G. D. Hachtel
M. H. Mack
R. R. O'Brien

Semiconductor Analysis Using Finite Elements—Part II: IGFET and BJT Case Studies

Semiconductor-like Applications of Finite Elements (SAFE), a novel, nonlinear, general-purpose two-dimensional finite-element code, is applied to problems in device modeling. Case studies of contemporary insulated gate field effect transistor (IGFET) and bipolar junction transistor (BJT) structures are given which demonstrate the reliability, versatility, and efficiency of finite-element methods in general and of the SAFE program in particular. The user-defined SAFE physical model is compared with experiments on a doubly implanted short-channel IGFET. Computer experiments are performed, indicating how to select the type, distribution, and numerical-integration method of finite elements for maximally efficient, assured-convergence, engineering-accuracy analysis, either steady state or transient.

1. Introduction

We describe the solution of 2-D (two-dimensional) semiconductor device problems using SAFE [1], a program for Semiconductor-like Applications of Finite Elements. The key features of finite-element [1–15] (*vis-à-vis* finite-difference) methods [16, 17] for two-dimensional device analysis are ease of local grid refinement (but see [17, 18]), ability to choose basis functions which closely correspond to true solutions, and the smoothing properties of an integral, rather than differential, formulation. (See [11], Chapter 1 and its references.) However, complete and user-oriented exploitation of these properties requires imposing program development costs.

We employ as vehicles in this description finite-element case studies of insulated gate field effect transistor (IGFET) and bipolar junction transistor (BJT) structures. Thus we show how the user can specify (as in ASTAP [19] analysis of integrated circuit models) the number of equations to be solved and their analytic form (*i.e.*, type of nonlinearities) as well as the specific structure (*e.g.*, device dimensions and doping profile) to be analyzed. Two-dimensional solutions of the Poisson's (thermal equilibrium of BJT or IGFET), Poisson's and electron continuity (n-channel IGFET), and Poisson's and hole and electron continuity (BJT) equations are described. The program is so formulated that it would be a simple matter to do simultaneous thermal analysis [14, 20] by including the thermal continuity equation. This could be accomplished by specifying that four rather than three equations are to be solved, and giving a subroutine which defines thermal flux in terms of temperature and its gradients.

The reliability and efficiency of the SAFE program (see Ref. [1] for details of the computer implementation) are documented by a treatment of our basic numerical algorithm, which features 1) simultaneous (*i.e.*, fully implicit) solution of the user-specified PDEs; 2) a quadratically convergent Newton's method (appropriately modified for this application class); and 3) linear-equation solution by direct, state-of-the-art sparse-matrix techniques [21–23].

These are sensible, probably necessary choices for a general-purpose program. However, Buturla and Cottrell [5] have shown that specific cases exist for which fast convergence can be obtained solving the three semiconductor equations [*cf.* Eq. (4a)] sequentially, thus permitting certain efficiencies. We further address the critical question of how to select the type and distribution of the finite elements, as well as the type and accuracy of the basic numerical integration formula. Our goal in this

selection is maximally efficient, engineering-accuracy, assured-convergence analysis. Our basic Newton's method has been modified to handle the exponential nonlinearities typical of the semiconductor Poisson's and transport equations. We employ Newton step-limiting and prediction [24] for this purpose.

In Section 2 our report begins, for completeness, with a very brief introduction to the finite-element methods used in the SAFE program. The reader is referred to [11] for more detail about finite-element theory, and to [1] for the specific algorithms of the SAFE program. Section 3 describes the user specification of the IGFET and BJT modeling problems. In Section 4, the computed solutions of the 2-D device are compared to experiment.

In Section 5 we discuss the selection, specification, and refinement of finite-element grids. In Section 6 we summarize our experience to date on the efficiency-*versus*-accuracy tradeoff, and in Section 7 we document our particular modifications of Newton's method. Section 8 is devoted to the problem of transient analysis.

In Sections 6 through 8, the SAFE program, as applied to 2-D device analysis, is further compared with previous work [1-9, 13-17].

2. Finite-element analysis

The SAFE program applies to problems described by a set of $NSC \geq 1$ semiconductor-like PDEs of the form

$$\nabla \mathbf{F}(\mathbf{u}, \nabla \mathbf{u}, x, y) - \mathbf{c}(\mathbf{u}, \dot{\mathbf{u}}, x, y) = 0, \quad (x, y) \in \Omega , \tag{1a}$$

where Ω is the domain over which (1a) is to be satisfied subject to the boundary conditions

$$\mathbf{U}(x, y) = \mathbf{M}(x, y)\mathbf{u}(x, y) + \mathbf{N}(x, y)\nabla_N \mathbf{u}(x, y), \quad (x, y) \in \partial\Omega , \tag{1b}$$

where $\partial\Omega$ is the boundary of Ω and $\nabla_N \mathbf{u}$ is the component of $\nabla \mathbf{u}$ normal to $\partial\Omega$.

The SAFE program regards $\mathbf{F}$, $\mathbf{c}$, and $\mathbf{u}$ as sets with NSC members, *i.e.*,

$$\mathbf{u} = \begin{bmatrix} {}^1u \\ {}^2u \\ {}^{NSC}u \end{bmatrix}, \mathbf{c} = \begin{bmatrix} {}^1c \\ {}^2c \\ {}^{NSC}c \end{bmatrix}, \mathbf{F} = \begin{bmatrix} {}^1F \\ {}^2F \\ {}^{NSC}F \end{bmatrix}. \tag{1c}$$

Thus (1) constitutes NSC partial differential equations in NSC unknowns.

Note that each member ${}^\eta F$, $\eta = 1, 2, \cdots, NSC$, of the set $\mathbf{F}$ represents a generalized flux, *i.e.*, a flow vector in a 2-D Cartesian space, whereas ${}^\eta u$ and ${}^\eta c$ are scalars.

Finite-element methods gain their special characteristics from two key properties. First, they include subdivision of the domain Ω of the PDE into a union of subdomains Ω^l, $l = 1, 2, \cdots, NEL$ (number of elements), called "finite elements," *i.e.*,

$$\Omega = \bigcup_{l=1}^{NEL} \Omega^l . \tag{2a}$$

The Ω^l are polygons (usually triangles) with vertices called "nodes" of the approximation. Second, finite-element methods approximate the solution, $\mathbf{u}$, of (1) in terms of basis functions $\phi_n(x, y)$, $n = 1, 2, \cdots, NDOF$ (number of degrees of freedom), each of which are polynomials in each finite element Ω^l, *i.e.*,

$$\mathbf{u}(x, y) = \sum_{n=1}^{NDOF} \alpha_n \phi_n(x, y) . \tag{2b}$$

Usually the $\phi_n(x, y)$ are nonzero only on a small subset of the Ω^l. The coefficients α_n of this expansion are called "generalized coordinates," and each is uniquely associated with the unknown function $\mathbf{u}$, evaluated at a certain node of the approximation, *viz.*,

$$\mathbf{u}(x_n, y_n) = \alpha_n .$$

Sometimes more than one generalized coordinate is associated with a given node, and sometimes α_n stands for a spatial derivative or other function of $\mathbf{u}$, rather than for $\mathbf{u}$ itself. The generalized coordinates are determined by the Galerkin conditions

$$\begin{aligned} r_n(\alpha) &= \int_\Omega \phi_n(\nabla \cdot \mathbf{F} - \mathbf{c}) d\Omega \\ &= \int_\Omega \phi_n \mathbf{F} \cdot d\partial\Omega - \int_\Omega (\nabla\phi_n \cdot \mathbf{F} + \phi_n \mathbf{c}) d\Omega \\ &= 0, \qquad n = 1, 2, \cdots, NDOF . \end{aligned} \tag{2c}$$

As $NDOF$ becomes larger, the PDE residual $\nabla \cdot \mathbf{F} - \mathbf{c}$ is annihilated by (2c) in an increasingly larger function space. In view of the approximation (2b), (2c) represents $NDOF \times NSC$ nonlinear equations in $NDOF \times NSC$ unknowns, which can be expressed in the vector form

$$\mathbf{r}(\boldsymbol{\alpha}) = 0 . \tag{3a}$$

Note that each component of the arrays $\mathbf{r}$ and $\boldsymbol{\alpha}$ is thought of as an NSC-member set in the sense of (1c). The Galerkin equations (2c) are solved by the Newton iteration

$$\frac{\partial \mathbf{r}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \Delta\boldsymbol{\alpha} = -\mathbf{r}(\boldsymbol{\alpha}) , \tag{3b}$$

$$\boldsymbol{\alpha} = \boldsymbol{\alpha} + \Delta\boldsymbol{\alpha} . \tag{3c}$$

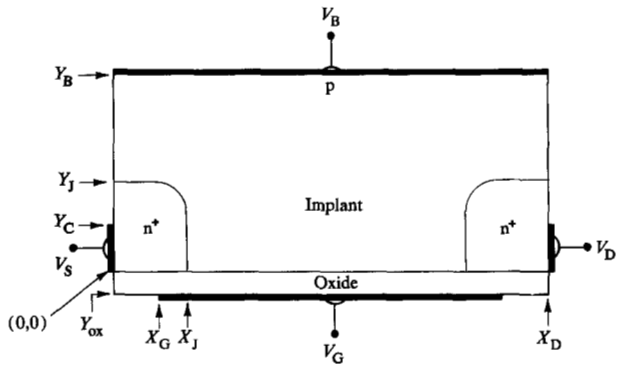

Figure 1 Ion-implanted IGFET structure.

