## 2.1 The Governing Equations

Semiconductor physics are characterised by three partial differential equations in the drift-diffusion model [5], [6].

$$\epsilon.\nabla^2\psi = -\rho \tag{1}$$

$$q\frac{\partial p}{\partial t} = -\nabla.J_p - qR \tag{2}$$

$$q\frac{\partial n}{\partial t} = \nabla.J_n - qR \tag{3}$$

These equations need to be solved subject to a set of boundary and initial conditions. The most important of these boundary conditions are the applied biases on the contacts. In transient simulations these can be time dependent.

Poisson's equation, (1), relates the electrostatic potential to the charge concentration $\rho$. The carrier continuity equations, (2) and (3), relate the rate of change of hole concentration $p$ and the rate of change of electron concentration $n$ to the divergence of their respective currents $J_p$ and $J_n$ plus the recombination rate $R$.

The charge concentration $\rho$ is given by:

$$\rho = q(p - n - (N_A - N_D) + \rho_F) \tag{4}$$

Where $N_A$ and $N_D$ are acceptor and donor atom concentrations, $q$ is the electron charge and $\rho_F$ is the interface trapped charge density.

The expressions for the current densities given below are derived from the Boltzmann transport equation:

$$J_p = -q\mu_p(v_T\nabla p + p\nabla(\psi - v_T\log(n_i))) \tag{5}$$

$$J_n = q\mu_n(v_T\nabla n - n\nabla(\psi - v_T\log(n_i))) \tag{6}$$

where $v_T = k_BT$. The Fermi-Dirac function, combined with the density of states, yields expressions for electron concentration in the conduction band and the hole concentration in the valence band. Under nondegenerate conditions they simplify to the well known Boltzmann approximations, $p =$

$n_i \exp(\phi_p - \psi)/v_T$ and $n = n_i \exp(\psi - \phi_n)/v_T$, where $\phi_p$ and $\phi_n$ are the hole and electron quasi-Fermi levels.

Currently EVEREST only allows Ohmic type Dirichlet contacts, which are idealised by assuming infinite contact recombination velocities and space-charge neutrality. Then the carriers are in thermodynamic equilibrium and both quasi-Fermi levels are equal to the applied bias. The other form of contact which can appear is a Schottky barrier. This will be implemented in a future release. At an Ohmic contact,

$$\phi_p = \phi_n = V_{app} \tag{7}$$

and charge neutrality gives:

$$p - n - (N_A - N_D) = 0 \tag{8}$$

Using (7) and (8), in conjunction with Boltzmann approximations for carrier concentrations, gives Ohmic Dirichlet boundary conditions for $\psi$, $p$ and $n$. The remainder of a device boundary in EVEREST is of homogeneous Neumann type, where the hole and the electron current densities and the electric field strength normal to the boundary vanish, yielding

$$\nu.J_n = \nu.J_p = \nu.\nabla\psi = 0 \tag{9}$$

where $\nu$ is the surface normal.

In a device containing oxide and semiconductor regions, such as a MOSFET, Poisson's equation applies to the whole device, where as the application of the continuity equations is limited to the semiconductor regions. A metal contact existing on an oxide region gives its applied bias added to the workfunction difference to the electrostatic potential in the oxide and the oxide/semiconductor interface acts as a Neumann boundary to the continuity equations. As noted earlier, EVEREST allows the inclusion of fixed interface charges at such layers and provision will be made in a future release for surface recombination.

### 2.2 Physical Models

This section lists the physical models currently implemented within EVEREST. They are accessed via the MODELS command. The parameter values can be changed via the MATERIAL command. Each model contains a set of default parameters which can be altered by the user at run time. A detailed discussion and treatment of these models is given in [6].

**Intrinsic carrier density models**

- Constant

$$n_i = A \tag{10}$$

- Temperature dependent (default)

$$n_i = \sqrt{AT^3} \exp(-\frac{B}{2V_T}) \tag{11}$$

**Bandgap narrowing**

$$n_{ie} = n_i \exp(d_{gap}/2V_T) \tag{12}$$

- None (default)

$$d_{gap} = 0 \tag{13}$$

- Impurity dependent
  When holes are majority carriers ( IF $N_A > N_D$)

$$d_{gap} = A\left[\log(N/N_{ref1}) + \sqrt{(\log(N/N_{ref1}))^2 + B}\right] \tag{14}$$

where $N$ is given by:

$$N = N_D + N_A \tag{15}$$

When electrons are majority carriers (ELSE)

$$d_{gap} = C\log(N/N_{ref2});\ N_D > N_{ref2} \tag{16}$$

$$d_{gap} = 0;\ N_D \leq N_{ref2} \tag{17}$$

where $N$ is given by (15).

**Recombination**

- None

$$R = 0 \tag{18}$$

- Shockley-Read-Hall (default)

$$R = \frac{np - n_{ie}^2}{\frac{(n+n_{ie})}{1/\tau_p + N/C_p^{SRH}} + \frac{(p+n_{ie})}{1/\tau_n + N/C_n^{SRH}}} \tag{19}$$

where $N$ given by (15).

- Auger recombination

$$R = (C_n n + C_p p)(pn - n_{ie}^2) \tag{20}$$

- Avalanche generation *(not yet implemented)*

$$R = (\alpha_n \exp(-b_n/\frac{|E.\nabla\phi_n|}{|\nabla\phi_n|})|J_n| + \alpha_p \exp(-b_p/\frac{|E.\nabla\phi_p|}{|\nabla\phi_p|})|J_p|)q^{-1} \tag{21}$$

- Surface recombination *(not yet implemented)*

$$R = \frac{np - n_{ie}^2}{\frac{(n+n_{ie})}{s_p} + \frac{(p+n_{ie})}{s_n}} \tag{22}$$

**Carrier mobility**

- Temperature dependent

$$\mu = \mu_0(\frac{T}{300})^{-\alpha} \tag{23}$$

- Impurity dependent (default)
  Hole mobility when majority carrier (IF $N_A > N_D$)

$$\mu(N) = \mu_{min} + \frac{\mu_{max}(\frac{T}{300})^{-\beta} - \mu_{min}}{1 + (N/N_{ref})^{\alpha}} \tag{24}$$

  where $N$ given by (15).
  Hole mobility when minority carrier (ELSE)

$$\mu(N) = \mu_{majority} \frac{fN_D + N_A + N_{ref}}{N_D + N_A + N_{ref}} \tag{25}$$

  Electron mobility when majority carrier (IF $N_D > N_A$)

$$\mu(N) = \mu_{min1} + \frac{\mu_{max}(\frac{T}{300})^{-\beta} - \mu_{min1}}{1 + (N/N_{ref})^{\alpha}} \tag{26}$$

  where $N$ given by (15).
  Electron mobility when minority carrier (ELSE)

$$\mu(N) = \mu_{min2} + \frac{\mu_{max}(\frac{T}{300})^{-\beta} - \mu_{min2}}{1 + (N/N_{ref})^{\alpha}} \tag{27}$$

  where $N$ given by (15).

