Multidimensional Semiconductor Device and Micro-Scale Thermal Modeling Using the PROPHET Simulator with Dial-an-Operator framework

Anand L. Pardhanani[1], **Graham F. Carey**[1]

Abstract: Rapid prototyping tools that combine powerful numerics with a flexible applications interface can play a significant role in micro-scale modeling and simulation. We demonstrate this idea using the PROPHET simulator. In the first part of the investigations we extend the simulator's capability to allow analysis of carrier transport in deep submicron MOSFETs using a hydrodynamic model. The model is numerically implemented within PROPHET's dial-an-operator framework by adding certain "flux" routines. Once implemented, the model becomes available for use in any number of spatial dimensions. We present results for MOSFET type test problems in one and two dimensions. The second application area that we explore here using the same simulator is microthermal analysis. The implementation of the thermal equation in PROPHET is demonstrated and results are presented for a test problem in three dimensions.

keyword: device modeling, micro-scale, PROPHET, dial-an-operator.

1 Introduction

Aggressive scaling down of deep submicron MOSFETs has resulted in the development of more advanced physical models to account for the effects of larger local electric fields, higher substrate doping concentrations and very thin gate oxides [Dutton and Yu (1993)]. The ability to numerically simulate and study these models in a timely, cost-effective manner is crucial for their validation as well as for their practical use in device design. Thus, there is continuing need for software platforms that provide rapid prototyping capability, with robust numerical methods and a flexible interface for implementing new models [see Carey, Richardson, Reed, and Mulvaney (1996); Dutton, Kan, Yergeau, Yu, and Rafferty (1997)]. The emerging field of MEMS modeling and simulation has many similar requirements with regard to numerical techniques and software flexibility. MEMS applications generally involve coupled simulation domains on which different physical models apply, so the simulation software must be capable of handling multiple types of models and domains.

In the present work we utilize the PROPHET simulation platform developed at Bell Laboratories, Lucent Technologies, since it provides a powerful and flexible dial-an-operator framework for implementing new PDE models in different application areas. The simulator was originally developed for semiconductor process modeling, and has been extensively used in process development and diffusion research at Bell Laboratories. It was later expanded to serve as a simulation platform for more general PDE applications. The enhancements included more sophisticated data structures and management strategies, and a flexible interface for configuring or adding new PDE models into the simulator. Recent joint work between the University of Texas at Austin, Stanford University and Bell Laboratories has built upon these utilities and added a range of new device simulation capabilities into the simulator [Rafferty, Yu, Biegel, Ancona, Bude, and Dutton (1998); Rafferty, Yu, Pardhanani, Carey, and Dutton (1999)]. The PROPHET package may be licensed free of cost for research purposes by contacting Dr. C. S. Rafferty at Lucent [Rafferty (1998)].

Our focus in the first part of this study is on implementing and simulating the hydrodynamic class of models for MOSFET type devices. We discuss the basic strategy for implementing this model into the PROPHET simulator, and we present results for prototype MOSFET structures. The second application we consider here is a heat transfer model for predicting thermal transport in integrated "heat pipes" used in micro-scale devices. We simulate the model in 3D with anisotropic heat conductivity, and demonstrate how PROPHET can be rapidly configured to carry out this task.

The rest of this paper is organized as follows: Sec. 2 discusses the class of device models that we consider here; Sec. 3 describes the basic methodology for adding these models into PROPHET; Sec. 4 presents simulation results for MOSFET structures; Sec. 5 discusses the micro-scale heat transfer application and presents numerical results; and Sec. 6 offers some concluding remarks.

2 Device models

The basic device equations consist of the electrostatic potential equation, which is solved in conjunction with a set of transport equations for the charged carriers. The electrostatic potential equation is

$$\nabla \cdot (\varepsilon \nabla \psi) = q\,(n - p - N_D) \tag{1}$$

with ψ = potential, q = electron charge, ε = permittivity of silicon, n = electron concentration, p = hole concentration,

[1] TICAM, The University of Texas at Austin, USA.

and N_D = net doping concentration.

Our present emphasis is on the hydrodynamic class of models for transport of energetic carriers in MOSFETs. For clarity of exposition we focus on a representative non-parabolic hydrodynamic model developed at the Microelectronics Research Center at the University of Texas [see Bordelon, Wang, Maziar, and Tasch (1990, 1992)]. This model is considered physically appropriate, since it is based on a realistic, non-parabolic energy band structure. For device models based on electron transport it is mathematically described by the following set of coupled equations

$$\nabla \cdot (\boldsymbol{J}_n) = 0$$
$$\nabla \cdot (\boldsymbol{S}_n) + \boldsymbol{J}_n \cdot \nabla\psi = -\frac{n(w-w_0)}{\tau_w} \tag{2}$$

where $\boldsymbol{J}_n$ and $\boldsymbol{S}_n$ respectively denote the current density and energy flux, which are defined by the following equations

$$\boldsymbol{J}_n = \tau_p q \left[\frac{2}{3m^*}\nabla(B(w)nw) - \frac{q}{m^*}n\nabla\psi\right]$$
$$\boldsymbol{S}_n = \boldsymbol{Q} - \Omega w \boldsymbol{J}_n / q \tag{3}$$

This model introduces the average energy w as an additional variable in the transport formalism. The momentum and energy relaxation times, τ_p and τ_w, are empirical functions, which are chosen from the work of Bordelon, Wang, Maziar, and Tasch (1992)

$$\tau_p = \frac{0.007\,q}{w} \times 10^{-12} \text{ s}$$
$$\tau_w = 0.46 \times 10^{-12} \text{ s} \tag{4}$$

The system is closed by assuming a Fourier type constitutive relation for the heat flux, $\boldsymbol{Q}$, of the form

$$\boldsymbol{Q} = -\frac{2}{3}\left(\frac{\gamma R}{K_B}\right) n\nabla w \tag{5}$$

The other quantities in Eq. 2 - 5 are defined as follows: $B(w) = (1+\alpha\frac{w}{q})/(1+2\alpha\frac{w}{q})$, q = electron charge $= 1.602 \times 10^{-19}$ C, m^* = effective mass $= 2.367 \times 10^{-31}$ kg, ε = permittivity of silicon $= 11.9\,\varepsilon_0$, $\varepsilon_0 = 8.854 \times 10^{-12}$ $\text{C}^2/(\text{joule m})$, K_B = Boltzmann constant $= 1.381 \times 10^{-23}$ joule/kelvin, $w_0 = \frac{3}{2}K_B T_L$ joule, T_L = lattice temperature in kelvin, and $\alpha = 0.5eV^{-1}$, $\Omega = 1.3$, $\gamma = 4.2 \times 10^{-26}$ $(\text{watt m}^2)/\text{kelvin}$, $R = 0$ to 0.5 are empirical constants.