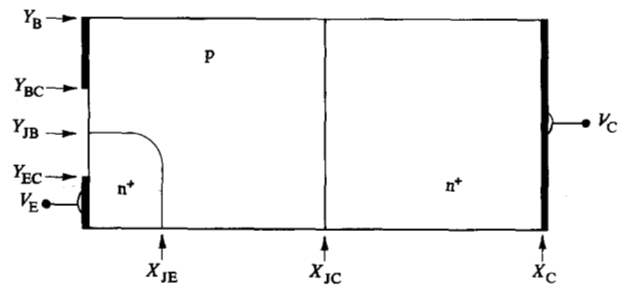

Figure 2 Buried-collector BJT structure.

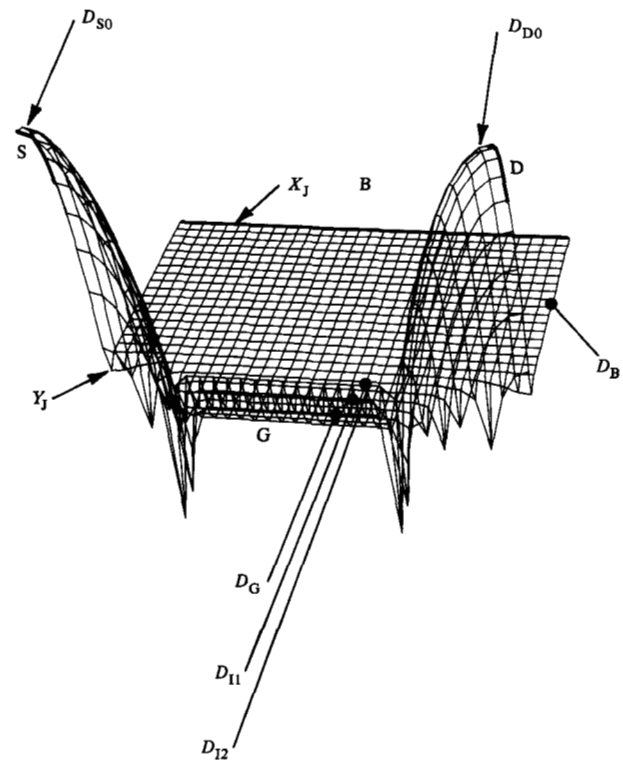

Figure 3 Log-doping double-implant IGFET. $D_{S0} = 4 \times 10^{19}$; $D_{D0} = 4 \times 10^{19}$; $D_B = 1.5 \times 10^{15}$; $D_G = -1 \times 10^{16}$; $D_{I1} = -4 \times 10^{16}$; $D_{I2} = -1 \times 10^{16}$; $Y_J = 1.7\ \mu$m.

The Jacobian $\partial \mathbf{r}/\partial \boldsymbol{\alpha}$ is a sparse matrix because the ϕ_n were assumed to have compact support, *i.e.*, to be nonzero over a small portion of the finite elements Ω^l.

3. IGFET and BJT modeling problems

The specific analytic forms of **F** and **c** are not built into the SAFE program but, as described in [1], are given by user-specified program modules GETF and GETC. For dc steady state ($\dot{\mathbf{u}} \equiv 0$) IGFET and BJT modeling problems we shall use the customary (see for example [1–10, 14–17, 25]) forms

$$\mathbf{u} = \begin{bmatrix} \psi_v \\ \phi_n \\ \phi_p \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} -q[p - n + D(x, y)] \\ -qR(p, n) \\ +qR(p, n) \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} \epsilon \nabla \psi_v \\ q\mu_n n \nabla \phi_n \\ q\mu_p p \nabla \phi_p \end{bmatrix} \tag{4a}$$

where ψ_v is the potential referenced to the valence band edge, $\phi_p(\phi_n)$ is the hole (electron) quasi-Fermi level, p, n, and D are the hole, electron, and ionized impurity densities where

$$n = N_C \exp\,[q(\psi_v - \phi_n - V_{GAP})/kT],$$

$$p = N_V \exp\,[q(\phi_p - \psi_v)/kT],$$

$$\mu_n = \mu_{n0}/(1 + \mu_{n0}\,|\nabla \phi_n|/V_{nL}),$$

$$\mu_p = \mu_{p0}/(1 + \mu_{z0}\,|\nabla \phi_p|/V_{pL}),$$

$$R(p, n) = (pn - n_i^2)/[\tau_p(n + n_T) + \tau_n(p + p_T)]. \tag{4b}$$

The remaining undefined quantities are appropriate physical constants taken from Sze [26].

The parameter *NSC*, which determines the number of equations and unknowns, is set by the user. For *NSC* = 1, SAFE solves the 2-D semiconductor Poisson's equations. If *NSC* = 2, the n-channel IGFET equations are used (add the electron continuity equation), and if *NSC* is set to 3, the BJT equations (also add the hole continuity equation) are solved. The IGFET and BJT device structures to be analyzed are shown in Figs. 1 and 2. The forms assumed for the doping profile $D(x, y)$ in these problems are illustrated by the perspective plots of Figs. 3 and 4.

Boundary conditions are user-specified by a program module GETBCO, which essentially gives **U**, **M**, and **N** for each given point (x, y), after testing to see if $(x, y) \in \partial\Omega$. Note that **U**, **M**, and **N** are *NSC*-member sets like **F**, **c**, and **u**. On the source contact of the IGFET structure of Figs. 1 and 3 we have specified the Dirichlet conditions

$$\mathbf{U}(x_S, y_S) = \begin{bmatrix} \psi_S \\ V_S \\ V_S \end{bmatrix}, \mathbf{M}(x_S, y_S) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{N}(x_S, y_S) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{5a}$$

where V_S is the voltage applied to the source (see Fig. 1) and ψ_S is given by the usual zero space charge condition

$$p - n + D(x_S, y_S) = 0\,, \tag{5b}$$

with p and n, specified by (4b), considered as functions of ψ_S. Similar expressions apply to the drain and substance contact regions of $\partial\Omega$.

On the part of $\partial\Omega$ not covered by metal contacts, we have specified the "natural" or "zero-flux" conditions

$$\mathbf{U} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{5c}$$

At the gate contact, we have the mixed conditions

$$\mathbf{U} = \begin{bmatrix} V_G + Q_{ss}T_{ox}/\epsilon_{ox} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{N} = \begin{bmatrix} -T_{ox}\epsilon_{si}/\epsilon_{ox} \\ 1 \\ 1 \end{bmatrix}, \tag{5d}$$

where $T_{ox} = |Y_{ox}|$ (Fig. 1), ϵ_{si} and ϵ_{ox} are the appropriate dielectric constants, and Q_{ss} is the equivalent charge "sheet density" localized at the interface between the IGFET channel and the gate oxide. Note that the top expressions in (5d) are easily derived from Gauss's law applied to the interface layer, and from the assumption of space-independent electric field in the oxide.

A similar treatment (except for the oxide) has been applied to boundary conditions for the BJT structure of Figs. 2 and 4.

4. Correlation of 2-D finite-element models to experiment

The usual method of adjusting the parameters of a two-dimensional IGFET program to match experimental data is to run a series of I_D calculations at drain and substrate voltages that are fixed and low (typically 0.1 V and 0 V) for values of V_G just above threshold. The source is grounded, *i.e.*, $V_S = 0$. In this region, the voltages are so low we expect to see no complications due to the mobility being affected by the field. By plotting I_D as a function of V_G we expect to obtain a straight line which can be extrapolated to find the threshold voltage of the program. This program threshold voltage will in general be offset from the device threshold voltage. The offset exists because the programs do not include the work-function difference between the gate metal and the semiconductor

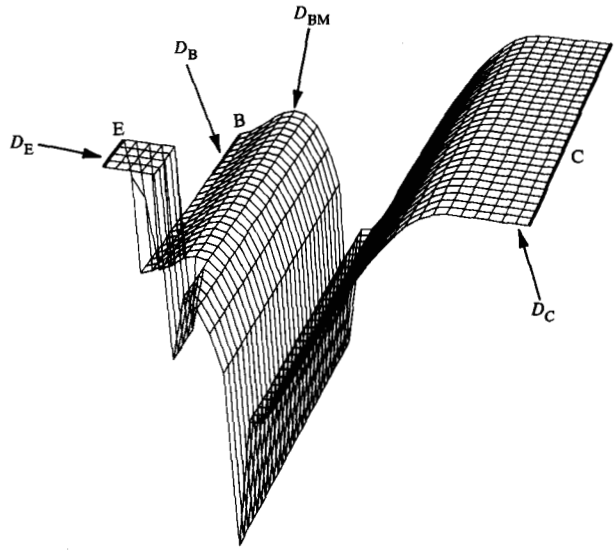

Figure 4 Log-doping BJT structure. $D_E = 1 \times 10^{20}$; $D_B = -1 \times 10^{18}$; $D_{BM} = -5 \times 10^{18}$; $D_C = 1 \times 10^{20}$.

