Fast and Smooth Highly Nonlinear Multidimensional Table Models for Device Modeling

PETER B. L. MEIJER

Abstract — **This paper presents a general scheme for the construction of device models for circuit simulators. Two general n-dimensional C^1 table models have been constructed under this scheme. Each table model can automatically reconstruct the exact behavior of the dc current expressions of two basic physical device models, namely the Ebers–Moll bipolar transistor model and the GLASMOST MOSFET model. Also, the evaluation times of the three-dimensional table model implementations are less than those of advanced physical CAD device models. The table models are generally very accurate and have negligible model development time. Both table models have been implemented in the SPICE-like circuit simulator PHILPAC.**

I. Introduction

THE QUALITY of device models is decisive for the attainable quality in the prediction and verification of the circuit behavior for a given electronic network. Device models are used to represent, for example, the terminal currents and charges of transistors as a function of the terminal voltages. The requirements for a good model in a circuit simulation environment are quite stringent. The models must obviously be accurate, because they limit the attainable simulation accuracy. Additional model restrictions may be posed by the simulation algorithms, such as the Newton–Raphson algorithm and the time integration scheme: good models are at least continuous as a function of each of the controlling variables, and preserve monotonicity in the device behavior. Usually the continuity of the first partial derivatives is also required, causing C^1 smoothness. Without these numerical properties, the simulation may take much longer to process, or it may even fail. Other restrictions follow from the charge and energy conservation laws. One should also pay attention to the model evaluation times, because state-of-the-art device models often cause a significant fraction of the simulation time to be spent in the evaluation of model expressions. It should be noted, however, that this fraction depends on many factors, among which are the circuit size, the model complexity, the simulation algorithms, and the degree of parallelism exploited.

Manuscript received January 3, 1989; revised July 17, 1989. This work was supported in part by the Dutch Government with Mega-Project subsidies. This paper was recommended by Associate Editor C.A.T. Salama.

The author is with Philips Research Laboratories, 5600 JA Eindhoven, The Netherlands.

IEEE Log Number 8933509.

Physical modeling has for many years been the most attractive approach to semiconductor device modeling for circuit simulation. It was possible to obtain very compact models by exploitation of knowledge about the underlying physical principles. Only a few relatively independent parameters were needed to arrive at a sufficiently accurate result. The parameters had a clear physical meaning, which is important for the feedback of simulation results to process control and optimization. Model development times were still acceptable, because in the relatively large devices one could neglect a number of physical side-effects, without much loss in accuracy and with a gain in simplicity. In particular one could, mainly from geometrical considerations, reduce the analysis of device operation to the analysis of a number of one-dimensional phenomena.

Unfortunately, this situation is now gradually deteriorating. One has to include more second-order effects for the very small devices in VLSI technology. The assumption of a one-dimensional current channel in transistors becomes increasingly inaccurate, due to nonuniform electrical fields and doping levels near the boundaries of the channel. Geometrical boundary effects in general become more influential, because the ratio of surface to volume increases for smaller devices. One is forced to perform very complicated multidimensional analyses, often resulting in complicated model expressions. Even the combination of the results of several one-dimensional analyses into a C^1 and, if relevant, monotonic multidimensional description becomes a nontrivial task. (In this paper, a multidimensional function is considered monotonic if it is monotonic as a function of any one of its controlling variables, keeping the remaining variables at any set of fixed values.) Furthermore, surface physics, e.g., for surface states and mobility degradation, is not as well developed as bulk physics. This means that the required boundary conditions in the analyses may not even be known. The most advanced physical models now use a significant number of largely empirical fit-parameters, and sometimes unphysical values are assigned to the physical parameters, because they have to account for effects that were not properly incorporated in the model. This may also reduce the independence of the parameters. Because of the increased model and modeling

0098-4094/90/0300-0335$01.00

complexity, both the model evaluation time during circuit simulation and the model development time increase.