- Field dependent
  Parallel field dependent

$$\mu(N, E_{par}) = \mu(N) \cdot \frac{1}{[1 + (\mu(N)\frac{|E.\nabla\phi|}{|\nabla\phi|}/v_{max})^{\beta}]^{1/\beta}} \tag{28}$$

  Total field dependent *(not yet implemented)*

$$\mu(N, E) = \mu(N, E_{par}) \cdot \frac{1}{[1 + c(E_{nor} - E_0)]} : \; E_{nor} > E_0 \tag{29}$$

$$\mu(N, E) = \mu(N, E_{par}) : \; E_{nor} \leq E_0 \tag{30}$$

  where

$$E^2 = E_{par}^2 + E_{nor}^2 \tag{31}$$

**Interface charge**

- Trapped interface charges can be included as fixed a density in $q/cm^2$.

**Default values**

The default values of all the above constants are given in Table 1. As noted above, all the defaults may be modified by the using the MATERIAL command.

| Process | Model | Parameters |
|---|---|---|
| Intrinsic | Constant | $n_i = 1.45 \times 10^{10}$ |
| carrier | Temp. dep. | $A = 9.6 \times 10^{32}$, $B = 1.206$ |
| Bandgap | Impu. dep. | $A = 9.0 \times 10^{-3}$, $B = 0.5$, $N_{ref1} = 10^{17}$, |
| | | $C = 18.7 \times 10^{-3}$, $N_{ref2} = 7.0 \times 10^{17}$ |
| Recombination | SRH | $\tau_p = 10^{-5}$, $\tau_n = 10^{-5}$, $C_p^{SRH} = 1.3 \times 10^{12}$, |
| | | $C_n^{SRH} = 5.0 \times 10^{11}$ |
| | Auger | $C_p = 0.95 \times 10^{-31}$, $C_n = 2.8 \times 10^{-31}$ |
| | Avalanche | $\alpha_p = 2.25 \times 10^7$, $b_p = 3.26 \times 10^6$, |
| | | $\alpha_n = 3.8 \times 10^6$, $b_n = 1.75 \times 10^6$ |
| | Surface | $s_p = 0$, $s_n = 0$ |
| Mobility | Temp. dep. ($p$) | $\mu_0 = 4.5 \times 10^2$, $\alpha = 0.61$ |
| | Temp. dep. ($n$) | $\mu_0 = 1.5 \times 10^3$, $\alpha = 0.91$ |
| | Impu. dep. ($p$) | $\mu_{min} = 47.7$, $\mu_{max} = 495$, $N_{ref} = 6.3 \times 10^{17}$, |
| | | $\alpha = 0.7$, $\beta = 0.61$, $f = 2.47$ |
| | Impu. dep. ($n$) | $\mu_{min1} = 92$, $\mu_{min2} = 230$, $\mu_{max} = 1360$, |
| | | $N_{ref} = 1.3 \times 10^{17}$, $\alpha = 0.91$, $\beta = 0.91$ |
| | Field Dep. ($p$) | $v_{max} = 9.5 \times 10^6$, $\beta = 1.0$ |
| | Field Dep. ($n$) | $v_{max} = 1.1 \times 10^7$, $\beta = 2.0$ |

Table 1: Default parameter values (all units are in $cm$, $V$, $s$)

## 2.3 Spatial Discretisation

The spatial discretisation is performed by the control region approximation on an underlying mesh of tetrahedral and hexahedral elements [6]. The control region approximation is applicable to many general conservation laws and consists of performing discrete flux balances over control region boundaries. These control regions tessellate the whole of the solution domain without overlap and although they could be of any shape, in the solver they are simply connected convex polytopes surrounding nodes. The discrete fluxes are calculated element by element for the edges of an underlying tetrahedral/hexahedral mesh. The electric flux between two nodes is calculated according a finite difference formula, the current densities are calculated according to the Scharfetter Gummel formula and the charge and recombination terms are approximated to be piecewise constant at the nodal values. EVEREST uses the variable $\psi$, $\phi_p$ and $\phi_n$.

## 2.4 Temporal Discretisation

The temporal integration of the spatially discretised semiconductor device equations is performed by the variable order, variable time step predictor corrector method of Gear for stiff problems [9]. The standard Gear algorithm is for systems of ODEs and EVEREST contains appropriate modifications to treat the algebraic equations arising from Poisson's equation. The program automatically selects the order of the integration and the time step to compute a solution to a user defined tolerance on the local truncation error. The maximum order allowed is fifth.

## 2.5 Nonlinear Solution Process

The solution of the nonlinear equation system obtained by discretising the semiconductor equations can be split into three stages; initial guess, nonlinear iteration scheme and linear solution. In this section we discuss the nonlinear scheme.

The nonlinear process consists of an initial Gummel iteration followed by a fully coupled iteration process bound by a continuation method on the applied bias. The continuation method controls the magnitude of the applied bias step to ensure convergence in numerically difficult simulations. In the current version of the solver, the inner iterations of the Gummel process and the iterations of the fully coupled process are of damped Newton type with correction transformation. It is possible to turn off the damping process or omit the correction transformation. To save computational effort it is also possible to restrict the solution of equations to Poisson's equation only or to Poisson's equation and one continuity equation. When a continuity equation is not solved the corresponding quasi-Fermi level is held fixed at the initialised value, which is discussed in the next section.

The number of Gummel iterations prior to entering the fully coupled solver is controlled using the parameters GUMTOL and GUMLIM. When the update size is less than GUMTOL or when the number of Gummel iterations reaches GUMLIM the Gummel process is terminated and the coupled solver started. Default values allow one Gummel iteration before entering the coupled solver.

## 2.6 Initialisation

The initialisation scheme is a variation of that presented by Edwards *et al* [7] which projects quasi-Fermi levels from one solution point to another by assuming zero change in the local divergence of current. Solution of (32) and (33) with constant upto date $p$ and $n$, yields increments to quasi-Fermi levels, hence new values of $\phi_p$ and $\phi_n$ can be calculated.

$$\nabla(\mu_n n \nabla(\delta\phi_n)) = 0 \tag{32}$$

$$\nabla(\mu_p p \nabla(\delta\phi_p)) = 0 \tag{33}$$

The electrostatic potential is then determined by holding the majority carrier concentrations constant.

## 2.7 Linear Solution Process

The linear systems which arise from the nonlinear discretisation are sparse. To minimise storage requirements, the linear solver is written specially for sparse systems. Discretised Poisson's equation and Laplace's equations in the initialisation yield symmetric positive definite matrices which are solved by the conjugate gradient method with incomplete Choleski preconditioning [8]. The nonsymmetric systems which arise from the continuity equations and from the fully coupled problem are solved by the conjugate gradient squared method with incomplete LU preconditioning. By default the current implementation uses a variation of the preconditioning, due to Eisenstat [10], which avoids explicit construction of the incomplete LU factors.

## 2.8 Adaption

Automatic mesh adaption may be used to generate a mesh that captures the important features of the solution without use of an excessive number of nodes and elements. This section describes the implementation of the adaptive solution level and gives brief details of the method used.

Having executed the pre-processor, the mesh generator and the doping descriptor a set of neutral files will exist called <*name*>.GEO, <*name*>.MSH, <*name*>.DOP. The mesh file should contain a coarse mesh encompassing geometric features of the device. The mesh generator will also produce a binary file, normally called <*name*>.DBASE. The <*name*>.DBASE file contains information about the mesh structure. Since this file is overwritten in the course of an adaptive calculation it may be useful to keep a copy of it. If the mesh generator built into the preprocessor is used, instead of the standalone mesh generator *inimsh*, then the adaption file will be called HEX.DBS, though it contains the same information. The preprocessor only generates this file for small initial meshes.