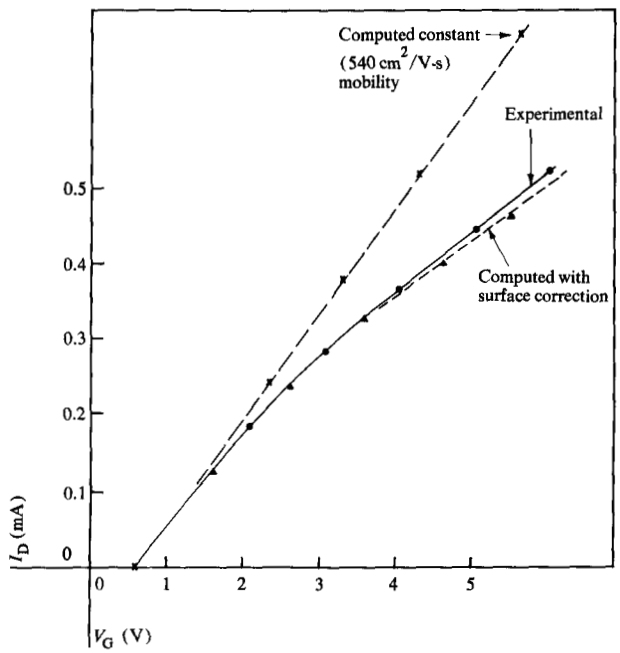

Figure 5 Theory *versus* experiment for first-order mobility model. $V_B = 0$ V; $V_D = 0.1$ V.

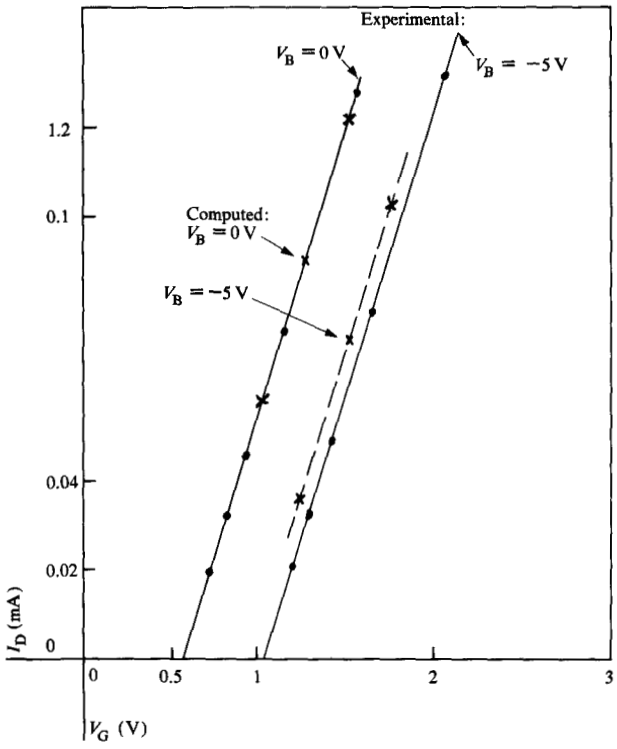

Figure 6 Correlation of effect of substrate voltage. $V_D = 0.1$ V.

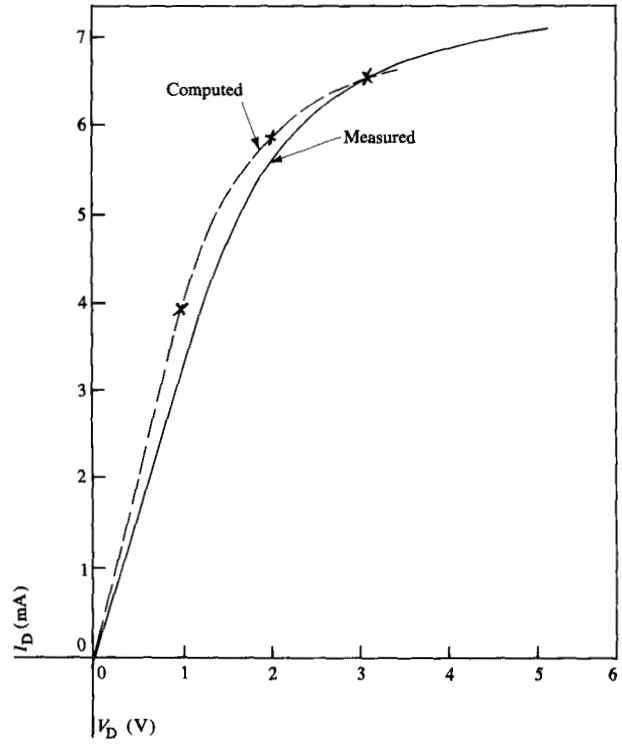

Figure 7 Comparison of computed and measured I_D *versus* V_D IGFET characteristics. $V_G = 4.55$ V; $V_B = 0$ V; $p_S = p_D = (n_i^2/2 \times 10^{20})$ cm^{-3}.

or the surface interface charge. The offset should be consistent with typical values of these quantities.

The results of efforts to match the SAFE program theory to an experimental IGFET characterized by the data of Figs. 1 and 3 are shown in Fig. 5. The threshold voltage of the calculated curves (crosses and triangles) was shifted from 2.1 V to 0.6 V to match the data. The justification for this shift is that the program does not include the work-function difference between the aluminum gate metal and the silicon, or the surface charge density. Also, the SAFE program uses the valence band in the semiconductor, instead of midgap, as its potential reference. This difference in reference level accounts for 0.54 V of the shift. The work-function difference could account for about 1.0 V. For example Sze ([26], p. 472) gives 0.9 V as the shift in threshold voltage for aluminum on 1×10^{16} uniformly doped n-type silicon with 50 nm oxide. A more detailed calculation using 40 nm oxide and the actual implanted profile will be necessary to determine the specific shift in the structure of Fig. 1 caused by work-function difference. Finally, a surface charge density of $Q_{ss}/q = 1 \times 10^{11}$/cm^2 [Eq. (3c)] would give a threshold shift of 0.18 V. Hence the shift of 1.5 V is not unreasonable, $1.54 + 0.18 \approx 1.7$.

The dashed curve (crosses in Fig. 5) corresponds to a constant μ_{n0} value [Eq. (4b)] of 540 cm^2/V-s. Although this value is much lower than the value of about 900 cm^2/V-s that is expected in a bulk sample doped at the level of $1\text{–}4 \times 10^{16}$ cm^{-3}, as in the channel region of Fig. 3, it is not inconsistent with the values of "effective mobility" commonly quoted in reports on IGFET analysis. However, the experimental curve bends over at high gate voltages, suggesting that the "effective mobility" decreases with increasing gate voltage [16].

Having established the I_D *versus* V_G transfer characteristics of the IGFET, we can now check whether our model displays the proper dependence of drain current on the drain and substrate voltages V_D and V_B. Figure 6 shows the I_D *versus* V_G characteristics for $V_D = +0.1$ V, and for $V_B = 0$ V (as in Fig. 5) and $V_B = -5$ V. Note that the computed currents are properly displaced to the right (crosses in Fig. 6) for $V_B = -5$ V, in agreement with experiment. Figure 7 shows the computed (crosses) and experimental I_D *versus* V_D characteristics for $V_B = 0$ V, $V_G = 4.55$ V. Again satisfactory agreement with experiment is obtained.

5. Finite-element grids and sample solutions

We shall refer to a particular decomposition, Eq. (2a), of the domain Ω of a given problem as a finite-element grid. Grid selection is a key aspect of finite-element analysis of semiconductor devices, just as it is for finite-difference

methods. In semiconductor device analysis it is generally necessary (for accuracy, not for stability) to have a finely spaced grid wherever extreme values or slopes of doping, space charge, current density, or electric field occur. IGFET analysis [5-9, 16] is especially difficult because of the extreme confinement of the conducting channel at high gate voltages. Channel widths of 2 nm are typical, whereas channel lengths and source junction depths are on the order of micrometers.

In the SAFE program, the user has two options in specifying the finite-element grid, as illustrated in Figs. 8-10. The first, and most convenient, option is to specify a "rectangular" grid of triangular finite elements. In this case the user specifies NX and NY, the number of x and y subdivisions, and two tables, TABLX and TABLY, giving $\Delta X(i)$, $i = 1, 2, \cdots, NX$ and $\Delta Y(j)$, $j = 1, 2, \cdots, NY$. Thus the domain Ω is subdivided into rectangular subregions, just as in finite-difference methods. The SAFE program then further subdivides each rectangular subregion into two triangles. Such "rectangular" finite-element grids for the FET structure of Figs. 1 and 3 and for the BJT structure of Figs. 2 and 4 are shown in Figs. 8 and 9, respectively, which are discussed below.

The second, and most efficient, user option in the SAFE program is to specify the triangular elements individually, giving for each element the names of nodal points its vertices will lie on. Then, the user specifies a list of the intended x and y coordinates of each nodal point. Triangulation is obtained automatically using the so-called Vornoi diagram, which is valid in the 3-D case (and even higher-dimensional cases) as well. (Briefly, the Vornoi diagram is a geometric construction for locating $n + 1$ points which are mutually closest in an n-dimensional Euclidian space.) In this case, arbitrary local refinement of any given subregion of Ω is possible. That is, the grid may be fine in regions of interest and coarse in regions such as charge neutral contact regions, where not much electronic action is taking place, thus permitting finite-element analysis with a minimum number of elements. Such a "special" finite-element grid, defined for the IGFET structure of Figs. 1 and 3, is shown in Fig. 10.