Note that in (2)-(3) it is also assumed that the drift kinetic energy is negligible, and the system reduces to the corresponding parabolic hydrodynamic model if we set $\alpha = 0$ and $\Omega = 3/2$.

Since the main issue is transport of electrons in the class of problems considered here, we can use either the quasi Fermi approximation or the drift-diffusion equations for modeling the transport of holes. The quasi Fermi approximation is given by

$$p = n_i \exp\left((\psi_p - \psi)/\psi_T\right) \tag{6}$$

where n_i is the intrinsic carrier concentration, ψ_p is the hole quasi Fermi potential, and $\psi_T = K_b T/q$. The drift-diffusion system for holes is

$$\nabla \cdot \boldsymbol{J}_p = 0 \tag{7}$$

with

$$\boldsymbol{J}_p = -\mu_p p \nabla\psi - D_p \nabla p \tag{8}$$

Here $\boldsymbol{J}_p$ = hole current density, μ_p = mobility and D_p = diffusivity.

Eq. 1-5, with the quasi Fermi approximation Eq. 6 or the drift-diffusion system Eq. 7-8, constitute the hydrodynamic system considered here for modeling charged electron transport in MOSFETs.

3 Implementation in PROPHET

Conceptually, the basic methodology used in PROPHET consists of decomposing equations into terms, and treating each term as a combination of a geometric and a physical operator. New application models can be implemented by either building the corresponding PDE system from a predefined set of geometric and physical operators, or by creating new physical operators via a well-defined interface. The framework also includes a database library which enables easy access to coefficients, parameter values or other properties that pertain to pre-configuring the supported applications. More comprehensive details regarding the set up of PDE systems and the structure of the database library are discussed in Rafferty (1996); Rafferty and Smith (1996).

PROPHET offers three different levels at which a user may interface with it to simulate new PDE applications. We classify these as follows: (1) input file level, (2) database level, and (3) flux routine level. The input file level allows for setting up the new PDE system and all the associated coefficients or parameters via a single input file, using PROPHET's command syntax. This approach can be used when the PDE system under consideration can be completely constructed using some combination of physical and geometric operators already built into PROPHET.

The database level involves hybrid implementation that uses an input file in conjunction with modifying and expanding the file that constitutes PROPHET's database library. This offers an alternative way of defining PDE systems that can be constructed using builtin operators. The advantage of using the database library to set up some or all of the PDE system is that the input file can then focus primarily on controlling specific design parameters, output quantities, and other run-time details that are of interest in the particular application.

Table 1 : Method of construction for each term in the hydrodynamic model: terms that require writing a new physical operator are denoted by "new" in column 3.

Eqn. #	Term	Physical operator		Geometric
		method	name	operator
(1)	$\nabla \cdot (\varepsilon \nabla \psi)$	builtin	lapflux	box_div
(1)	$q(p-n+N_D)$	builtin	potflux	nodal (or, volume)
(6)	$n_i \exp(-(\psi_n - \psi)/\psi_T)$, $n_i \exp((\psi_p - \psi)/\psi_T)$	builtin	quasiFermi	nodal (or, volume)
(7)	$\nabla \cdot \boldsymbol{J}_p$ with $\boldsymbol{J}_p$ in eqn. (8)	builtin	drift_diffusion	box_div
(2)	$\nabla \cdot \boldsymbol{J}_n$, $\nabla \cdot (\boldsymbol{S}_n + \boldsymbol{J}_n \psi)$ with $\boldsymbol{J}_n, \boldsymbol{S}_n$ in (3)	new	hydrodynamic	box_div
(2)	$-n(w-w_0)/\tau_w$	new	energyrel	nodal (or, volume)

The flux routine level involves programming of new physical operators, and using them in the database or input file to build new PDE systems. This level of usage becomes necessary when the PDE system under consideration cannot be fully constructed using the pre-defined operators. PROPHET provides a simple, organized interface for defining new operators, and a uniform, consistent way of accessing them from either the input file or the database library.

The PDE system that constitutes the hydrodynamic model involves several terms that can be constructed from the predefined operators. These include the gradient, divergence and Laplace terms, as well as terms that comprise the drift-diffusion equations, which were implemented in PROPHET as part of our previous joint work with Stanford University and Bell Laboratories. However, the hydrodynamic system also includes certain new terms that require creation and use of new operators, which must be done at the flux routine level. Thus, the full hydrodynamic model is implemented using a combination of pre-defined operators and new ones. We emphasize that the new operators needed here correspond to the "physical" part, as opposed to the "geometric" part, of the term. No new geometric operators are needed to implement the hydrodynamic model.

Tab. 1 lists all the terms in the hydrodynamic system and the other related equations given in Sec. 2. The "physical operator" heading indicates whether the term can be constructed via currently available builtin operators or whether a new operator is necessary. These cases are respectively denoted by "builtin" or "new" in the "method" column. We emphasize that each physical operator, whether new or builtin, basically serves as a prototype for constructing the general form of a term. Consequently, we always try when possible, to interpret or rewrite new terms into common forms, so that we can generate them with the fewest possible new operators and associated flux routines. Each "new" row in Tab. 1 groups together the terms that we construct using a single new physical operator. As the table shows, we define two new operators for use in conjunction with the existing device operators to implement the full hydrodynamic model.

The mechanism for programming new operators involves writing "flux" routines that take a standard argument list which provides nodal values of the dependent variables and their spatial derivatives. Using this information, the flux-routine computes the numerical value of the required operation at each of the given nodes. Any additional coefficients or parameters needed for constructing the function may be supplied to the routine via the database library or the input file.