The Solver module first solves the semiconductor device equations on the initial mesh. Control then passes to the refinement routine which introduces new nodes and elements in areas of high local discretization error and returns an enhanced mesh. The solver then calculates doping etc. at new nodes, solves the semiconductor equations at the new nodes, solves the semiconductor equations on the complete mesh and recalls the mesh refinement routine. This loop is continued until the mesh refinement does not introduce any more new nodes. The adaptive process is then terminated and the solution, the mesh and the doping is written to file.

Adaption at subsequent bias conditions is likely to refine the mesh further. In order to restrict the number of meshes and doping densities which need writing to file, the solutions at subsequent bias cases are written out only at the nodes on the mesh at the end of the first bias case. Therefore a simulation using adaptive meshing and containing several bias cases results in the creation of one new mesh file, one new doping file and a results file with the corresponding number of solutions. It is important to realise that this approach leads to some loss of nodal information without any loss of terminal information such as currents.

One disadvantage of adaption in this manner is that a large number of iterations may be required to generate a suitable mesh, each iteration requiring the solution of the device equations. A faster method that helps to define the location of junctions is to refine on the variation of doping. This can be done by setting the parameter MAXDOP, described below. This method helps with the initial refinement to resolve the junctions in the device. Refinement on potential is required to pick up other features, such as the channel in MOS devices. The refinement on doping will divide any edge in the mesh (of parent blocks) for which

$$\frac{h|N_1 - N_2|}{\max(|N_1|, |N_2|, N_{ref})} > l_{min} \tag{34}$$

is true, where $N_1$, $N_2$ are the net doping at either end of the edge. $N_{ref}$ and $l_{min}$ default to $10^{14} cm^{-3}$ and 0.05 microns respectively, and may be altered by the user. In practice a value of $N_{ref}$ of about $10^{-2}$ times the maximum doping in the device may be a reasonable initial guess. $l_{min}$ should be set to a minimum element size taking into account the size of important features and allowable number of elements.

The refinement based on potential uses the change in displacement field, $D = \epsilon E$, between two elements as a measure of the error associated with the existing edge size. Refinement is made if the components of $D$ normal to the common face of two elements fail to satisfy the criteria

$$\frac{|D^+ - D^-|h}{\epsilon} \frac{h}{h + l_{min}} < \varepsilon_{tol} \tag{35}$$

where $\varepsilon_{tol}$ is an absolute tolerance and $l_{min}$ is a soft limit of the size of the smallest element to be generated. The units of $\varepsilon_{tol}$ are volts and it represents the level of uncertainty in the potential due to the discrete representation used.

For refinement on hole current, the refinement criteria used is :

$$\frac{|J_p^+ - J_p^-|h}{\overline{\mu_p}(\overline{p + n_i})} \frac{h}{h + l_{min}} < \varepsilon_{tol} \tag{36}$$

where $\overline{\mu_p}$ is the average mobility, $\overline{p + n_i}$ the average hole density (limited by the intrinsic level) and $J_p$ is the hole current through the common element face. A similar limit on the minimum element size is introduced via $l_{min}$. Refinement on electron current uses the same expression, cast in terms of the corresponding electron variables.

The refined mesh is always written to the file REFINE.MSH and the corresponding nodal doping to REFINE.DOP. It is important to use both of these files when running the post-processor, rather than the original mesh and doping files. It is possible to perform further fixed or adaptive runs using REFINE.MSH as the starting mesh as long as you retain the associated database file in the latter case. The file REFINE.DOP is not suitable for re-use by the solver, but the original doping file may be used, since the solver reads the parameterised form of the doping information.

At run time the user is given control of several parameters to specify the degree of adaption. These are explained in the command reference in the appendices, but we just highlight the parameters to the solve command which correspond to the terms used in the above equations:

**RREFTOL** Refinement tolerance, $\varepsilon_{tol}$.

**LMINTOL** Limit on smallest element size, $l_{min}$.

**NREFTOL** Limit on doping refinement, $N_{ref}$.

**MAXDOP** Maximum number of nodes permitted on doping criteria.

**MAXPSI** Maximum number of nodes permitted on potential criteria.

**MAXJP** Maximum number of nodes permitted on hole current criteria.

**MAXNOD** Maximum number of nodes permitted in the mesh.

**MAXELS** Maximum number of elements permitted in the mesh.

**MAXTLS** Maximum number of tetrahedral elements permitted in the mesh.

The parameter MAXCEL controls the allocation of space for surface charge data on the refined mesh (*not yet implemented*). These parameters are needed to define the dimensions of certain arrays and limit the total amount of refinement performed on each equation. Some experimentation is usually required to find a satisfactory set of parameters for a given device geometry. In general it is better to adjust the tolerances so that refinement stops on the above criteria rather than letting it stop on reaching the maximum number of allowed nodes.

will only give information on the syntax of the named command and its parameters.

The syntax of a command, as described by the help tools, includes the name of each parameter plus its 'type', 'status' and 'value'. For example, in the HELP command there are two parameters named KEY and OPTION. By using the name of the parameter, you can vary the order in which the parameters are listed, or choose to miss out a parameter altogether. If the parameter name is not used, then the parameter order given above would be:

```
HELP KEY=<command name> OPTION=SYNTAX
```

It is not necessary to give the full command or parameter name. Permitted abbreviations are indicated by the uppercase portion of the name displayed by the help facility.

It is important to understand the 'types' of parameter used by the command decoder. The type of a parameter may be any one of **real, integer, string, real list, integer list, string list** or **choice**.

**REAL** Must be a signed real number. The characters D and E may be used to indicate the exponent of a real number.

**INTEGER** Must be an integer consisting of a sign and digits only. Any other characters will produce an error message and the entire command will be ignored.

**STRING** Must be a string of characters delimited by spaces.

**REAL LIST** List parameters are used for entering tables of data. In the case of real lists, the command decoder expects a list of real numbers. The list of values should be enclosed by parentheses and entries separated by commas or spaces, e.g. (1.0,2.0,3.0). There is no restriction on the number of entries in a list structure. Parentheses are not necessary when only one entry is given.

**INTEGER LIST** As for REAL LIST, but the entries are now integers.

**STRING LIST** As for REAL LIST, but the entries are now strings.

**CHOICE** This must be a single string which matches one of the string values in the choice list. The list of choices is given in the 'Current value' field displayed by the help facility.

The 'status' of a parameter may take one of three values; **required, retained** or **reset**. For a **required** parameter, a value must be given in the command line, otherwise the command will be rejected. For **retained** and **reset** parameters, it is not necessary to give a value, the parameter will be indicated by an uppercase string, while the alternatives will be in lower case. For **retained** parameters, the value given by the user will replace the default value for the duration of the session within the program, or until a further command overwrites it. For **reset** parameters, the default value cannot be changed.