Note in the "rectangular" grid of Fig. 8 that the grid in the x-direction is uniform, whereas the spacing in the y-direction is so fine that the triangles become indistinguishable near the gate contact. Our numerical experiments have shown that such hyper-refinement of the grid in the IGFET channel is necessary for accurate current computation. However, the aspect ratio of the thinnest rectangles is (6.35 μm/16) to (0.005 μm), or about 800 to 1. Since ([11], Chapter 1) the error in finite-element approximation is in some cases inversely proportional to

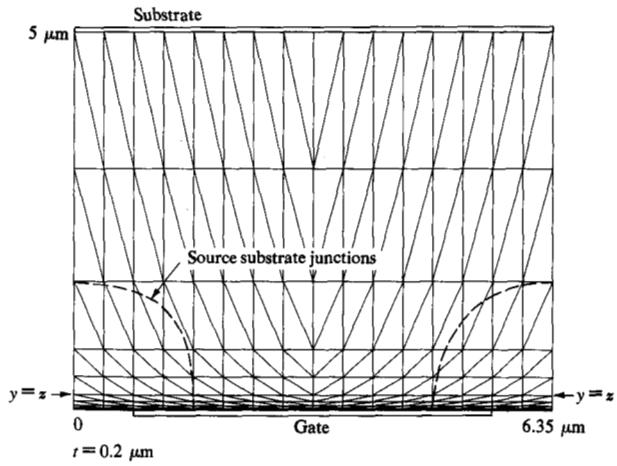

Figure 8 16 × 14 ($NX \times NY$) rectangular IGFET grid (entire 6.35-μm × 5-μm grid).

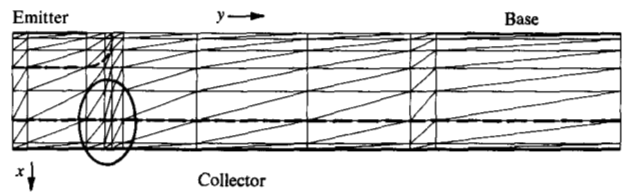

Figure 9 7 × 10 ($NX \times NY$) rectangular BJT grid.

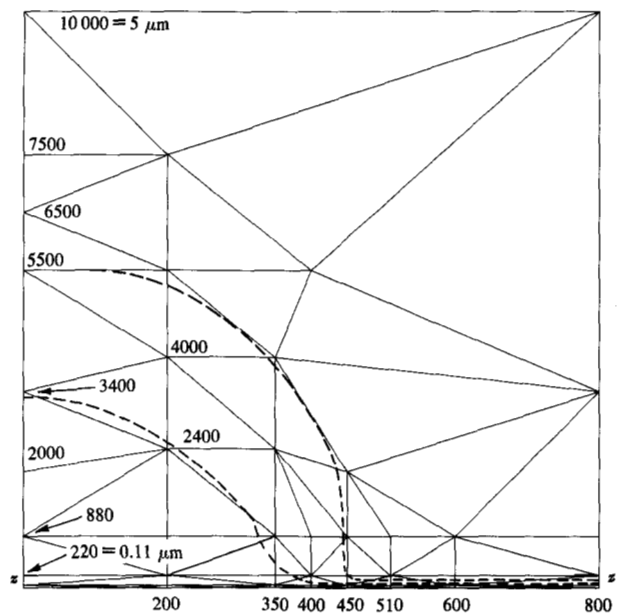

Figure 10 Special (locally refined) IGFET grid (entire half-grid).

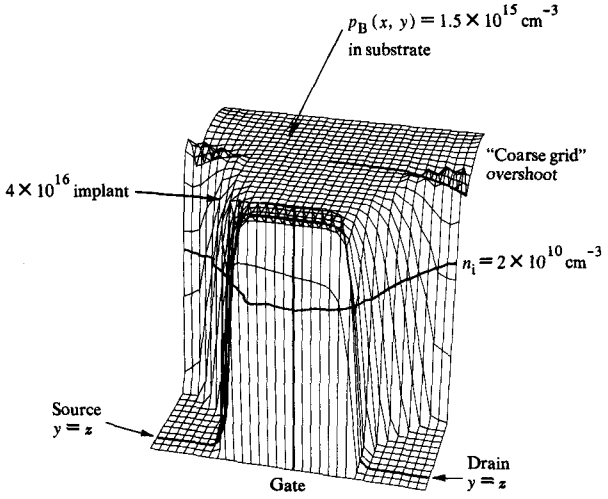

Figure 11 Log hole density for IGFET rectangular grid of Fig. 8: $V_S = V_B = 0$; $V_D = 0.1$ V; $V_G = 4.55$ V.

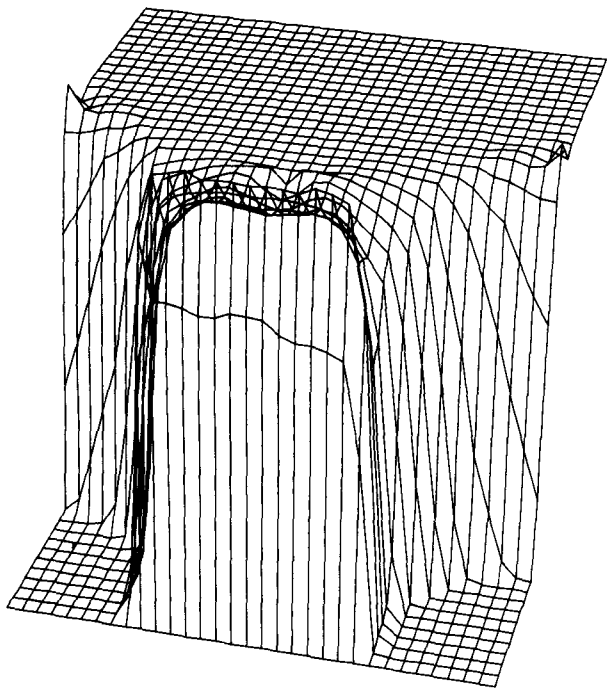

Figure 12 Log hole density for IGFET special grid.

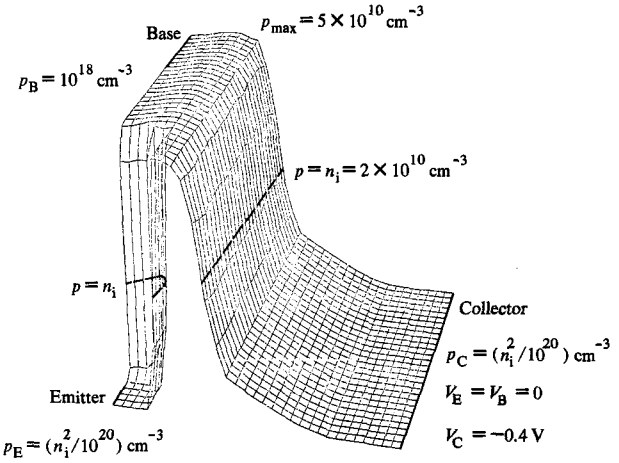

Figure 13 Log hole density for BJT.

the smallest angle in the approximation, the numerical results for such a grid must be looked at carefully. (See Section 6 for a discussion of such accuracy problems.) Also, note that there is no special refinement around the source-substrate and drain-substrate metallurgical junctions [dashed quarter circles in Fig. 8].

The "rectangular" 7 × 10 BJT grid of Fig. 9 is specially refined about the emitter-base junction (dashed quarter-oval in Fig. 9). The horizontal dashed line represents the collector-base junction. However, the limitations of the rectangular grid are apparent in the sub-rectangles (circled in Fig. 9) defined $\Delta Y_{3\text{-}7}$ and $\Delta X_{4\text{-}7}$. The refinement of the elements in these sub-rectangles is unnecessary, since the refinement in the y-direction was needed only to resolve the emitter-base sidewall injection.

The "special" IGFET grid [Fig. 10] illustrates an attempt to endow the finite-element grid with an IGFET personality. Figure 10 shows a half-grid with x-symmetry around the center line $x = 3.175$ μm. The nodes of the grid are located at integer coordinates IX, IY with $IX = 300$ corresponding to $x = 3.175$ μm, $IY = 10\,000$ to $y = 5$ μm. The dashed lines in Fig. 10 show the edges of the source-channel-drain to substrate space charge layer. This grid has 123 nodes and 207 elements, as compared with 255 nodes and 448 elements for the "rectangular" grid of Fig. 8. Nevertheless, the special grid shows essentially equal refinement in the channel region and greater refinement in the space charge layer delimited by the dashed lines. This economy is gained at the expense of a very coarse grid [see upper right of Fig. 10] in the charge neutral substrate region and, as will be discussed below, is well worth the price.

Figure 11 shows a computer perspective plot of $\log_{10} p(x, y)$, where p stands for the hole density of (4b). The flat corners of the bottom of the plot represent the essentially negligible hole density in the n-type charge-neutral source and drain contact regions. The dashed

bow-shaped curve in the middle shows the contour of the $p = n_i = 2 \times 10^{10}\ \text{cm}^{-3}$ line, *i.e.*, the substrate edge of the IGFET depletion region. The computation employed the "rectangular" grid of Fig. 8 and was for applied voltages above threshold of $V_S = V_B = 0$, $V_D = 0.1$, and $V_G = 4.55$. Note the extensive charge neutral substrate region where $p = p_B = 1.5 \times 10^{15}\ \text{cm}^{-3}$, except on the charge neutral substrate implant, where $p(x, y)$ follows the doping profile. It may be seen that the grid of Fig. 8, although quite accurate in the channel region, exhibits a "coarse grid" overshoot along the $x = 0$ and $x = 6.35$-μm lines. That is, there is inadequate resolution of the source-substrate and drain-substrate space charge layers. Figure 12 illustrates computed results for the same IGFET bias case as Fig. 11, but with the special finite-element grid of Fig. 10. The only difference between Figs. 11 and 12 is the choice of grid. Figure 11 shows an accurate representation of the holes in the physically important channel region. It has some "coarse grid" overshoot on the sides. Figure 12, computed using a grid with increased resolution on the sides, has less "coarse grid" overshoot at the cost of reduced accuracy in the channel.