To illustrate, let us consider a nodal operator such as `potflux` (see Tab. 1), whose construction is particularly straightforward, since it depends upon directly available nodal quantities. The primary ingredient in the flux routine for `potflux` is a loop that computes the flux and its derivatives with respect to each dependent variable at each node. It follows from Eq. 1 that, at any node, the flux operator and its derivatives are

$$f = -q(p-n+N_D), \quad \frac{df}{dn} = q, \quad \frac{df}{dp} = -q \tag{9}$$

The flux routine simply codes up these equations for f, df/dn and df/dp inside the loop over the nodes. The detailed syntax regarding argument lists and data structures is discussed in the references [Rafferty (1996); Rafferty and Smith (1996)].

The nodal operator `energyrel` in the hydrodynamic model is programmed in a very similar way to `potflux`, although it is also possible to construct it purely by superposing a series of existing arithmetic (nodal) operators. The main complexity in implementing the hydrodynamic model is in constructing the `hydrodynamic` operator. The terms defined by this operator model the current density and energy flux, which are the primary source of numerical instability and non-physical oscillations in simulation results. Before programming this operator it is necessary to develop suitable discretization techniques to circumvent the numerical instabilities. We use an extension of the Scharfetter-Gummel approach derived for the current-density and energy-flux terms in the present hydrodynamic system. Details of this derivation for the case of a general coordinate mapping are given in Pardhanani and Carey (1997). Here we state these results in the original coordinate system, and give the usual one-dimensional version that is used in a finite-volume setting to discretize the $\nabla\cdot\boldsymbol{J}_n$ and $\nabla\cdot\boldsymbol{S}_n$ terms. Accordingly, if we use subscript i to denote quantities at node i, the (constant) current-density and energy flux components on the mesh segment between nodes i and $i+1$ are given by

$$\frac{J_n}{q} = \frac{2}{3}\frac{C_\tau B(w_i^{av})}{m^*\Delta x}\left[\frac{n_{i+1}}{w_{i+1}}\beta(X) - \frac{n_i}{w_i}\beta(-X)\right]\frac{(w_{i+1}-w_i)}{\ln(w_{i+1}/w_i)} \quad (10)$$

$$S_n = \nu[w_{i+1}\beta(Y) - w_i\beta(-Y)] \quad (11)$$

where

$$X = \left[\frac{3}{2}\frac{q}{B(w_i^{av})}\frac{(\psi_{i+1}-\psi_i)}{(w_{i+1}-w_i)} - 2\right]\ln\left(w_{i+1}/w_i\right)$$

$$Y = \frac{\Omega J_n/q}{\nu}$$

$$\nu = \frac{H}{\Delta x}\ln\left(\frac{n_{i+1}}{n_i}\right)\frac{n_{i+1}n_i}{n_{i+1}-n_i}$$

and $\beta(x) = x/(e^x-1)$ denotes the Bernoulli function. The other quantities introduced in (10) - (11) are

$C_\tau =$ coefficient of $\tau_p = 0.007\times 10^{-12}$,

$\Delta x =$ local mesh spacing $= (x_{i+1}-x_i)$,

$w_i^{av} =$ average energy along edge $\Delta x = \frac{1}{2}(w_{i+1}+w_i)$

$H = -\frac{2}{3}\left(\frac{\gamma R}{K_B}\right) =$ coefficient of heat-flux term in Eq. 5

In practice it is also necessary to consider the limiting case where w is locally constant ($w_i = w_{i+1}$), since the current-density expression (10) breaks down for this case. Similarly, (11) breaks down when n is locally constant ($n_i = n_{i+1}$). For these cases, it can be shown that (10) and (11) reduce to

$$\frac{J_n}{q} = \frac{2}{3}\frac{C_\tau B(w_i)}{m^*\Delta x}\,[n_{i+1}\beta(X_l) - n_i\beta(-X_l)]$$

$$S_n = \frac{H}{\Delta x}n_i[w_{i+1}\beta(Y_l) - w_i\beta(-Y_l)] \quad (12)$$

with

$$X_l = \frac{3}{2}\frac{q}{B(w_i)}\left(\frac{\psi_{i+1}-\psi_i}{w_i}\right)$$

$$Y_l = \Omega\frac{J_n}{q}\frac{\Delta x}{Hn_i}$$

To implement the `hydrodynamic` operator in PROPHET we must write a flux routine that calculates J_n/q and S_n for each of the given mesh segments using Eq. 10 - 12. The main ingredients in the routine are loop structures with conditional statements to select the specific discretization formula based on local behavior of n and w. Due to the complexity of the terms in this flux operator we compute their derivatives with respect to the dependent variables using numerical differencing, instead of the usual practice of implementing analytical derivatives. The numerical differencing method is much easier to implement, but it incurs slightly higher computational cost.

Besides the energy flux term, the joule heating term ($\boldsymbol{J}_n\cdot\nabla\psi$) must also be implemented with care to ensure numerical stability. This is because the Scharfetter-Gummel formulation treats the components of $\boldsymbol{J}_n$ as constant along element edges, which makes their values discontinuous at the node locations. Thus the dot product, whose values are needed at the nodes, must be computed using averaging or some other consistent procedure. We follow the approach of using vector identities in conjunction with the carrier continuity equation to transform this term to the form of divergence of a "flux" as follows

$$\nabla\cdot(\boldsymbol{J}_n\psi) = \nabla\cdot(\boldsymbol{J}_n)\psi + \boldsymbol{J}_n\cdot\nabla\psi \quad (13)$$

Since the carrier continuity equation requires $\nabla\cdot\boldsymbol{J}_n = 0$, (13) reduces to $\nabla\cdot(\boldsymbol{J}_n\psi) = \boldsymbol{J}_n\cdot\nabla\psi$. Thus, the steady-state energy equation can be written as

$$\nabla\cdot(\boldsymbol{S}_n + \boldsymbol{J}_n\psi) = -\frac{n(w-w_0)}{\tau_w} \quad (14)$$

In our implementation in PROPHET we simply add the $\boldsymbol{J}_n\psi$ contribution to the energy flux $\boldsymbol{S}_n$ after calculating it as part of the new `hydrodynamic` operator. Thus this new operator constructs the entire left hand side inside the divergence operator in (14).