## 3.3 Command Reference

A list of available commands is given below with a brief description of the purpose of the command. A full definition of each command, including syntax and parameters is given in the appendices. Note that the internal commands are common to all modules within the EVEREST suite, while the application commands are not.

| Application commands | |
|---|---|
| **BIASING** | to specify contact potentials |
| **CATALOGUE** | to specify the catalogue neutral file name |
| **CHARGE** | to specify surface charge |
| **COMMENT** | to insert a comment in the command file |
| **CUEVENT** | to include a charge upset event |
| **DOPING** | to specify the doping neutral file name |
| **END** | to end the program run |
| **GEOMETRY** | to specify the geometry neutral file name |
| **INTEGRATE** | to specify contact for node wise current integral |
| **MATERIAL** | to change material properties |
| **MESH** | to specify the mesh neutral file name |
| **MODELS** | to select physical models |
| **OPTIONS** | to specify run time options |
| **OUTPUT** | to specify the results neutral file name |
| **PHYSICS** | to specify the physics neutral file name |
| **SOLVE** | to start the solution sequence |
| **STEPS** | to specify initial bias fraction |
| **TIMES** | to specify transient output times |
| **TITLE** | to identify a program run |
| **TRANSIENT** | to specify transient biasing |
| Internal commands | |
| **HELP** | to access the help system |
| **READ** | to specify a command input file |
| **SYNTAX** | to provide the syntax of a command |
| **WRITE** | to provide monitoring of a session |
| **MORE** | to display the contents a file |
| **CHANGE** | to change working directory |
| **RENAME** | to rename a file |
| **COPY** | to copy a file |
| **RM** | to delete (remove) a file |
| **LIST** | to provide directory listing |

Table 2: EVEREST Solver commands

## 4 Examples and results

This section describes three steady state examples, a transient example and one use of automatic mesh refinement. In each case we list the solver commands to run the example and present some results. In the following sections it is assumed that geometry, mesh and doping neutral files are correctly generated using the pre-processor and the doping profile generator prior to using the solver module. The plots subsequent to the simulation are generated by the post-processor. A more complete set of examples can be found in [12] and [13].

### 4.1 Diode containing one dimensional effects

#### Description

The device has dimensions $10 \times 2 \times 5$ $\mu m$ in $x$, $y$ and $z$ directions and is divided into 2 blocks of equal size with their interface at $x = 5$. The blocks are uniformly doped with impurities of acceptor type to $10^{16} cm^{-3}$ in one block and of donor type to $5 \times 10^{16} cm^{-3}$ in the other. The mesh contains 40, 1 and 1 divisions in $x$, $y$ and $z$ directions which amount to 164 nodes and 40 hexahedral elements. The contacts are at $x = 0$ and $x = 10$ and the applied biases are $0V$ at the acceptor contact and $0V$ to $-1.0V$ at the donor contact. The geometry of the device and an isometric plot of the potential on the plane $z = 0$ are given in Figures 2 and 3 and the input to the solver is listed below. It is assumed that the files PN.GEO, PN.MSH and PN.DOP have already been created before running the solver.

#### Input to the solver

```
Everest: GEO PN
Everest: MES PN
Everest: DOP PN
Everest: OUT PN
Everest: PHY PN
Everest: CAT PN
Everest: BIAS C1 (0.0)
Everest: BIAS C2 (0.0,-0.2,-0.4,-0.6,-0.8,-1.0)
Everest: SOLVE
```

In this run the bias at contact `C1` is kept at zero while the voltage on `C2` is changed from zero to $-1.0V$ in steps of $-0.2V$. The steady state solution is saved to the file PN.RES at each of the six bias cases. The default physical models have been used.

### 4.2 Diode containing three dimensional effects

#### Description

The device consists of two hexahedral blocks of semiconductor of dimensions $1 \times 1 \times 0.5$ and $0.5 \times 0.5 \times 0.5$ $\mu m$. The larger is uniformly doped with impurity of acceptor type to $10^{15} cm^{-3}$ and the smaller is doped with impurities of donor type to $10^{15} cm^{-3}$. The contacts are at the two faces parallel to the interface. The mesh contains $4 \times 4 \times 4$ divisions in the smaller block and $8 \times 8 \times 4$ in the larger in $x$, $y$ and $z$ directions respectively. This generates 505 nodes and 320 hexahedral elements. The

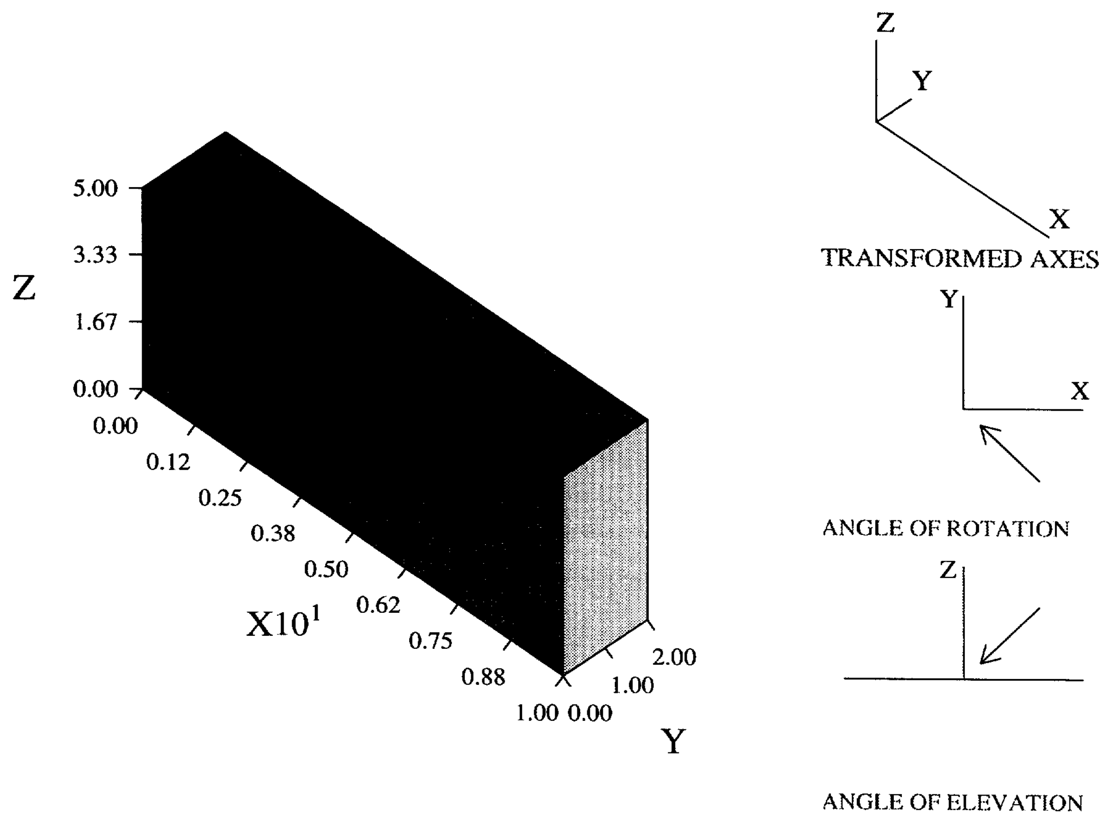



Figure 2: Geometry and mesh of the 1D diode.