A growing interest in the use of table models instead of physical models has been the consequence. The table models approximate the device behavior, using a finite number of table points, which can be obtained from measurements or device simulations (or from existing physical models, as is done in this paper for testing purposes). A number of recent papers on the application of multidimensional table models for semiconductor device modeling can be found in the references [1]–[4], [6]–[9], [12], [13], [15]–[18], [20], [21]. Table models have the advantage of a very short model development time, and it is often possible to guarantee some important numerical properties beforehand, such as C^1 smoothness and monotonicity. Piecewise-polynomial table models, e.g., tensor products of B-splines [4], [5], [8], [17], [21], are very popular. For B-spline tensor products, monotonicity is guaranteed when using a set of monotonic B-spline coefficients [5], [8]. One may simply take table values as a set of coefficients, which leads to the so-called variation diminishing (VD) splines. However, high accuracy in multidimensional piecewise-polynomial models can be obtained only by using many table points. In particular, the highly nonlinear exponential characteristics of bipolar and MOS transistors cannot be modeled without using excessive numbers of table points. Ad hoc hybrid combinations of piecewise-polynomial table models and physical models can be and have been made to deal with this problem, but only at the expense of decreasing generality and increasing model development time, sometimes sacrificing some smoothness guarantees as well. Table models, being black-box models, also have the disadvantage that different parameters generally do not represent different physical phenomena, as each parameter represents an unknown mix of underlying physical effects. This impedes the interpretation of model characteristics, and hence, the feedback to technology. However, this argument is somewhat weakened by the fact that today the physical models also incorporate many empirical parameters.

This paper describes a general hybrid approach to automatic device modeling. The hybrid approach combines several advantages of physical modeling and piecewise-polynomial table modeling, while it avoids several of their disadvantages. Modeling the (partly) exponential behavior of many semiconductor device characteristics requires a more general approach than is found in piecewise-polynomial table models. The local expressions of a piecewise model should reflect the qualitative behavior of the particular region in which the model piece lies. For example, a device characteristic that behaves (locally) as a function $f(x) = e^{ax}$, $a \gg 1$, is best modeled (locally) by a simple function that can express this behavior exactly, in order to obtain an accurate representation that requires few table points. A piecewise-polynomial description would require a high density of table points to accurately model the same behavior. Accurate piecewise-polynomial modeling of highly nonlinear multidimensional behavior such as $f(x_1, x_2) = e^{a_1 x_1} + e^{a_2 x_2}$, $a_1, a_2 \gg 1$, soon becomes impractical. Therefore, the basic strategy followed in this paper is to apply general physics-based heuristics to select an appropriate one-dimensional local description. The local descriptions are then combined into a C^1 multidimensional model. Transitions in qualitative behavior are detected and processed automatically, guided by the heuristics. In a recent paper [15], a very general approach to such an automatic construction of smooth and highly nonlinear multidimensional table models was presented. High model accuracy was obtained using few table points. The MOSFET table model example was C^1 and monotonic. However, table preprocessing was hampered by the infinite number of so-called virtual table points, causing the model evaluation times to be rather long on sequential computers. This paper presents a scheme without virtual table points. Extensive table preprocessing now becomes possible, by shifting the application of the heuristics to a preprocessing phase. This gives a dramatic decrease in the model evaluation times during circuit simulation, at the expense of an acceptable increase in memory demands after preprocessing (for choosing functions and calculating coefficients). What is important is that the number of measurements or device simulations, needed to provide the data, does not increase. The basic principles stated in [15] still apply to this paper, only the methods are different. The slight loss in generality, because not all n-dimensional behavior is covered (exactly) by the methods of this paper, appears to be no disadvantage in practice. The behavior of bipolar transistors as well as MOSFET's can be represented very accurately for a suitable choice of the controlling variables. The exact reconstruction of the behavior of the dc current expressions of a basic bipolar transistor model and a basic MOSFET model, as shown in this paper, reflects a new view to estimating the general usefulness of table models. It also demonstrates the promise of high accuracy with reasonable amounts of data, when following the approach described in this paper.

II. Methods

Let there exist a (generally unknown) C^1 n-dimensional function $\Phi(x_1, \cdots, x_n)$ that exactly describes (part of) the behavior of a device. The aim is to approximate this function as closely as possible. In semiconductor device models, the controlling variables $x_1, \cdots, x_n$ are usually a set of independent voltages across terminal pairs of an $(n+1)$-terminal device. The function Φ represents a dependent physical quantity, such as a current or a charge. It is assumed that the state of a device is determined instantaneously by the externally applied controlling variables. This assumption implies that phase shift effects due to the nonzero response times of a device are neglected. The dynamic behavior is then determined solely by the dc characteristics for currents and the time-derivatives of equivalent terminal charges [19].