The right hand side of (14) is easy to construct, since it involves only nodal values. We do this via the new flux routine `energyrel`, which simply loops over the nodes list and computes the value of the right hand side function and its derivatives with respect to n and w.

After constructing the new operators in Tab. 1, they can be used in an input file to build the entire system of coupled equations that constitute the hydrodynamic model. For instance, in the MOSFET simulation studies that we consider in the next section, we use the hydrodynamic system to model electron transport and the drift-diffusion system to model holes. The equation system is constructed in the input file in PROPHET's scripting language using the following syntax

```
system name=hdnddp
+   sysvars=psi,n,w,p
+   nterm=8
+   term0=box_div.lapflux(psi|psi)@{silicon,oxide}
+   term1=volume.potflux(n,p,netdope|psi)
+          @{silicon}
+   term2=box_div.hydrodynamic(psi,n,w|n,w)
+          @{silicon}
+   term3=volume.energyrel(n,w|w)@{silicon}
+   term4=dirichlet.device_dirichlet(netdope|
+     psi,n,p)@{silicon/source,silicon/drain,
+     silicon/gate}
+   term5=dirichlet.default_dirichlet(0|w)
+          @{silicon/source,silicon/drain}
+   term6=constraint.continuity(psi|psi)
+          @{silicon/oxide,silicon/poly}
+   term7=box_div.drift_diffusion(psi,p|p)
+          @{silicon}
```

Here we have used the name `hdnddp` to label the system and to reference it later in the input file. The unknowns that we solve for are *psi*, n, w and p. The system of equations is constructed using 8 "terms" derived from the available operators. The general syntax for constructing a term has the form

```
  termi=a*geotype.phytype(i0,i1,i2,... |
        o0,o1,o2,...)@{material}
```

where a is an optional constant coefficient, "geotype" is the name of the geometric operator needed to construct the term, "phytype" is the name of the physical operator, and the parentheses enclose the list of input (or dependent) variables on the left and the list of equations that use this term on the right. The equations are identified with reference to the primary unknown that they model (the correspondence is unique because the number of unknowns must match the number of equations). The "material" field allows for the possibility of defining different equation systems to solve on different parts of the domain.

In the system definition shown above, `term0` and `term1` together define the electrostatic potential equation (Eq. 1) in Sec. 2; `term2` and `term3` set up the carrier continuity and the energy equations in Eq. 2; `term4` and `term5` define the Dirichlet boundary conditions and the portions of the boundary where they apply (all unspecified boundary conditions automatically default to vanishing normal flux at the boundary); `term6` is an internal constraint that is needed because we solve the Laplace equation for ψ inside the gate oxide region and couple this to the full hydrodynamic system via continuity of ψ at the oxide-semiconductor interface; `term7` sets up the carrier continuity equation for the holes using the drift-diffusion model.

The input file may also be used for specifying any coefficients or parameter values that the new flux routines may require, or for overriding any default values contained in the database library. In the present case we setup the coefficients for the relaxation time, and for the energy flux terms in this way. In addition, the input file is used to specify the problem's geometry, dimensions, doping profile, and the bias conditions at which the simulation is to be performed. We generally employ an incremental continuation strategy in applied bias to perform the simulations. In addition, continuation in model type is used at the start of the simulation, whereby the equilibrium solution is computed first, followed by drift-diffusion solution for both carriers, followed by hydrodynamic solution for electrons and drift-diffusion for holes.

One of the basic features of PROPHET is that all operators are valid for any choice of spatial dimensions. The new flux routines for implementing the hydrodynamic model follow this basic paradigm. Therefore, the equation system constructed using the syntax shown above may be used for simulation in any number of dimensions. In the next section we consider test cases in one and two dimensions, and present numerical results obtained with this hydrodynamic model.

4 Numerical results

Numerical studies using the hydrodynamic model implemented in PROPHET have been carried out on MOSFET type device structures in one and two dimensions. For the 1D tests we consider two different $n^+ - n - n^+$ diode structures with channel-lengths of 0.4μm and 0.08μm respectively. In both cases the doping in the n^+ regions is 10^{20} cm^{-3} and in the n region is 10^{15} cm^{-3}, with abrupt transition at the junctions. Fig. 1 and 2 show the computed solution for these cases with 4 volts applied at the drain (i.e., the right hand boundary). The number of grid nodes used in these calculations was 73 and 65 respectively for the 0.4μm and 0.08μm channel-lengths. As we would expect, the maximum energy is significantly higher for the case of short channel-length. This contributes to the "hot carrier" effects, whose impact becomes significant at short channel-lengths. The trend in the electron density plot in Fig. 2 is due to the fact that the electron velocity decreases from left to right in the channel region of this device (see, for example, Pardhanani and Carey (1994) for similar results using a different simulation package). The reason for this behavior may be that the present hydrodynamic model assumes the drift kinetic energy is negligible throughout the device. Fig .3 compares drain current-density versus drain voltage for the two test cases.

The two-dimensional MOSFET test structure has the geometry and doping profile shown in Fig. 4. The gate oxide region is 0.6μm long and 0.005μm thick. The simulation was carried out using a grid with 891 nodes, nonuniformly graded toward the source and drain junctions, and the channel region below the gate. As in the 1D case, we compute the equilibrium solution first and then employ incremental continuation in the voltage applied at the gate and the drain. Results are computed for