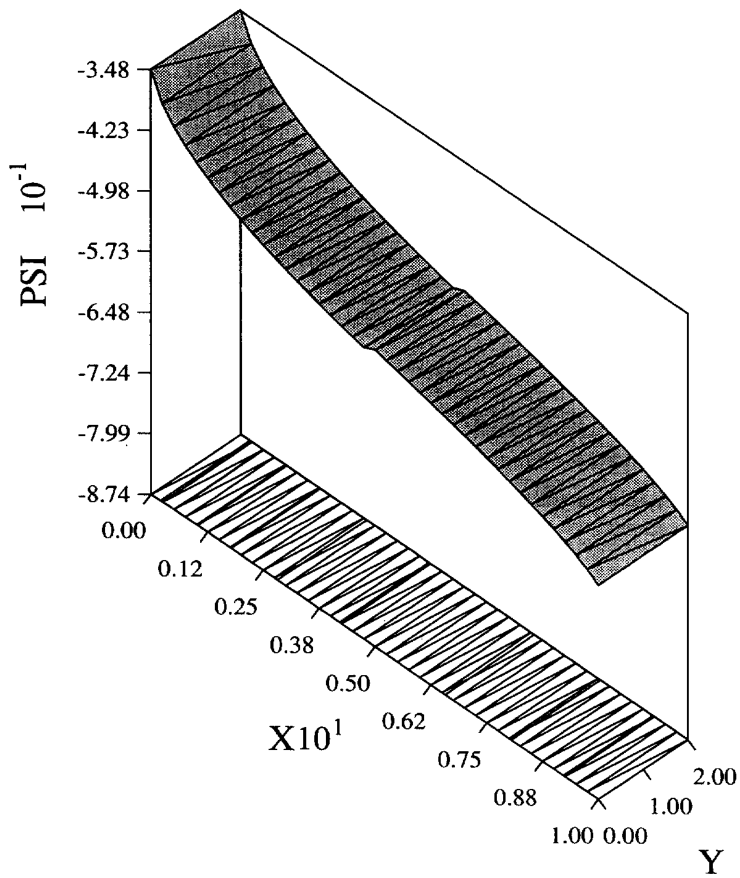



Figure 3: Isometric plot of potential on the Z=0 plane of the 1D diode ($V_{C2} = -1$)

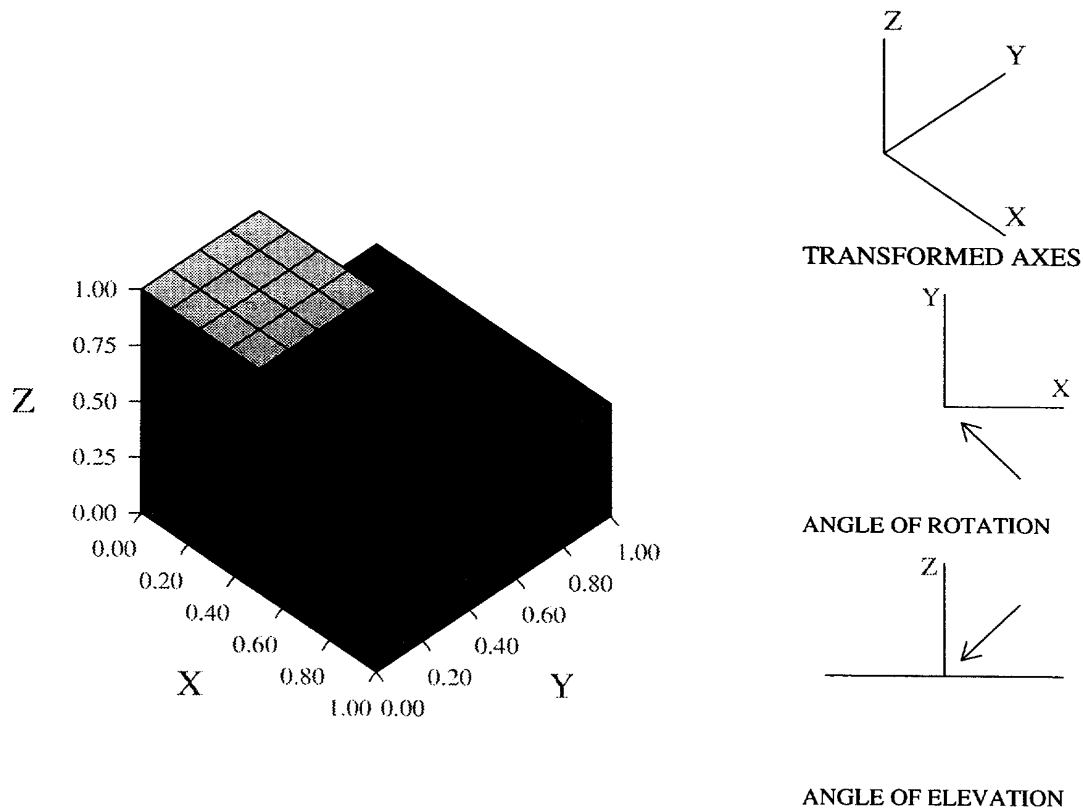



Figure 4: Geometry and mesh of the blocks diode

device is simulated with $0V$ on the donor contact and $0V$ to $1V$ on the acceptor contact. The geometry of the device is shown in Figure 4 and a vector plot of current flow in Figure 5 on the plane $x = 0$.

**Input to the solver**

```
Everest: GEO BLOCKS
Everest: MES BLOCKS
Everest: DOP BLOCKS
Everest: OUT BLOCKS
Everest: PHY BLOCKS
Everest: CAT BLOCKS
Everest: BIAS TOP (0.0)
Everest: BIAS BOTTOM (0.0,0.2,0.4,0.6,0.8,1.0)
Everest: SOLVE
```

### 4.3 Corner diode with oxide overlay

**Description**

The device consists of a block of semiconductor with an oxide overlay except for one corner, where a contact covering the whole corner and part of the top surface of the oxide is located. The other contact