For the construction of a table model, it is convenient to impose some regularity upon the table structure. Therefore, given the numbers $n \in \mathbb{N}^+$, $N_k \in \mathbb{N}^+$, $k = 1, \cdots, n$ and $x_{k,i_k} \in \mathbb{R}$ with $x_{k,1} < \cdots < x_{k,N_k}$, $k = 1, \cdots, n$, we

consider an n-dimensional hyperrectangular grid Γ_N consisting of points $(x_{1,i_1},\cdots,x_{n,i_n})$:

$$\Gamma_N \triangleq \{(x_{1,i_1},\cdots,x_{n,i_n}) | k=1,\cdots,n: i_k=1,\cdots,N_k \wedge x_{k,1}<\cdots<x_{k,N_k}\}. \quad (2.1)$$

The grid has a table value $F_{i_1,\cdots,i_n}\in\mathbb{R}$ associated with each of the grid points $(x_{1,i_1},\cdots,x_{n,i_n})$. x_{k,i_k} is the i_kth discrete grid position in the kth direction.

To avoid cluttering the following text with conditions, we will not consider the boundaries of the table. The choice of an extrapolation scheme is a separate problem lying beyond the scope of this paper. For the same reason, the value range of indexes in the formulas is not given, unless there exist relevant restrictions that are not related to the table boundaries. The type of a variable (real or integer) should from now on be clear from the context.

Exact table values and exact calculations are assumed, to distinguish the fundamental theoretical limitations of the table models from practical limitations due to measurement errors, physical noise in the data and/or finite precision in the computer representations and operations. So we have $F_{i_1,\cdots,i_n}\equiv\Phi(x_{1,i_1},\cdots,x_{n,i_n})$.

For the moment, we also assume that we have methods that are powerful enough to find exact descriptions along all gridlines $(x_{1,i_1},\cdots,x_{k-1,i_{k-1}},x_k,x_{k+1,i_{k+1}},\cdots,x_{n,i_n})$.

Exact gridline descriptions still give insufficient information about the behavior elsewhere in hyperspace. For example, in the case of equidistant gridlines one may add any (even infinitely differentiable) function:

$$H(x_1,\cdots,x_n)\cdot\prod_{k=1}^{n}\sin\left(\frac{x_k-x_{k,i_k}}{x_{k,i_k+1}-x_{k,i_k}}\cdot\pi\right) \quad (2.2)$$

for arbitrary i_k and bounded H, without modifying the behavior along the gridlines in any way. Nevertheless, we can make a reasonable guess about the relation between behavior along gridlines and behavior elsewhere. In fact, we will make two different guesses, both of which turn out to be correct (exact) for the current expressions of two basic physical device models, namely the Ebers–Moll bipolar transistor model [10] and the GLASMOST MOSFET model [19].

It is advantageous to construct local n-dimensional descriptions, to enable accurate modeling of a device characteristic with qualitatively different behavior in neighboring regions. The local n-dimensional descriptions are constructed from local one-dimensional descriptions. The exact behavior of Φ along any gridline $(x_{1,i_1},\cdots,x_{k-1,i_{k-1}},x_k,x_{k+1,i_{k+1}},\cdots,x_{n,i_n})$ in the interval $x_k\in[x_{k,i_k},x_{k,i_k+1}]$ will, therefore, be denoted by C^1 functions $\phi^{(k)}_{i_1,\cdots,i_n}(x_k)$, giving

$$\phi^{(k)}_{i_1,\cdots,i_n}(x_{k,i_k})=F_{i_1,\cdots,i_n}$$
$$\phi^{(k)}_{i_1,\cdots,i_n}(x_{k,i_k+1})=F_{i_1,\cdots,i_{k-1},i_k+1,i_{k+1},\cdots,i_n}. \quad (2.3)$$

A useful local interpretation of this definition is that any particular grid point $(x_{1,i_1},\cdots,x_{n,i_n})$ has n associated functions $\phi^{(k)}_{i_1,\cdots,i_n}(x_k)$, because $k=1,\cdots,n$. All these functions interpolate the table value $F_{i_1,\cdots,i_n}$ at this grid point. The kth function also interpolates the table value belonging to the next grid point in the kth direction, i.e., the neighboring table point at x_{k,i_k+1}. The functions $\phi^{(k)}$ drawn in Fig. 1 may help to visualize this interpretation for $n=2$. The other function parts of Fig. 1 are discussed in a following section.

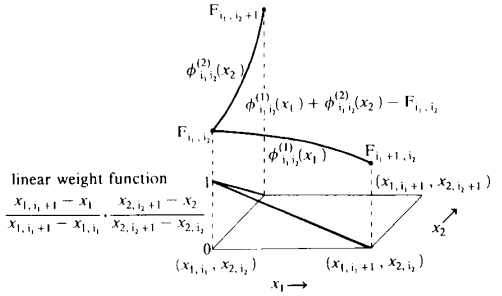


Fig. 1. Table model 1 principles for $n=2$, $\tau=\mathbf{0}$.