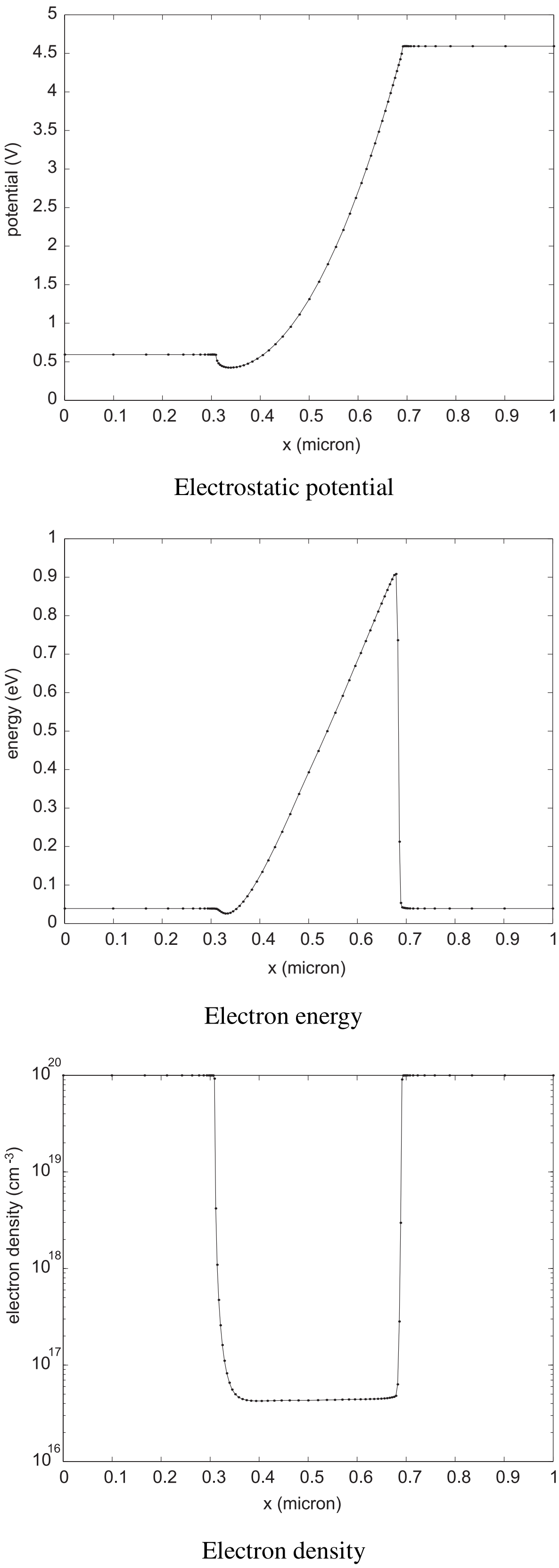

Figure 1 : Simulation results for 0.4μm channel-length $n^+ - n - n^+$ diode with 4 volts bias applied at drain.

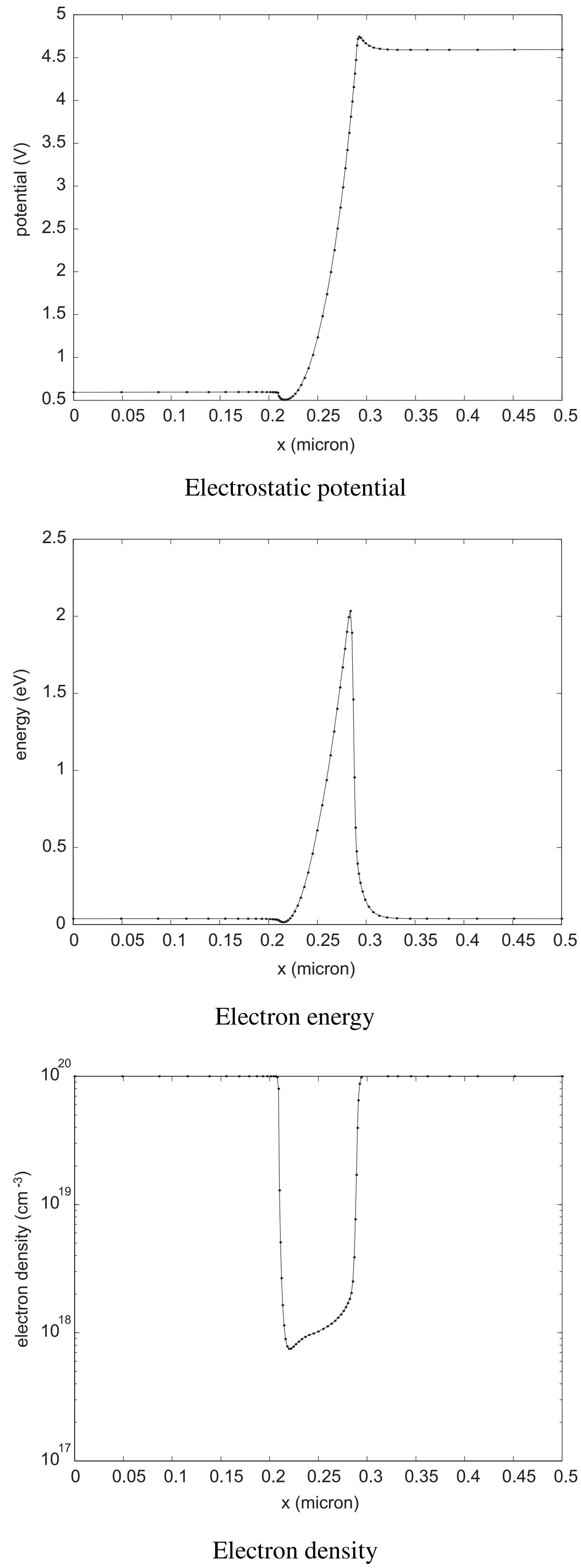

Figure 2 : Results for 0.08μm channel-length diode with 4 volts at drain.

gate and drain voltages ranging from 0 to 4 volts, and the corresponding drain voltage versus drain current characteristics are plotted in Fig. 5.

Numerical performance of these simulations once again confirms that the hydrodynamic system is considerably harder to solve than either the reduced (constant-energy) hydrodynamic system or the drift-diffusion equations. In the 2D case study shown here we found it necessary to restrict the incremental voltage steps at the drain to 0.02 - 0.04 volts to achieve convergence, which is more than an order of magnitude smaller than one can use with the drift-diffusion equations. Thus, each current-voltage curve in Fig. 5 represents 100 to 200 separate simulation points.

The computed solution at the strongest gate and drain bias is shown in Fig. 6.