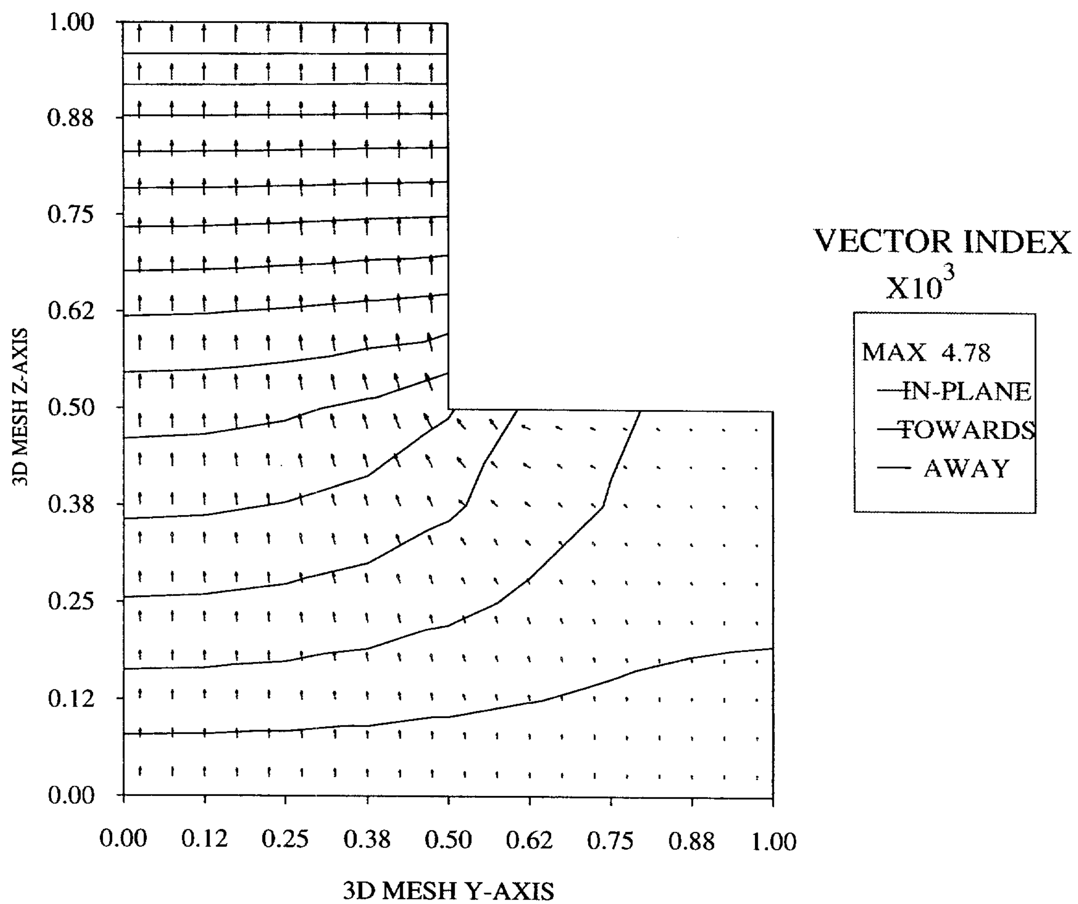



Figure 5: Vector plot of the total current on the X=0 plane of the blocks diode ($V_{bot} = 1$). The lines are contours of potential.

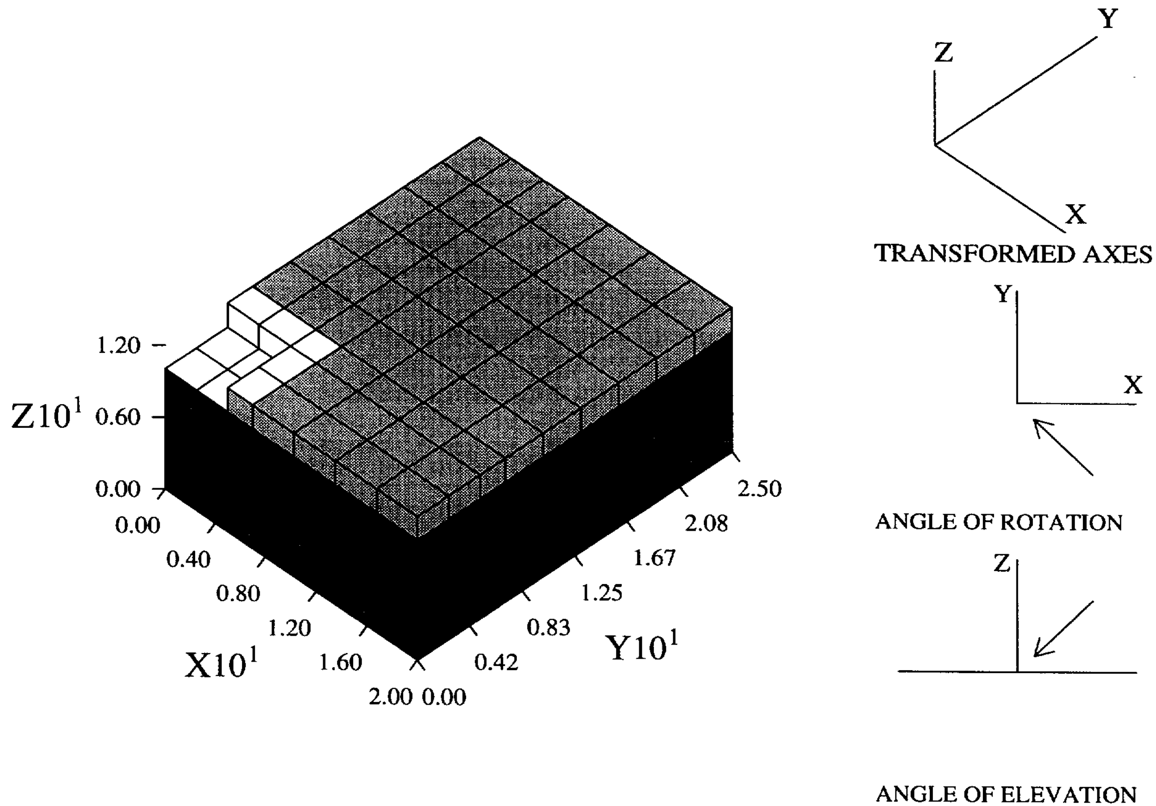



Figure 6: Geometry of the corner diode

is located on the base of the semiconductor parallel to the oxide semiconductor interface. The doping profile is generated firstly by implanting a uniform background doping of donor type at $5 \times 10^{16} cm^{-3}$ and then two Gaussian profiles of peak values $10^{18} cm^{-3}$ and $10^{17} cm^{-3}$ of type acceptor. A mesh of 476 nodes and 311 hexahedral elements is defined. The device is simulated with $0V$ on the acceptor contact and $0V$ to $-1.0V$ on the donor contact. The device geometry is shown in Figure 6. As an illustration of the output, the potential on the plane $z = 10$ is shown in Figure 7. The commands to run this simulation are listed below.

**Input to the solver**

```
Everest: GEO CORNER
Everest: MES CORNER
Everest: DOP CORNER
Everest: OUT CORNER
Everest: PHY CORNER
Everest: CAT CORNER
Everest: BIAS TOP (0.0)
Everest: BIAS BOTTOM (0.0,-0.2,-0.4,-0.6,-0.8,-1.0)
Everest: SOLVE
```