Figure 13 shows a similar computed hole density plot for the BJT structure of Figs. 2 and 4 and the grid of Fig. 9. The bias condition was emitter- and base-grounded and the collector at 0.4 V.

Figure 13 shows some roughness in the emitter-base space charge region (dashed quarter-oval in Fig. 9) but, overall, compares nicely with the npn BJT doping profile of Fig. 4. Note the hump in the hole distribution corresponding directly to the out-diffused nature of the base diffusion, which shows that a substantial portion of the BJT base region is charge neutral.

6. Accuracy and computer resource requirements for finite elements

We attempt in this section to convey an overall picture of the cost-effectiveness of the finite-element method, *i.e.*, the quantity of computer resource required for analysis of a specified accuracy. To this end we have compiled in Table 1 a set of case study data for various grids and finite-element types, mainly for the numerically more difficult IGFET structure but also for the expensive (*NSC* = 3 for BJT *versus* 2 for IGFET analysis) BJT structure. After discussing the notation of Table 1, we present in Figs. 14 and 15 a reduction of data which explicitly shows the cost/accuracy tradeoff. Finally, in Fig. 16, we discuss the effect of the formula used for the numerical integration of (2c).

Some of the FET data in columns 10–14 in Table 1 are repeated, for ease of reference, from [1]. The case

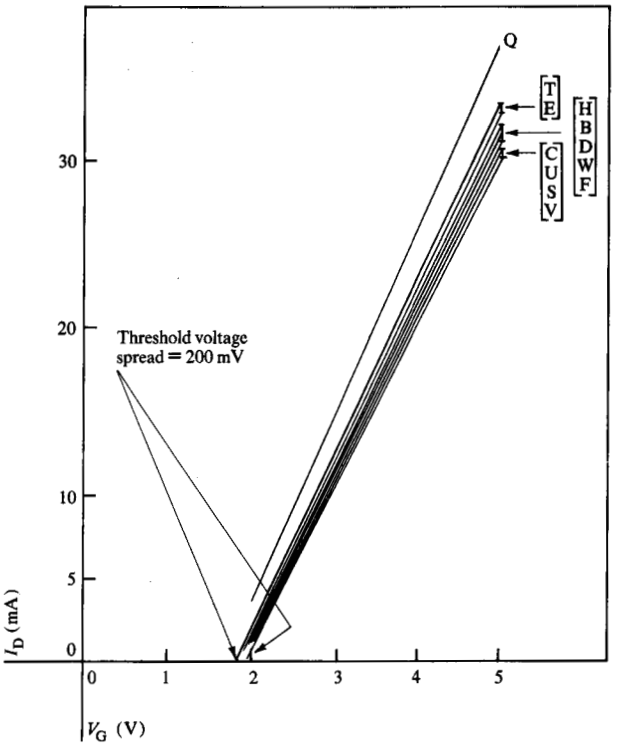

Figure 14 Computed I_D *versus* V_G curves.

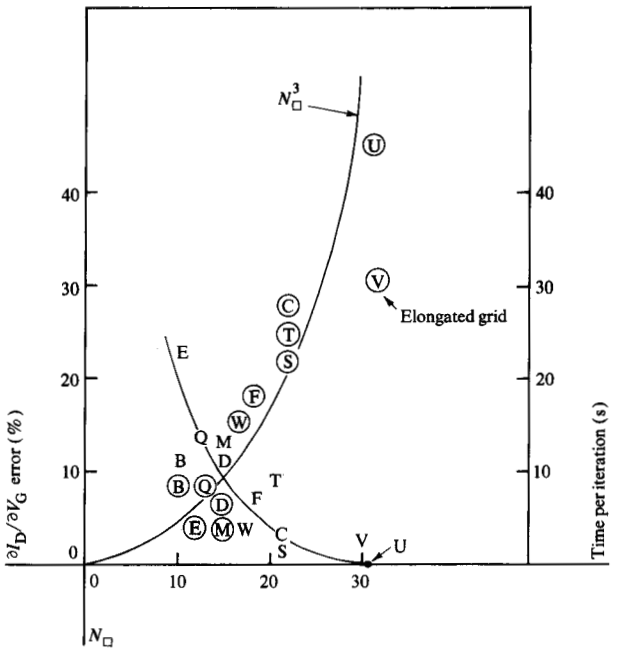

Figure 15 The accuracy/iteration-time tradeoff.

Table 1 Accuracy and computer resource data.

						Accuracy and time data[2]			*Sparse-matrix data*[3]				
IGFET	*Element type*	*NY × NX*	*NEL*	*Nodes*	*N*	I_{DM}^{1} (mA)	*Slope*[4] *error (%)*	*Iteration time* (s)	*NDOF*	*NZ, fills* (×1000)	*Mults* (×1000)	*Storage* (M-bytes)	*Setup time* (s)
U	quad	14 × 16	448	957	30	30.36	—	49	1914	38.2, 112	5068	1.2	458
S	quad	7 × 16	224	495	21	29.8	1.7	22	990	18.4, 30.3	856	0.6	50
V	quad	7 × 32	448	975	30	29.9	2.5	33	1950	37.8, 84.8	3015	1.2	186
C	lin	14 × 32	896	495	21	31.6	3.5	33.6	990	11.6, 32.5	738	0.4	54
W	quad	spec	152	339	17	31.5	6	24	678	12.9, 18.2	478	1.2	62
M	quad	7 × 8	112	255	15	32	6.1	14	510	8.7	245	0.28	14
E	quad	spec	170	377	18	31.7	6.4	15.5	754	14.6, 29.2	940	0.4	33
F	quad	14 × 8	224	493	21	32.6	7.1	22.1	986				49
D	lin	14 × 16	448	255	15	31.6	8.1	16.3	510	5.6, 11.7	225	0.4	14
B	lin	7 × 16	224	136	11	31.2	10.3	7.5	272	2.7, 3.3	49.1	0.4	7
Q	quad	4 × 4	76	179	12	36.8	13	6	358				8
E	lin	spec	152	94	12	33.9	21	5.1	188	1.9, 1.8	25.6	0.4	4
BJT													
BIPO	lin	10 × 20	400	231	20			53	693	12.3, 32.6	114.2	0.3	79
BIP	lin	spec	95	146	9.7			12.1	285	4.6, 3.5	72.8	0.3	54.1

[1]IGFET drain current for $V_S = V_B = 0$, $V_D = 0.1$, $V_G = 5$ V.
[2]Obtained for 19 integration points per triangle.
[3]For the SL-MATH [22] Program.
[4]Slope = $\partial I_D/\partial V_G$. Error is average of error at $V_G = 3.4$ and 5 V, $V_D = 0.1$, $V_B = V_S = 0$ (compared to Case U).

identification letters in the first column of Table 1 have the same meaning as in [1].

The *NY × NX* column of Table 1 contains the number of y and x subdivisions of the grid if "rectangular," but contains the notation "spec" if the finite-element grid is "special," *i.e.*, locally refined. The next three columns of data give the number of elements (*NEL*) and nodes (*Nodes*), plus the number of subdivisions, N, of an equivalent square grid, given by

$$N_{\square} = \frac{NEL \times NDEG^2}{2}, \tag{6}$$

where *NDEG* is the degree of the polynomial in a given element, *i.e.*, 1 for linear or 2 for quadratic elements. The factor 2 is present because the purpose of (6) is to provide comparison with well-known finite-element data [27] which have been compiled for rectangular, rather than triangular (two triangles = one rectangle), elements.

The seventh column in Table 1 gives I_{DM}, the drain current corresponding to $V_S = V_B = 0$, $V_D = 0.1$ and $V_G = 5.0$ V. Case U at the top of the column gives what we believe to be the "best" answer. Next to I_{DM} is the error (relative to Case U) in the slope (or transconductance) $\partial I_D/\partial V_G$ for $V_S = V_B = 0$, $V_D = 0.1$, but averaged over $V_G = 3$, 4, and 5 V.

The ninth column gives the time in seconds required for one pass through the Newton iteration (3), on an IBM System 370/168 operated under the VM (CMS) timesharing system. All data are for a 19-point integration formula [28] applied to the Galerkin integrals of (2c). That is, the integrands of (2c) were evaluated at 19 selected points inside each element and were summed with appropriate weights according to standard numerical integration formulae. As discussed below, the iteration times are disproportionally large where the element count is large, but to a good approximation the times are strongly dominated by the time required to solve the sparse system of *NDOF* linear equations (3b).

Solution of the sparse-matrix equations was carried out with the SL-MATH package [22]. The right-hand portion of Table 1 begins with *NDOF*, the rank of the Jacobian matrix $\partial \mathbf{r}/\partial \boldsymbol{\alpha}$. *NZ* [nonzero count in $\partial \mathbf{r}/\partial \boldsymbol{\alpha}$ of (3b)], *fills* [extra nonzeros created in Gauss elimination of $(\partial \mathbf{r}/\partial \boldsymbol{\alpha})$], and *Mults* (multiplication count) are tabulated at the right of *NDOF*. The numbers shown should be multiplied by 1000 to get actual counts. Thus Case U has a sparse (1914 × 1914) matrix with 32 200 nonzeros. The LU factors of $(\partial \mathbf{r}/\partial \boldsymbol{\alpha})$ have 38 200 + 112 000 = 150 200 nonzeros in Case U, representing a sparsity of 150 000/(1914 × 1914) × 100 = 4% in the Gaussian elimination. While still quite sparse, comparison with sparse-matrix data [19] for IC

circuit model simulation shows that finite-element analysis of device PDEs leads to substantially denser sparse matrices.