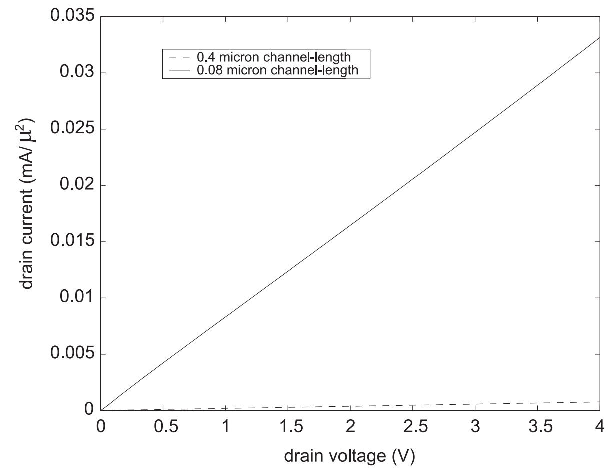

Figure 3 : Drain current-density versus voltage for diode structures.

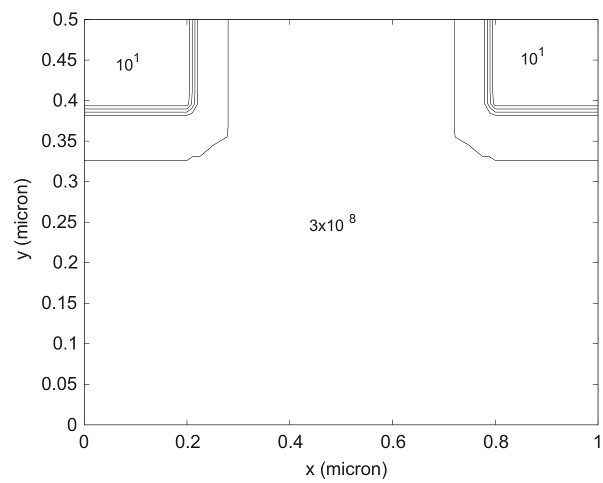

Figure 4 : Geometry and doping profile for 2D MOSFET test structure.

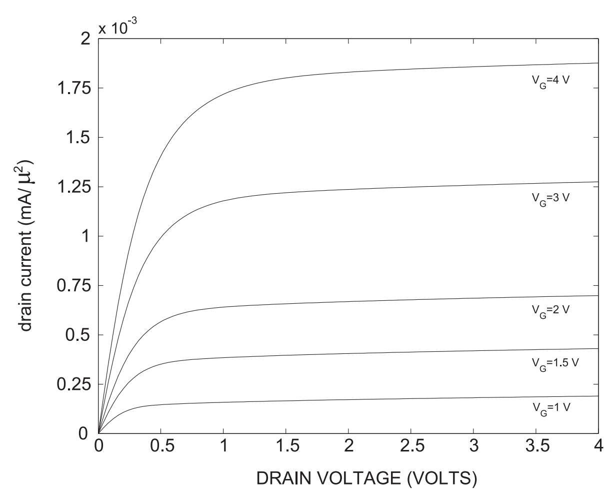

Figure 5 : Drain current-density versus drain voltage plots for 2D MOSFET at different gate voltages.

5 Modeling micro-scale heat pipes

The use of integrated arrays of micro pipes for thermal management has been proposed for certain micro-scale device types such as those that arise in MEMS applications [Mallik, Peterson, and Weichold (1992); Peterson, Duncan, and Weichold (1993)]. The ability to predict and control thermal transport through these heat pipes is important for their effective design and use. In this section we consider an application that involves the modeling of 3D anisotropic thermal transport (e.g., for cooling) in such devices.

A simple technique to model the effect of heat pipe arrays is to use an anisotropic heat conduction coefficient in a standard thermal transport model [see Huang, Liu, and Toh (1996)]. We use this approach in our preliminary work to extend the PROPHET simulator to this class of applications. The thermal transport equation that we use here is from Huang, Liu, and Toh (1996)

$$\rho C_p \frac{\partial T}{\partial t} - \left[k_1 \frac{\partial^2 T}{\partial x^2} + k_2 \frac{\partial^2 T}{\partial y^2} + k_3 \frac{\partial^2 T}{\partial z^2} \right] = \frac{qS(x,y,z)}{Q} \qquad (15)$$

where ρC_p represents density times specific heat of the material, T is the temperature, t is the time, x, y, z are the spatial coordinates, k_i (i=1, 2, 3) are the heat conductivities in the 3 coordinate directions, q is the heat load over a region described by the distribution function $S(x,y,z)$, and $Q = \int S(x,y,z)dxdydz$.

Fig. 7 shows a sketch of the device geometry, which is also the domain used in subsequent simulations. The material is assumed to be silicon and the heat pipes are assumed to be oriented in the y-direction, following the convention in Huang, Liu, and Toh (1996). This implies that the heat conductivity in the y-direction is much larger than that in the other two directions, so we use $k_2 >> k_1 = k_3$ in the numerical studies.

PROPHET does not provide a builtin operator for implementing anisotropic coefficients in a diffusion term. To provide this capability we wrote a new flux routine to construct the “physi-