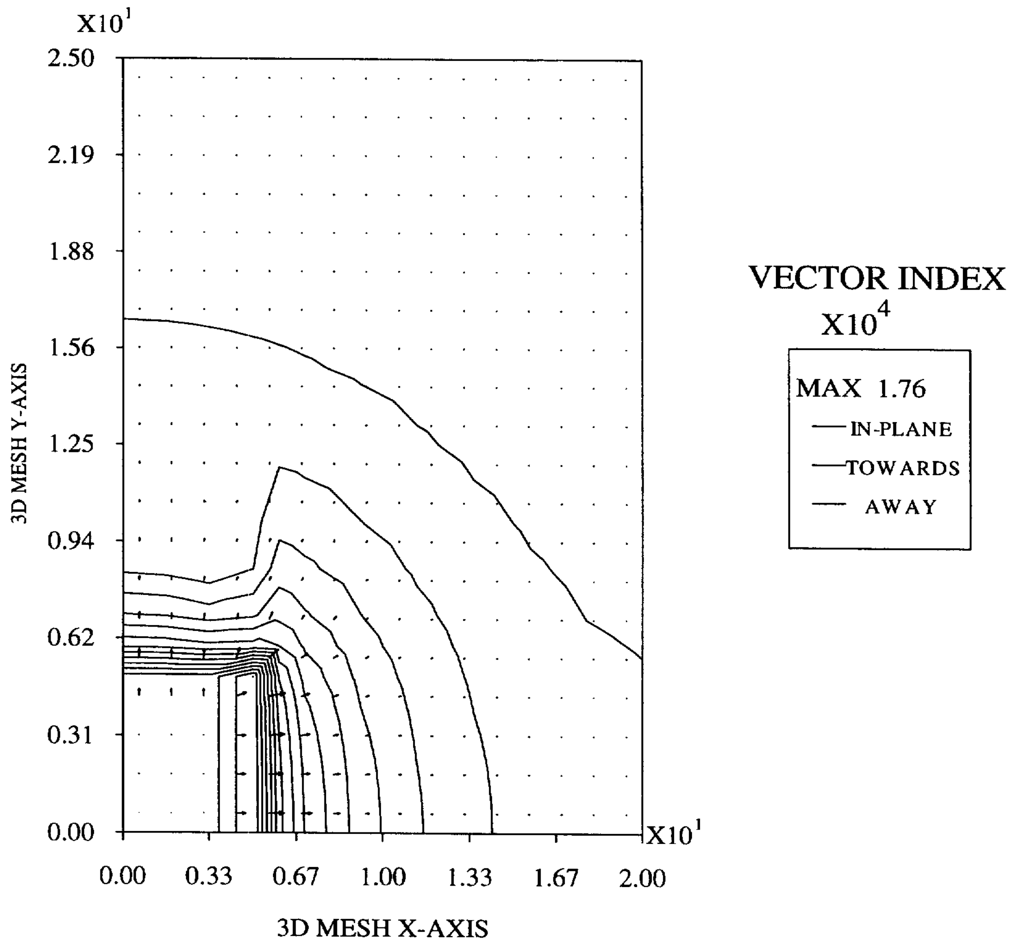



Figure 7: Contours of potential on the $z = 10$ plane of the corner diode with a forward bias of $-1V$. Also shown are vectors of current flow.

### 4.4 Diode under transient applied bias

#### Description

The device has dimensions $10 \times 2 \times 5\ \mu m$ in $x$, $y$ and $z$ directions and is divided into 2 blocks of equal size with their interface at $x = 5$. The blocks are uniformly doped with impurities of acceptor type to $10^{18} cm^{-3}$ in one block and of donor type to $10^{18} cm^{-3}$ in the other. The mesh contains 80, 1 and 1 divisions in $x$, $y$ and $z$ directions which gives 324 nodes and 80 hexahedral elements. The contacts are at $x = 0$ and $x = 10$. This device has the same structure as in Figure 2. Before starting a transient run a steady state solution has to be computed to provide the initial conditions. Therefore the first solve command is used to obtain this initial state.

The TRANSIENT command specifies a $-1.0V$ change applied to the C2 contact linearly in $1ns$ starting at zero time. The TIMES command requests the solution to be written to file at times 1, 2 and $3ns$.

The SOLVE command instructs the solver to read the first solution in the file `PN.RES` and to commence a transient run from it. It is important to supply the boundary conditions corresponding to the STARTing case and to all the other cases in file before it.

#### Input to the solver

```
Everest: GEO PN
Everest: MES PN
Everest: DOP PN
Everest: OUT PN R
Everest: PHY PN R
Everest: CAT PN R
Everest: BIAS C1 0.0
Everest: BIAS C2 0.0
Everest: SOLVE
Everest: TRANSIENT C2 LINEAR -1.0 1.0D-9 0.0
Everest: TIMES (1.0D-9,2.0D-9,4D-9,5D-8)
Everest: SOLVE START=1 TYPE=TRANS TRANTOL=1D-4,
               CGSTOL=1D-10 COUPTOL=1D-9
```

The hole and electron currents on the contact `C2` are shown in Figure 8.

### 4.5 Mesh refinement for current flow around an oxide corner

#### Description

This is a simple test problem to illustrate the use of automatic mesh refinement. The structure used here is a uniformly doped $n$-type region in which current flows around an oxide corner. Figure 9 shows the geometry of the device which is effectively two dimensional. Current flows around the oxide from one contact to the other. The coarse initial mesh used in the simulation is also shown in the Figure, which contains 98 nodes and 36 hexahedral elements. The doping in the silicon region is a uniform level of $10^{17} cm^{-3}$.

Since the doping is uniform, there is no point in attempting doping refinement in this case. Also the heavy $n$-type doping means that the hole current $J_p$ will be negligible compared to the electron

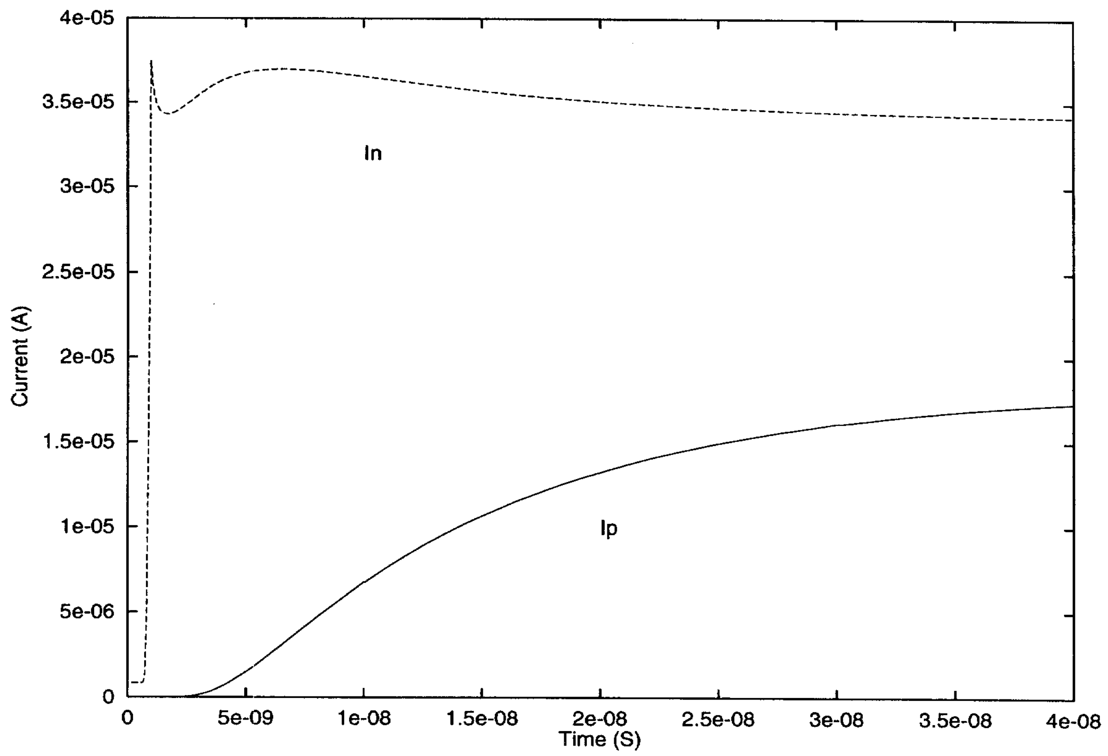
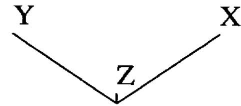
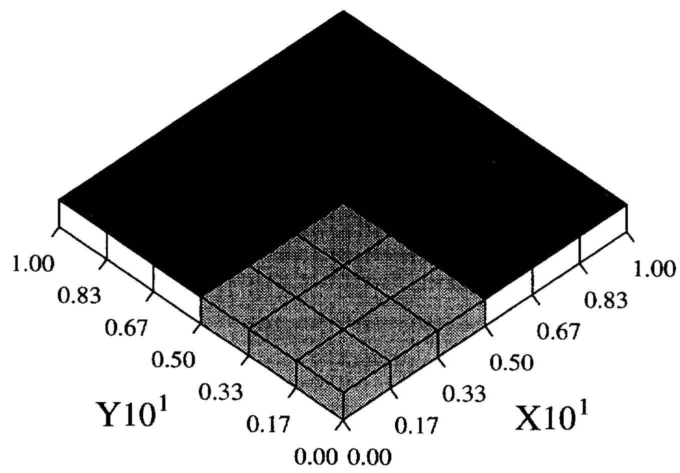
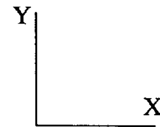