The transformation of behavior in $\mathbb{R}$ along the set of (piecewise) gridlines into behavior in $\mathbb{R}^n$ will be denoted by the grid-operator $\hat{\gamma}$:

$$\hat{\gamma}\{\phi^{(k)}_{i_1,\cdots,i_n}(x_k)\}\triangleq F(x_1,\cdots,x_n) \quad (2.4)$$

with the requirements that F should at least be C^0 in the x_k direction at x_{k,i_k} and C^1 elsewhere. In other words, discontinuities in the first partial derivative to x_k are allowed only at $x_k=x_{k,i_k}$. Later on an operator will be defined that transforms such a piecewise C^1 F into a globally C^1 function. We also wish to preserve the assumedly exact descriptions along the gridlines, so we require

$$F(x_{1,i_1},\cdots,x_{k-1,i_{k-1}},x_k,x_{k+1,i_{k+1}},\cdots,x_{n,i_n})\equiv\Phi(x_{1,i_1},\cdots,x_{k-1,i_{k-1}},x_k,x_{k+1,i_{k+1}},\cdots,x_{n,i_n}). \quad (2.5)$$

Once we have constructed such a $\hat{\gamma}$, the question arises what class of functions is reconstructed exactly from exact gridline descriptions, i.e., which functions Φ are eigenfunctions of $\hat{\gamma}$, fulfilling

$$\hat{\gamma}\{\phi^{(k)}_{i_1,\cdots,i_n}(x_k)\}=\Phi(x_1,\cdots,x_n). \quad (2.6)$$

A function Φ that meets this condition needs only the finite set of exact descriptions along gridlines to obtain the exact n-dimensional behavior. For $\hat{\gamma}$ to be useful, multidimensional expressions of some basic physical device models should be eigenfunctions of $\hat{\gamma}$.

In the following, τ denotes a vector of dimension n with elements 0 or 1, and the summation over τ from $\mathbf{0}$ to $\mathbf{1}$ sums all 2^n different combinations. We now consider two grid-operators, $\hat{\gamma}_1$ and $\hat{\gamma}_2$, each fulfilling the conditions on F mentioned between (2.4) and (2.5). Both operators are symmetrical in the controlling variables $x_1,\cdots,x_n$, and both involve only $n2^{n-1}$ functions $\phi^{(k)}$ in a model evaluation, i.e., the number of edges of an n-dimensional hyperrectangle.

2.1. Table Model 1

For $x_k \in [x_{k,i_k}, x_{k,i_k+1}]$, $k=1,\cdots,n$, the local n-dimensional description is defined by

$$\hat{\gamma}_1\left\{\phi^{(k)}_{i_1,\cdots,i_n}(x_k)\right\} \triangleq F_1(x_1,\cdots,x_n) \triangleq \sum_{\tau=\mathbf{0}}^{\mathbf{1}}\left[(1-n)F_{i_1+\tau_1,\cdots,i_n+\tau_n} + \sum_{k=1}^{n}\phi^{(k)}_{i_1+\tau_1,\cdots,i_{k-1}+\tau_{k-1},i_k,i_{k+1}+\tau_{k+1},\cdots,i_n+\tau_n}(x_k)\right] \cdot \prod_{k=1}^{n}\left(\tau_k+(-1)^{\tau_k}\cdot\frac{x_{k,i_k+1}-x_k}{x_{k,i_k+1}-x_{k,i_k}}\right). \tag{2.7}$$

This expression turns out to be an n-dimensional generalization of the two-dimensional Coons surface patch with linear blending functions, see [11]. It can also be viewed as a multilinear weighting of nonlinear multidimensional functions, each of them interpolating (at least) $n+1$ of the table points located at the vertices of the hyperrectangle spanned by $x_k \in [x_{k,i_k}, x_{k,i_k+1}]$. Fig. 1 illustrates this view for the part of (2.7) that corresponds to $n=2$, $\tau=\mathbf{0}$.