The storage required is quoted in megabytes. These data represent nominal dimensioning. The SL-MATH package would still work, but less efficiently, if less storage were provided [22].

The rightmost column gives the sparse-matrix preprocessing time for ordering and symbolic factorization of $\partial \mathbf{r}/\partial \boldsymbol{\alpha}$. When, as in Case U, the storage provision is inadequate, the preprocessing, or setup, time is disproportionally large, as was the case in the Newton iteration time discussed above.

To show more explicitly the cost/accuracy tradeoff, we present in Fig. 14 the computer I_D *versus* V_G curves for the cases described in Table 1, and have plotted in Fig. 15 the slope error of these curves and the Newton iteration time *versus* the equivalent size parameter $N_{\square}$.

Figure 14 shows that the top four cases in Table 1 produce I_D *versus* V_G curves which would be virtually indistinguishable if the proper translation adjustment were made (Section 4). The other cases, in good correlation to the relative coarseness of their finite-element grid, show various errors in either slope or displacement. Since, as stated in Section 4, the absolute translation along the V_G axis is not a critical parameter, we have chosen the slope or "transconductance" error, averaged over the 3-5 V interval, as the accuracy parameter in Table 1.

Figure 15 shows slope error (left ordinate scale, circled data points) plotted *versus* the equivalent size parameter $N_{\square}$. Also plotted (right ordinate scale) is Newton iteration time. The time data have been adjusted to remove the effect of excess time spent on numerical integration in cases where the element count is large (see discussion of Fig. 16 below). The time data are compared with a plot of $N_{\square}^3$ seconds. The comparison is favorable, since it has been shown [23] that the time required to solve the linear finite-element equations for a square grid must increase at least as fast as $KN_{\square}^3$ where K is a constant determined by the sparse-matrix method and the computing environment. Figure 16 shows the slope error decreasing with an approximately $N_{\square}^3$ dependence also, and suggests that engineering accuracy solutions (error <5%) require somewhat less than 20 s per Newton iteration in the SAFE program.

For comparison, Buturla and Cottrell [29] have reported an IBM System/370 Model 165 time of 2 s each for the

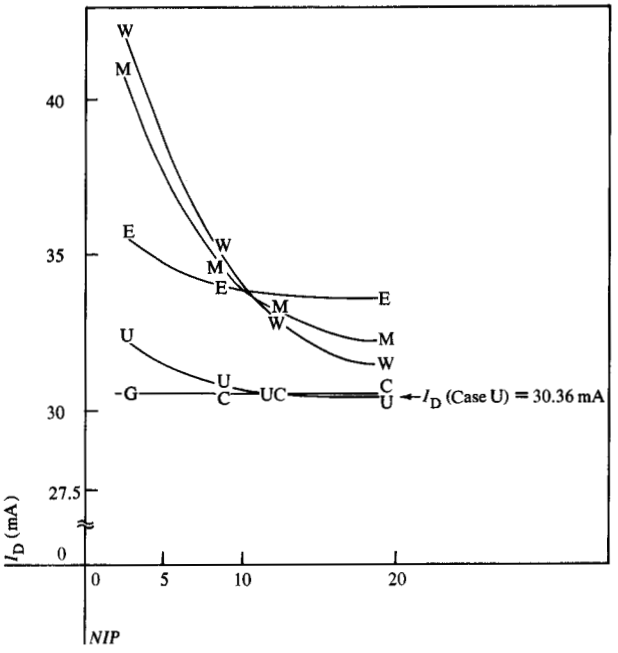

Figure 16 IGFET drain current *versus* number of numerical integration points. $V_S = V_B = 0$; $V_D = 0.1$; $V_G = 5$ V.

Poisson's and continuity equations. However, these equations are solved sequentially, rather than simultaneously, so several passes through each equation are equivalent to one simultaneous pass. Convergence in fewer than ten passes would be necessary for their program to surpass the SAFE program in execution speed. This comparison is possible since both programs employ the SL-MATH [22] sparse-matrix package. However, the SAFE program has the option of using the compiled-code GNSO package, which is two to four times faster than SL-MATH but requires twice the storage. Also, the SAFE program has the advantage of offering the option of quadratic as well as linear elements. As expected from the discussion of Section 6, the data of Table 1 show that for a given grid of nodes (compare Case D with Cases M and C with Case S) quadratic elements are more accurate.

Figure 16 shows the effect of the numerical integration formula [28] employed for the Galerkin integrals of (2c). The ordinate is the computed drain current for $V_S = V_B = 0$, $V_D = 0.1$, $V_G = 5.0$ V, and the abscissa is *NIP*, the number of points at which the integrands of (2c) are evaluated. Case U of Table 1 (448 quadratic elements) is regarded as the standard, and shows about a 10% error for *NIP* = 3, decreasing to the standard value for *NIP* = 19. Case C (896 linear elements) shows an almost constant error of about 1%, indicating that integration error is not

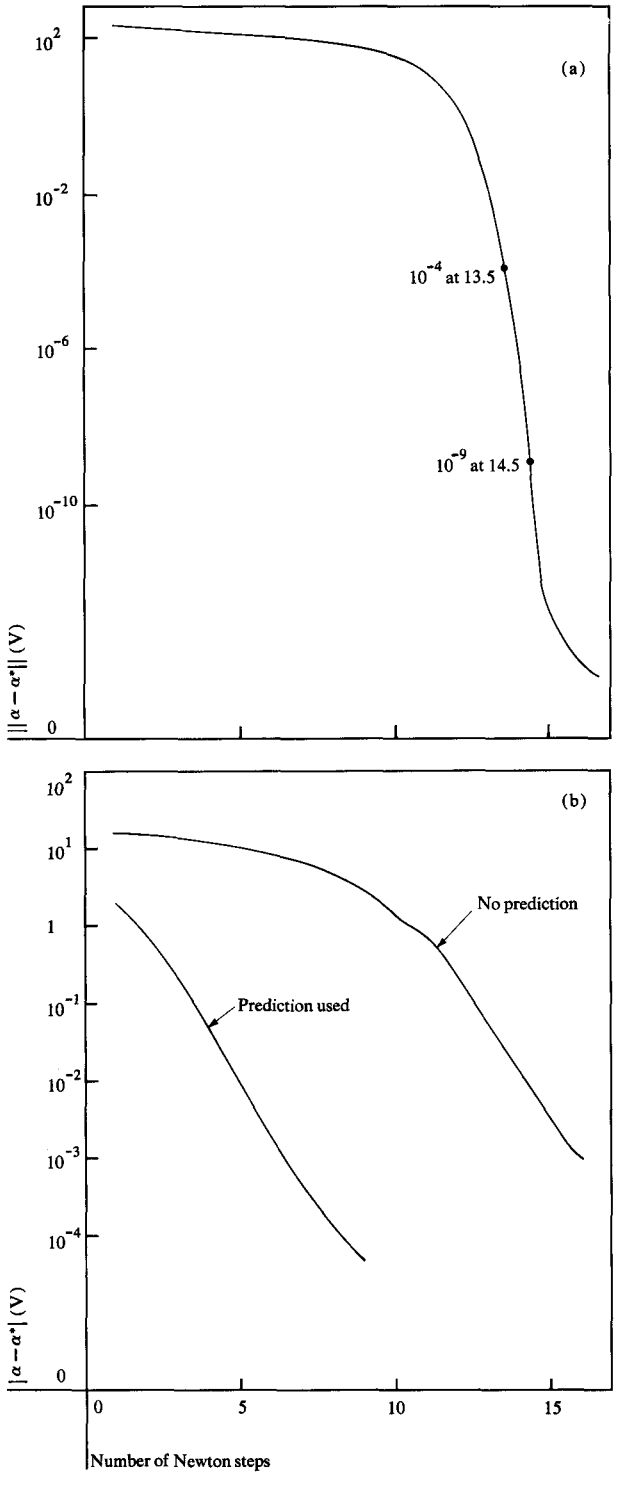

Figure 17 (a) Quadratic convergence of Newton iteration. (b) Effect of prediction on convergence of Newton iteration.

the dominant error in case C. Thus the 448 quadratic elements of case U, with $N = 31$, give greater accuracy but require more accurate integration than the linear case C. Figure 16 also shows that cases with fewer elements (quadratic cases M and W, linear case E) show more dependence than the accurate cases C and U.

7. The modified Newton iteration

The Newton iteration stated briefly in (3b) and (3c) converges quadratically. That is, if the solution to (3a) is $\boldsymbol{\alpha}^*$ [*i.e.*, if $\mathbf{r}(\boldsymbol{\alpha}^*) = 0$] and if $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|$ equals, say, 10^{-4} on the νth pass through (3b), we should have $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\| < 10^{-8}$ on the $(\nu + 1)$st pass if ν is sufficiently large. This is an important feature with regard to program reliability, because convergence is not only fast but definite. With the sequential method discussed above, not only is convergence linear, but it is sometimes difficult to decide whether convergence has occurred or not. However, quadratic convergence is ensured only if $\boldsymbol{\alpha}$ can somehow arrive within a certain neighborhood of $\boldsymbol{\alpha}^*$. The SAFE program has two safeguards to ensure this arrival. First, the full Newton step $\Delta\boldsymbol{\alpha} = -(\partial\mathbf{r}/\partial\boldsymbol{\alpha})^{-1}\mathbf{r}(\boldsymbol{\alpha})$ is not added as shown in (3c) but a scaled Newton step $\mathbf{D}\Delta\boldsymbol{\alpha}$ is added instead, where $\mathbf{D}$ is a diagonal matrix for which

$$D_{nn} = \begin{cases} 1, \text{ if } |\Delta\alpha_n| \leq DALIM_n,\ n = 1, 2, \cdots, NDOF, \\ DALIM_n \times SIGN(\Delta\alpha_n), \text{ if } \Delta\alpha_n > DALIM_n, \end{cases} \quad (7)$$

and the quantities $DALIM_n$ are user-specified. In the IGFET problem we have used $DALIM = 0.5$ for odd values of n (for odd n, the α_n correspond to the electron quasi-Fermi potential) and $DALIM_n = 2.0$ for even values of n (for even n, the α_n correspond to the electrostatic potential).