Figure 8: *Hole and electron currents from the transient simulation of switch on in a diode.*

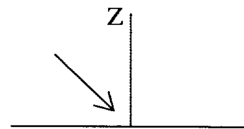



TRANSFORMED AXES




ANGLE OF ROTATION




ANGLE OF ELEVATION

Figure 9: Geometry of the refinement test structure, showing the two contacts (white), the oxide region (light grey) and the $n$ type silicon (dark grey).

current $J_n$. To accurately find the electric field within the device it would be appropriate to refine on potential, using the parameter MAXPSI. However in this example we choose to refine just on electron current, and compare terminal currents with those obtained using a set of finer uniform meshes.

**Input to the solver**

```
GEO REFTEST
MES REFTEST
DOP REFTEST
OUT REFTEST R
CAT REFTEST R
PHY REFTEST R
BIAS BOT (5)
BIAS TOP (0)
SOLVE ADAPTION=ON,REFILE=REFTEST.DBASE,MAXNOD=6000,
      MAXELS=6000,MAXTLS=4000,MAXDOP=0,LMINTOL=0.5,
      MAXPSI=0,RREFTOL=0.05,MAXJP=0
END
```

Setting MAXDOP, MAXPSI and MAXJP to zero prevents any refinement except on the electron current. MAXNOD and MAXELS are set to fairly large values for a simple 2D problem, so the mesh size can be limited by the two tolerances RREFTOL and LMINTOL. The former is a measure of the error in current between elements while the latter is a limit which prevents refinement of elements at scales much below $0.05\mu m$ in this case.

Figure 10 shows the electron current at one terminal plotted against the reciprocal of the number of nodes in the mesh. Two curves are shown, one using meshes generated with varying levels of RREFTOL and the other using purely uniform meshes where no automatic refinement has been used. Accurate terminal currents can be obtained with far fewer nodes using the current refinement.

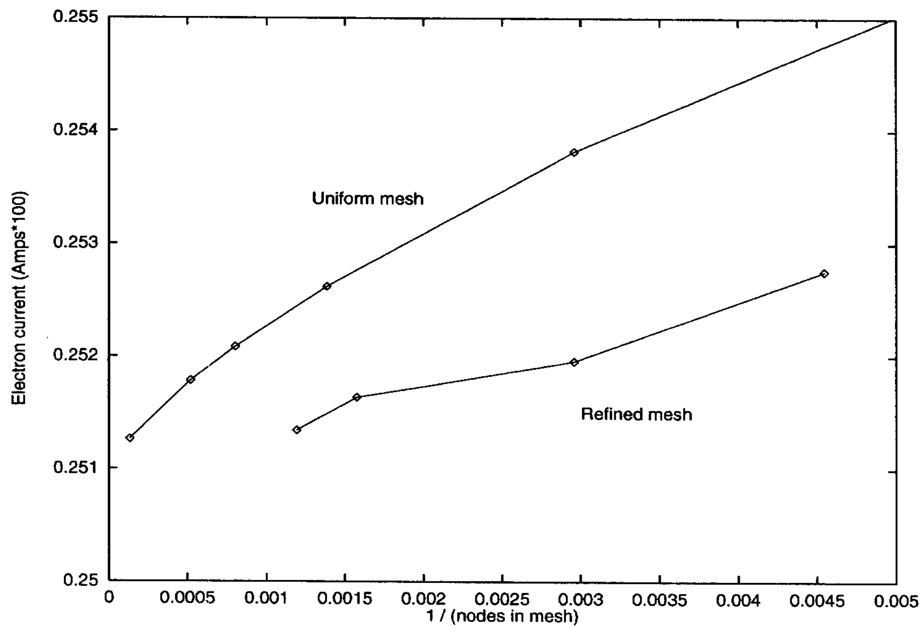



Figure 10: Electron current as a function of the reciprocal of the number of nodes in the mesh. The automatically refined mesh converges faster to the limiting value.

# References


[1] *Three dimensional algorithms for a robust and efficient semiconductor simulator with parameter extraction: the EVEREST final report*, C.Greenough, Report RAL92082, Rutherford Appleton Laboratory, Chilton (1992).

[2] *EVEREST Pre-processor user manual Version 3.0*, N.Ferguson *et al*, Report RLUR17908, Rutherford Appleton Laboratory, Chilton (1990).

[3] *EVEREST Doping generator Version 3.0*, G.A.Duffett and M.A.Towers, *et al*, Report RLUR17907, Rutherford Appleton Laboratory, Chilton (1990).

[4] *EVEREST Post-processor user manual Version 3.0*, P.A.Mawby *et al*, Report RLUR17909, Rutherford Appleton Laboratory, Chilton (1990).

[5] *Numerical analysis for semiconductor devices*, M.Kurata, Lexington Books, Massachusetts (1982).

[6] *Analysis and simulation of semiconductor devices*, S.Selberherr, Springer Verlag, Wein (1984).

[7] *Initial guess strategy and linear algebra techniques for a coupled 2D semiconductor equation solver*, S.P.Edwards, A.M.Howland and P.J.Mole, In NASECODE IV: Proceedings of the Fourth Intl. Conf. on the Numerical Analysis of Semiconductor Devices, (Ed. J.J.H.Miller), Boole Press, Dublin (1985).

[8] *Matrix computations*, G.H.Golub and C.F.Van Loan, Third edition, John Hopkins University press, Baltimore (1996).

[9] *Numerical initial value problems in ordinary differential equations*, C.W.Gear, Prentice Hall, New Jersey (1971).

[10] *Efficient implementation of a class of pre-conditioned iterative methods*, S.C.Eisenstat, SIAM J. Sci. & Stat. Comp., **2**, No. 1, 1-4, (1981).

[11] *RALBIC - A simple neutral file for finite element data.* , C.R.Emson, Report RAL 87102, Rutherford Appleton Laboratory, Chilton (1987).

[12] *Comprehensive testing of the EVEREST Solver module*, D. Gunasekera and C.Greenough, Report RULR17911, Rutherford Appleton Laboratory, Chilton (1989).

[13] *Transient Benchmark Tests*, D. Gunasekera and C.Greenough, Report RULR17910, Rutherford Appleton Laboratory, Chilton (1989).

# A Internal Commands

| | | |
|---|---|---|
| A.1 | MORE | to display the contents a file |
| A.2 | CHANGE | to change working directory |
| A.3 | RENAME | to rename a file |
| A.4 | COPY | to copy a file |
| A.5 | RM | to delete (remove) a file |
| A.6 | LIST | to provide directory listing |
| A.7 | WRITE | to provide monitoring of a session |
| A.8 | READ | to redirect the input stream to read from a file |
| A.9 | SYNTAX | to provide the syntax of a command |
| A.10 | HELP | to access HELP system |