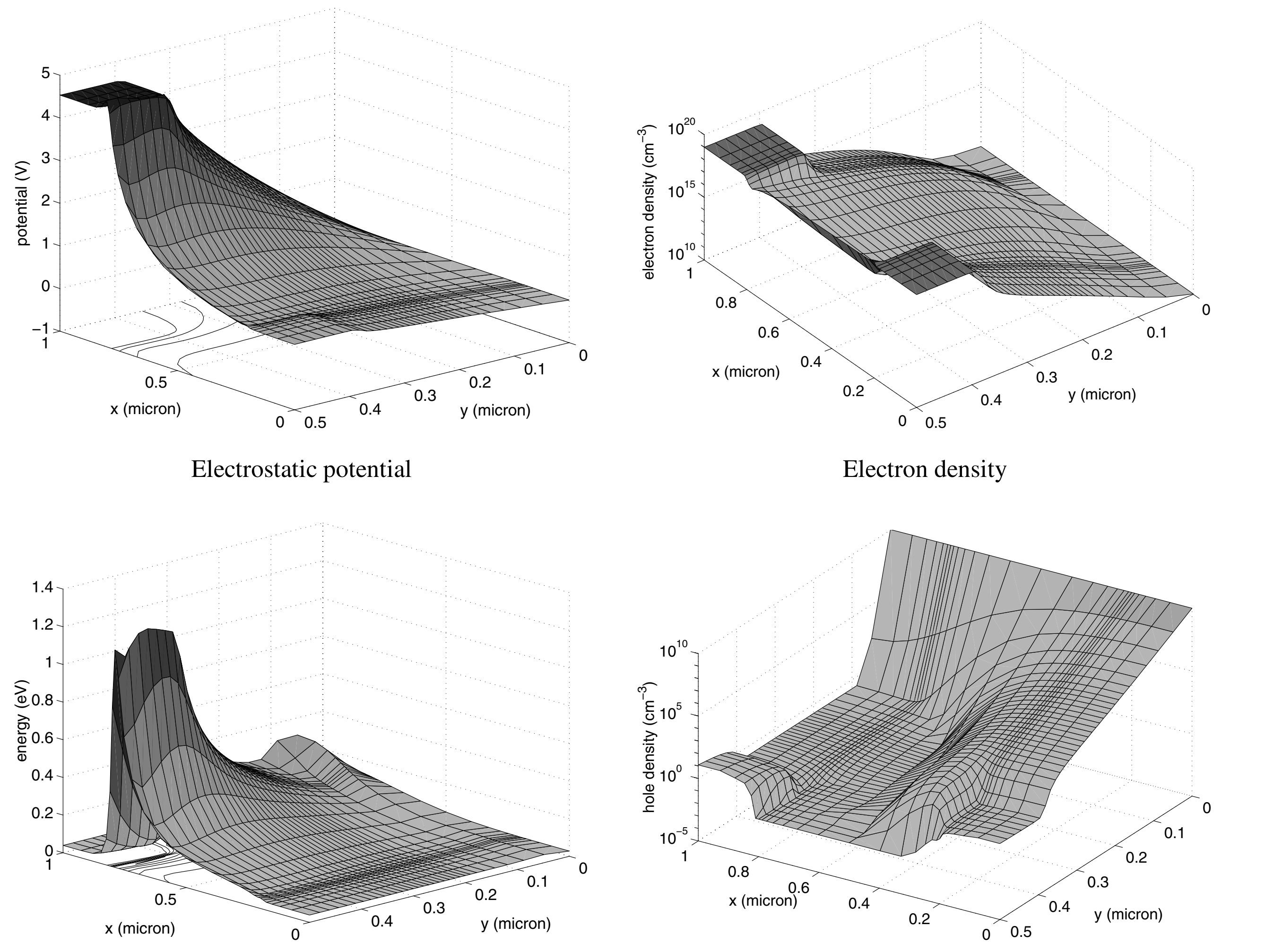

Figure 6 : Computed solution for 2D MOSFET with bias condition $V_G = V_D = 4$ volts.

cal" part of the operator, which corresponds to the dot product

$$\boldsymbol{k} \cdot \boldsymbol{\nabla} T \tag{16}$$

where $\boldsymbol{k} = [k_1, k_2, k_3]^T$. The key component of this routine involves computing a direction-dependent diffusion coefficient for each local mesh segment (element edge) using the given k_1, k_2 and k_3 along with the direction cosines of the mesh segment. Eq. 16 is constructed by combining such contributions from all the element edges that adjoin each node. This new physical operator when used with the standard `box_div` geometric operator (see Sec. 3) generates the desired heat conduction term with anisotropic coefficients.

The numerical studies are performed using the test case in Huang, Liu, and Toh (1996), which corresponds to the physical dimensions $a = b = 20$ mm and $c = 0.705$ mm. The heat source distribution function is assumed to be a planar square

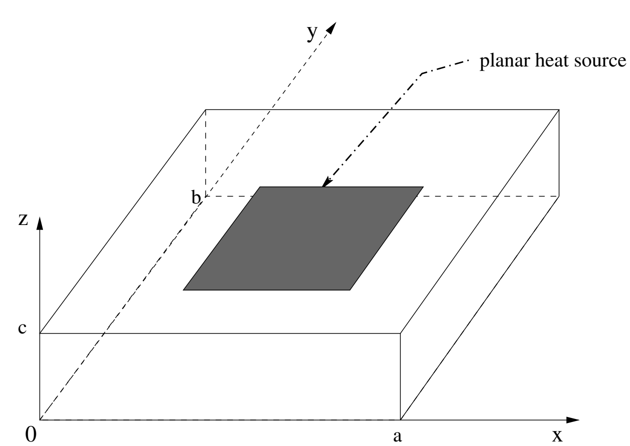

Figure 7 : Geometry of silicon wafer used in micro-scale thermal application. The dimensions used in our numerical simulations are $a = b = 20$ mm, $c = 0.705$ mm.

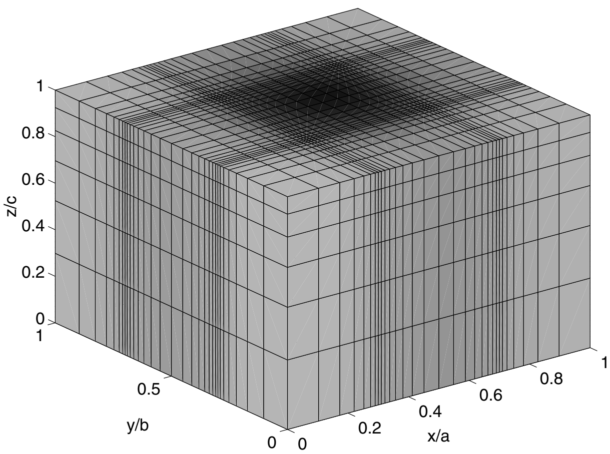

Figure 8 : Scaled temperature distribution on the surface of the domain at steady-state.