The second safeguard involves prediction of $\mathbf{u}(x, y)$ for a new bias condition based on converged results for previous bias conditions. For example, suppose an $I_D - V_D$ characteristic is being generated by the SAFE program and $\mathbf{u}(x, y)$ has already been computed for two values of V_D, say, $V_D = 0$ and $V_D = 0.001$ V. Then if u is desired, say, at 1 V, we set

$$u_1(x, y) = u_{0.001}(x, y) + \frac{u_{0.001} - u_0}{(0.001 - 0)}(1 - 0.001), \quad (8)$$

i.e., we perform linear extrapolation. As has been demonstrated in [24], such linear, or even quadratic (requiring three known values of $\mathbf{u}(x, y)$, prediction can reduce the total number of Newton iterations required in sweeping a dc *I-V* characteristic by a factor of 2 or more.

The quadratic convergence of the Newton iteration, as well as the role of the step limitation (*DALIM*) and prediction mechanisms, is illustrated in Fig. 17. Note in Fig. 17(a) that the convergence is slow and linear for the first ten Newton steps, because of the restraint of the *DALIM* parameters. The case shown ranges from 0 to 1 V on the

gate, with $V_S = V_B = V_D = 0$. At the eleventh iteration, α enters the neighborhood of $\alpha*$ and quadratic convergence commences. A noise level of $||\alpha - \alpha*||$ is reached around 10^{-14}, corresponding to the 12–13-decimal-place accuracy of a 370 machine in double precision. Figure 17(b) shows the advantageous effect of linear prediction for a higher bias case ($V_G = 5$, $V_B = 0$, $V_D = 2$, $V_S = 0$). Note that the curve for prediction is considerably below the curve for the no-prediction case. However, both curves show non-quadratic convergence due to the effects of the *DALIM* restriction.

8. Transient analysis

In this section we illustrate our treatment [1] of the transient case, using a first-order backward difference scheme in the time domain. The method is particularly compatible with our dc steady state program and requires only slightly increased storage. Since we solve the three equations for carriers and potential simultaneously by Newton's method, we achieve quadratic convergence for small changes in bias conditions.

The transient analysis mechanism, involving storage of the solution at the previous time step, has enabled us to also implement a linear prediction feature into the program. Use of this linear prediction as a starting guess for the Newton's iteration has enabled us to achieve a significant reduction in CPU time for some cases, namely a) accurate transient analysis, and b) the sweeping out of computed dc *I-V* characteristics.

The set c of scalar functions is computed in terms of the unknown potentials

$$\mathbf{u} = (\phi_p, \psi_v, \phi_n), \tag{9}$$

by a subroutine GETC (*cf.* [1], Section 4) which can be user-supplied or defaulted by the user to a built-in version. The transient analysis capability is achieved by providing in GETC for the computation of dp/dt and dn/dt. The default version of GETC uses the relations

$$\begin{aligned} dp/dt &= (\partial p/\partial \zeta_p) \cdot (d\phi_p/dt - d\psi_v/dt) \\ &\doteq (q/kT)\,(d\phi_p/dt - d\psi_v/dt)\,, \\ dn/dt &= (\partial p/\partial \zeta_n) \cdot (d\psi_v/dt - d\phi_n/dt) \\ &\doteq (q/kT)\,(d\psi_v/dt - d\phi_n/dt) \end{aligned} \tag{10}$$

for this purpose. Here ζ_p and ζ_n stand for the chemical potentials. If Fermi statistics are desired it is only necessary to redefine in GETC the functional dependence of p and n on the chemical potentials.

We handle the time differentiation in (4) by means of the first-order backward differentiation formula

$$d\phi_p/dt = (\phi_p - \phi_{p_{old}})/\Delta t\,, \tag{11}$$

and use similar relations for ψ_v and ϕ_n. This required provision for the storage of $\phi_{p_{old}}$, $\psi_{v_{old}}$, and $\phi_{n_{old}}$ at each point of the finite-element grid, and addition of the calculations implied by (9) and (10) into the default subroutine GETC. It is to be noted that the difference approximation (11) is fully implicit, *i.e.*, A-stable [24]. Consequently, the incorporation of transient analysis places no restriction on the maximum allowable size of the finite elements. Also, since with or without the time domain we solve the semiconductor finite-element equations simultaneously, we obtain quadratic Newton convergence for any size time step. In particular, with this formulation the dc steady state calculations become a special transient case for which $\Delta t \to \infty$.

The storage of the solutions at one backward time point gives us the opportunity to predict the solutions at the next time point. To do this we assume that the solution is varying linearly with time in the neighborhood of the current time step. We call the current time t_n, the new (*i.e.*, next to be computed) time step t_{n+1}, and the backward time step t_{n-1}. Due to our linearity assumption we may rewrite (8) as

$$\begin{aligned} &\phi_p(t_{n+1}) = \phi_p(t_n) + [\phi_p(t_n) - \phi_p(t_{n-1})]/\Delta t\,, \\ &\Delta t = t_n - t_{n-1}\,. \end{aligned} \tag{12}$$

As we show below, we have found that prediction is useful for accurate transient analysis and for sweeping out dc *I-V* characteristics. Of course, one must be careful not to use prediction during highly nonlinear portions of the transient response unless the time step is suitably small. If this rule is not followed, the exponential nonlinearities are likely to cause overflows and/or nonconvergence of the Newton's iteration.

Note that when sweeping out a dc *I-V* characteristic, the potentials in the vicinity of the ohmic contacts of typical semiconductor devices will vary linearly with the sweeping parameter, *i.e.*, "track" the applied biases. If the time variable is used for the sweeping parameter, the condition just stated for effective prediction will usually be met.

We have used a device like the simple device structure of Fig. 2 (except that the base contact is on the top horizontal surface) as a model problem for demonstrating the transient results. The structure is two-dimensional, and the base current I_B is flowing in the y-direction, *i.e.*, normal to the direction of flow of the emitter and collector currents I_E and I_C. The overall dimensions of this npn structure were 4.8 μm (length) by 0.025 μm (width). For simplicity, we have assumed a one-dimensional doping profile typical of bipolar technology. The emitter and

Table 2 Summary of numerical results.

t	qV_E/kT	I_E	I_C	I_B	ν_p/ν
0.00	28.00	1.68	1.68	0.00	—
0.25	28.25	1.97	1.74	0.23	4/5
0.50	28.50	2.22	1.86	0.35	4/4
0.75	28.75	2.47	2.00	0.45	4/4
1.00	29.00	2.68	2.14	0.55	3/4
1.25	29.00	2.50	2.23	0.27	4/5
1.50	29.00	2.49	2.30	0.18	4/5
∞	29.00	2.38	2.38	0.00	3/4

collector junctions were at $x_{JE} = 2.27$ μm and $x_{JC} = 3$ μm, and the peak base doping was $D_{BM} = 1.3 \times 10^{16}$ cm^{-3}.

Our numerical results are summarized in Table 2. The first column of Table 2 gives the simulation time in ns, and the second column gives the emitter voltage, normalized to units of kT/q, which is seen to be a truncated ramp function of time. V_B and V_C are set to 0, *i.e.*, grounded, throughout. The next three columns give the conventional terminal currents in mA. The last column shows the effect of the prediction mechanism on the required number of Newton iterations at each time step. Here ν_p stands for the number of Newton iterations with the help of prediction and ν for the number required without prediction.

Three aspects of the data are to be emphasized.

1. *Current conservation*
   The first and last rows of Table 2 correspond to dc steady states. Note that in these cases $I_E = I_C$, and I_B is small, indicative of current conservation in the presence of very low bulk recombination rates. Note that current conservation holds to within acceptable accuracy throughout the transient response, *i.e.*, $I_B = I_E - I_C$. Note that I_B peaks when the emitter voltage hits its up level and decays to zero at $t \to \infty$. This indicates that base current flows primarily to supply the extra hole density required to support the increased value of collector current, in keeping with well-established "charge-control" principles of device operation.

2. *Accuracy of transient response*
   Note that the currents are smooth functions of time. This reflects the fact that throughout most of the device the hole quasi-Fermi potential ϕ_p is rising linearly, essentially tracking the collector current. According to the depletion layer theory of pn junctions, this implies that the other potentials are changing smoothly as well. This suggests that the time steps are sufficiently small to ensure the accurate current computation. This has been verified by retaking the data of Table 2 using twice the time step, *i.e.*, $\Delta t = 0.5$ ns. The results (not shown) were essentially the same, which verifies the accuracy of Table 2. Note that in the last row of Table 2 the infinite time step was taken with no effect on the accuracy or stability of the numerical solution. This is the favorable result of choosing a fully implicit (A-stable) difference operator for the time derivatives. Similarly, it follows that at any point of a transient analysis, the dc steady state may be reached in effectively one more time step. Also, for problems which have convergence difficulties, the transient analysis mechanism can be incorporated into a "continuations" method of dc solution [24], which converts the given problem into a sequence of easier subproblems.