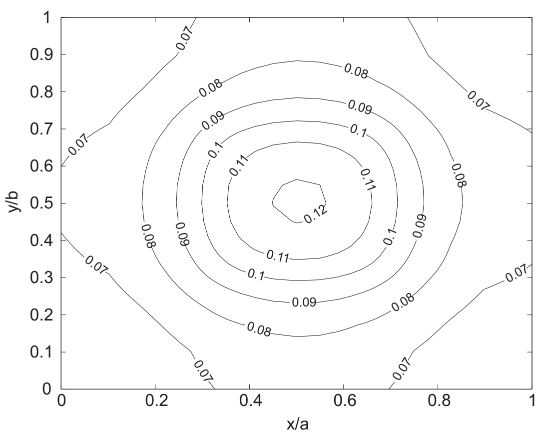

Figure 9 : Steady-state temperature contours on the bottom surface ($z = 0$).

on the top surface ($z = 0.705$), and is defined as

$$S(x,y,c) = \begin{cases} 1, & \text{for } 6 \leq x,y \leq 14 \\ 0, & \text{elsewhere} \end{cases} \tag{17}$$

The heat load amplitude factor q is 35 W, $k_1 = k_3 = 219W/mK$, and $k_2 = 10k_1$. We use a constant temperature initial condition and no-flux boundary conditions everywhere.

Fig.8 shows the steady-state (scaled) temperature distribution on the surface of the domain, and Fig. 9 shows a contour plot of the bottom surface ($z = 0$). These results agree well with those in Huang, Liu, and Toh (1996) in a qualitative sense, as well as in relative magnitudes. Direct quantitative comparisons have not been possible because the data given in the reference is not adequate to replicate their results.

6 Conclusion

The rapid prototyping capability of the PROPHET simulation platform has been extended to simulate a non-parabolic hydrodynamic model for MOSFETs in 2D. The discretization scheme includes special Scharfetter-Gummel formulations for the current density and energy flux terms in the hydrodynamic model. This increases the complexity of the implementation, but it is typically a necessary step to prevent numerical instabilities and to keep the grid size moderate. We have demonstrated how to implement this discretization scheme using the standard interface choices available in PROPHET. Simulation results have been presented for device structures in one- and two-dimensions.

We have also demonstrated the use of the PROPHET simulator to study 3D thermal transport due to heat pipes in microscale devices. The flux routine interface in PROPHET makes it straightforward to implement and simulate the effect of anisotropic heat conductivity, which is used to model the effect of the heat pipes. Preliminary numerical studies show good agreement with other published work.

The present study demonstrates the utility of a "dial an operator" type simulator for multiphysics applications, applied here in the context of microelectronic device simulation and microthermal analysis.

Acknowledgement: This research has been supported in part by the National Computational Science Alliance PACI Task # 791AT-51067A. We would like to thank Conor Rafferty at Bell Laboratories (Lucent Technologies), and Zhiping Yu and Bob Dutton at Stanford University for their help in various aspects of this work.

References

Bordelon, T. J.; Wang, X.-L.; Maziar, C. M.; Tasch, A. F. (1990): An efficient non-parabolic formulation of the hydrodynamic model for silicon device simulation. *IEDM Technical Digest*, pp. 353–356.

Bordelon, T. J.; Wang, X.-L.; Maziar, C. M.; Tasch, A. F. (1992): Accounting for bandstructure effects in the hydrodynamic model: a first-order approach for silicon device simulation. *Solid-State Electronics*, vol. 35, no. 2, pp. 131–139.

Carey, G. F.; Richardson, W. B.; Reed, C.; Mulvaney, B. (1996): *Circuit, Device and Process Simulation: Mathematical and Numerical Aspects*. John Wiley & Sons.

Dutton, R. W.; Kan, E. C.; Yergeau, D. W.; Yu, Z.; Rafferty, C. S. (1997): Next-generation tcad tools - supporting rapid prototyping of new models and numerics. In *NASA Device Modeling Workshop*. NASA.

Dutton, R. W.; Yu, Z. (1993): *IC Processes and Devices*. Kluwer, Boston.

Huang, X. Y.; Liu, C. Y.; Toh, K. C. (1996): A transient three-dimensional model of micro heat pipes used as an integral part of semiconductor devices. In *Microelectromechanical systems: Microscale thermal phenomena in electronic systems*, volume 59, pp. 37–47, ASME, New York.

Mallik, A. K.; Peterson, G. P.; Weichold, M. H. (1992): On the use of micro heat pipes as an integral part of semiconductor devices. *Journal of Electronic Packaging*, vol. 114, no. 4, pp. 436–442.

Pardhanani, A. L.; Carey, G. F. (1994): Adaptive grid and iterative techniques for submicron device simulation. In *Proceedings of the Third International Workshop on Computational Electronics*, pp. 248–251, Portland, OR.

Pardhanani, A. L.; Carey, G. F. (1997): A mapped scharfetter-gummel formulation for the efficient simulation of semiconductor device models. *IEEE Trans. CAD*, vol. 16, no. 10, pp. 1227–1233.

Peterson, G. P.; Duncan, A. B.; Weichold, M. H. (1993): Experimental investigation of micro heat pipes fabricated in silicon wafers. *Journal of Heat Transfer*, vol. 115, no. 3, pp. 751–756.

Rafferty, C.; Yu, Z.; Biegel, B.; Ancona, M.; Bude, J.; Dutton, R. (1998): Multi-dimensional quantum effect simulation using a density-gradient model and script-level programming techniques. In *SISPAD (Simulation of Semiconductor Processes and Devices) '98 Conference Proceedings*, pp. 137–140, Lueven, Belgium. Springer.

Rafferty, C. S. (1996): *Programmer's guide to the prophet database*. Technical memorandum, Bell Laboratories, Lucent Technologies.

Rafferty, C. S. (1998): Web guide to prophet. *http://www-tcad.stanford.edu/∼conor/prophet/guide.html*.

Rafferty, C. S.; Smith, R. K. (1996): *Solving partial differential equations with the prophet simulator*. Technical memorandum, Bell Laboratories, Lucent Technologies.

Rafferty, C. S.; Yu, Z.; Pardhanani, A. L.; Carey, G. F.; Dutton, R. W. (1999): *Semiconductor device simulation using prophet*. Ticam report, University of Texas at Austin, Austin, TX, (in preparation).