3. *Effect of prediction*
   Note in the last column of Table 2 that the prediction mechanism has had a significant effect on the number of Newton iterations required for the transient analysis. Summing the numbers in this column leads to the conclusion that since 26 iterations were required with prediction, and 31 without, about a 25% improvement is due to the prediction mechanism.

9. Conclusions

We have described the application of the SAFE finite-element program to field effect and bipolar transistor modeling problems. We have studied the problem of numerical integration inside the finite elements and have concluded that although relatively accurate numerical integration (*e.g.*, up to 19 point formulae) is sometimes required, the integration time is almost always outweighed in the overall computation by the sparse-matrix code for the solution of the linear Newton equations.

We have shown that a properly modified Newton's method, along with adequate grid refinement and an appropriate initial guess, reliably provides definite and quadratic convergence of the nonlinear iteration (usually, 20–30 iterations or less will suffice).

The physical model built into the SAFE program can be regarded as a default option for user-specified physics. The default physical model, coupled with appropriate specification of the finite-element grid, constitutes the SAFE device model. This model has been satisfactorily correlated with experiment for the short-channel IGFET device structure of Fig. 1. That is, terminal characteristics (I_D *versus* V_G, V_D, and V_B) are satisfactorily predicted, including voltage ranges above as well as below threshold.

The tradeoff between accuracy and computer resource requirements has been studied for two alternative sparse-matrix implementations, *i.e.*, 1) SL-MATH and 2) the compiled-code approach. Our main conclusion is that, although the full cost-effectiveness potential for the finite-element approach has not yet been realized, finite elements offer a major improvement over comparable finite-difference methods. We believe (*cf.* Fig. 5 of Ref. 1) we have achieved near-optimal efficiency when the compiled-code approach is used in a large dedicated partition of core. We believe, however, that storage requirements, which are substantial in our current sparse-matrix implementation, can be significantly reduced.

Implementation and testing of a first-order backward difference operator for time differentiation and prediction have been discussed. Higher-order methods, *cf.* [1], are sometimes advantageous for these purposes. The implementation would be identical for higher-order differentiation and prediction, except that it would then be necessary to store more than one backward time value of the finite-element solutions.

We believe several avenues of profitable future work offer themselves. The main memory requirement of the sparse-matrix code could be substantially improved without increasing computer time requirements. The generalized element method of B. Speelpenning [30] offers hope for such an improvement. Other possibilities are described in [31]. Finally, there is much work that could be done toward practical automatic grid selection, coarsening or refinement, either dynamically or by implementation of computer graphics aids, or both. It would be useful to develop an automatic scheme which evaluates the Jacobian only when necessary to retain quadratic convergence [31].

Acknowledgments

The authors are indebted to R. K. Brayton, F. G. Gustavson, H. H. Heilmeir, H. I. Stoller, and R. C. Joy for ideas and encouragement, to E. M. Buturla, P. P. Peressini, A. Phillips, and H. F. Quinn for discussions, and to B. Speelpenning for consultation and program organization. They thank A. Appel, T. R. Puzak, and A. Stein for their assistance with graphics. The constructive comments of the reviewers were especially appreciated.

References

1. G. D. Hachtel, M. H. Mack, R. R. O'Brien, and B. Speelpenning, "Semiconductor Analysis Using Finite Elements—Part I: Computational Aspects," *IBM J. Res. Develop.* **25,** 232–245 (1981, this issue).
2. J. J. Barnes and R. J. Lomax, "Finite Element Methods in Semiconductor Device Simulation," *IEEE Trans. Electron Devices* **ED-24,** 1082–1089 (1977).
3. R. J. Lomax, "Preservation of the Conservation Properties of the Finite Element Method Under Local Grid Refinement," *Comp. Meth. Appl. Mech. Eng.* **12,** 309–314 (1977).
4. R. J. Lomax, "Application of the Finite Element Method to Semiconductor Modeling," *Tech. Report No. UM-EPL-014289-T1, NTIS Accession No. PB287729/as*, Electron Physics Laboratory, University of Michigan, Ann Arbor, 1978.
5. P. E. Cottrell and E. M. Buturla, "Two-dimensional Static and Transient Simulation of Mobile Carrier Transport in a Semiconductor," *Proceedings of the NASECODE I Conference*, B. T. Browne and J. J. H. Miller, Eds., Boole Press, Dublin, Ireland, 1979, pp. 31–64.
6. E. M. Buturla, P. E. Cottrell, B. M. Grossman, and K. A. Salsburg, "Finite-Element Analysis of Semiconductor Devices: The FIELDAY Program," *IBM J. Res. Develop.* **25,** 218–231 (1981, this issue).
7. T. Adachi, A. Yoshii, and T. Sudo, "Two-Dimensional Semiconductor Analysis Using Finite Element Methods, *IEEE Trans. Electron Devices* **ED-26,** 1026–1031 (1979).
8. S. Selberherr, A. Schutz, and H. W. Potzl, "MINIMOS—A Two-Dimensional MOS Transistor Analyser," *IEEE Trans. Electron Devices* **ED-27,** 1540–1550 (1980).
9. J. J. Barnes, K. Shimohigasha, and R. W. Dutton, "Short Channel MOSFETs in the Punchthrough Current Mode," *IEEE Trans. Electron Devices* **ED-26,** 446–453 (1979).
10. W. L. Engl and H. Dirks, "Numerical Device Simulation Guided by Physical Approaches," *Proceedings of the NASECODE I Conference*, B. T. Browne and J. J. H. Miller, Eds., Boole Press, Dublin, Ireland, 1979, pp. 65–93.
11. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1973.
12. J. T. Oden, *Finite Elements of Nonlinear Continua*, McGraw-Hill Book Co., Inc., New York, 1972.
13. E. A. Wilson and W. E. Tchon, "Calculation of Transfer Potentials Using the Finite Element Method," *1973 SWIEEECO Record of Technical Papers*, 25th Annual Southwestern IEEE Conference and Exhibition, Houston, TX, April 1973.
14. L. J. Turgeon and D. H. Navon, "Two-Dimensional Non-Isothermal Carrier Flow in a Transistor Structure under Reactive Circuit Conditions," *IEEE Trans. Electron Devices* **ED-25,** 837–843 (1978).
15. J. J. Barnes, R. J. Lomax, and G. I. Haddad, "Finite Element Simulation of GaAs MESFET's with Lateral Doping Profiles and Submicron Gates," *IEEE Trans. Electron Devices* **ED-23,** 1042–1048 (1976).
16. G. D. Hachtel and M. H. Mack, "A Graphical Study of the Current Distribution in Short Channel IGFETS," *Digest of Technical Papers*, IEEE Int. Solid-State Circuits Conference, Philadelphia, PA, 1973, pp. 110–111.
17. K. Fukahori and P. R. Gray, "Computer Simulation of Integrated Circuits in the Presence of Electrothermal Interaction," *IEEE J. Solid-State Circuits* **SC-11,** 834–846 (1976).
18. R. H. MacNeal, "An Asymmetrical Finite Difference Network," *Quart. Appl. Math.* **XI,** 295–310 (1953).
19. W. T. Weeks, A. J. Jimenez, G. W. Mahoney, D. Mehta, H. Qassemzadeh, and T. R. Scott, "Algorithms for ASTAP—A Network Analysis Program," *IEEE Trans. Circuit Theory* **CT-20,** 628–634 (1973).
20. S. P. Gaur, "Two-Dimensional Carrier Flow in a Transistor Structure Under Non-Isothermal Conditions," Ph.D. Dissertation, University of Massachusetts, Amherst, 1974.
21. G. D. Hachtel, R. K. Brayton, and F. G. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design," *IEEE Trans. Circuit Theory* **CT-18,** 101–113 (1971).
22. *IBM System/360 and System/370, Subroutine Library-MATHematics, User's Guide*, Order No. SH12-5300, available through IBM branch offices.
23. J. A. George, "Nested Dissection of a Regular Finite Element Mesh," *SIAM J. Numer. Anal.* **10,** 345–363 (1973).

24. G. D. Hachtel and M. H. Mack, "A Pseudo-Dynamic Method for the Solution of Nonlinear Algebraic Equations," *Stiff Differential Systems*, R. A. Willoughby, Ed., Plenum Press, Inc., New York, 1974.
25. H. K. Gummel, "A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations," *IEEE Trans. Electron Devices* **ED-11,** 455–465 (1964).
26. S. Sze, *Physics of Semiconductor Devices*, Wiley-Interscience Publishers, New York, 1969.
27. I. S. Duff, A. M. Erisman, and J. K. Reid, "On George's Nested Dissection Method," *SIAM J. Numer. Anal.* **13,** 686–695 (1976).
28. A. H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.
29. E. M. Buturla and P. E. Cottrell, "Simulation of Semiconductor Transport Equations Using Coupled and Decoupled Solution Techniques," *Solid State Electron.* **23,** 331–334 (1980).
30. B. Speelpenning, "The Generalized Element Method, A Preliminary Report," *Amer. Math. Soc. Notices* **20**, Notice No. 73T-C18, p. A-280 (1973).
31. C. McMullen, F. G. Gustavson, and E. M. Buturla, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1981, unpublished results.

Received October 10, 1980; revised February 25, 1981

G. D. Hachtel and M. H. Mack are located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598; R. R. O'Brien is located at the IBM General Technology Division laboratory, East Fishkill Facility, Hopewell Junction, New York 12533.