2.2. Table Model 2

For $x_k \in [x_{k,i_k}, x_{k,i_k+1}]$, $k=1,\cdots,n$, the local n-dimensional description is defined by

$$\hat{\gamma}_2\left\{\phi^{(k)}_{i_1,\cdots,i_n}(x_k)\right\} \triangleq F_2(x_1,\cdots,x_n) \triangleq \sum_{\tau=\mathbf{0}}^{\mathbf{1}} F_{i_1+\tau_1,\cdots,i_n+\tau_n}\cdot\prod_{k=1}^{n}\left[\tau_k+(-1)^{\tau_k}\cdot w^{(k)}_{i_1+\tau_1,\cdots,i_{k-1}+\tau_{k-1},i_k,i_{k+1}+\tau_{k+1},\cdots,i_n+\tau_n}(x_k)\right] \tag{2.8}$$

with weight functions $w^{(k)}$ defined by

$$w^{(k)}_{i_1,\cdots,i_n}(x_k) \triangleq \frac{F_{i_1,\cdots,i_{k-1},i_k+1,i_{k+1},\cdots,i_n}-\phi^{(k)}_{i_1,\cdots,i_n}(x_k)}{F_{i_1,\cdots,i_{k-1},i_k+1,i_{k+1},\cdots,i_n}-F_{i_1,\cdots,i_n}} \tag{2.9}$$

and the conditions

$$F_{i_1,\cdots,i_{k-1},i_k+1,i_{k+1},\cdots,i_n} \neq F_{i_1,\cdots,i_n} \tag{2.10}$$

to avoid division by zero in (2.9). In practice one may apply small (insignificant) changes to the table values to satisfy conditions (2.10). Or else one may redefine the $w^{(k)}$ for the cases where division by (near-)zero values would cause problems. However, these modifications will be considered no further in this paper, because they violate the exactness assumptions in the theoretical treatment of exact reconstructability. Table model 2 can be viewed as a nonlinear weighting, with products of weights $w^{(k)}$ or $(1-w^{(k)})$, of the table values at the vertices of the hyperrectangle spanned by the controlling variables x_k. Fig. 2 illustrates this view for the part of (2.8) that corresponds to $n=2$, $\tau=\mathbf{0}$.

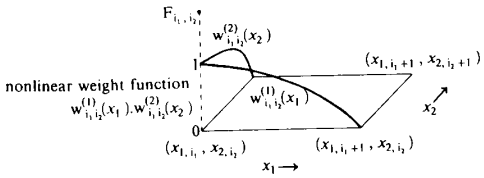


Fig. 2. Table model 2 principles for $n=2$, $\tau=\mathbf{0}$.

2.3. Eigenfunctions of Table Models 1 and 2

The usefulness of any table model for device modeling purposes is determined to a large extent by the behavior it can accurately represent, when using few table points. A good indication for this is the class of multidimensional functions that can be expressed exactly. In other words, we are interested in the eigenfunctions of $\hat{\gamma}$ satisfying (2.6), i.e., assuming exact descriptions along the gridlines. In the following, $0^0 \triangleq 1$:

A set of eigenfunctions of $\hat{\gamma}_1$ is given by

$$G_1(x_1,\cdots,x_n) \triangleq \sum_{k=1}^{n}\sum_{\tau=\mathbf{0}}^{\mathbf{1}} g^{(k)}_{\tau}(x_k)\cdot\prod_{i=1}^{n}x_i^{\tau_i} \tag{2.11}$$

with $g^{(k)}_{\tau}$ representing any C^1 function, in which τ and k are just indexes to identify different functions. The terms with $\tau_k=1$ may be dropped from (2.11) without loss of generality. For example, in the two-dimensional case, expression (2.11) is equivalent to the form

$$G_1(x_1,x_2) = g_1(x_1)+g_2(x_2)+x_2g_3(x_1)+x_1g_4(x_2). \tag{2.12}$$

A set of eigenfunctions of $\hat{\gamma}_2$ is given by

$$G_2(x_1,\cdots,x_n) \triangleq \sum_{\tau=\mathbf{0}}^{\mathbf{1}} c_{\tau}\cdot\prod_{k=1}^{n}\left[g^{(k)}(x_k)\right]^{\tau_k} \tag{2.13}$$

with $g^{(k)}$ being C^1 functions and c_{τ} being constants. Notice that g lacks the index τ in the expression for G_2. Therefore, in the two-dimensional case, expression (2.13) is equivalent to the form

$$G_2(x_1,x_2) = c_1+c_2g_1(x_1)+c_3g_2(x_2)+c_4g_1(x_1)g_2(x_2). \tag{2.14}$$

An outline of the general proofs for the exact reconstructability of (2.11) and (2.13) is given in the Appendix.

Any multilinear form can be written in the form (2.11) as well as (2.13), by using linear functions g. Therefore, piecewise-linear table models are special cases of both table model 1 and 2, provided the conditions (2.10) are satisfied in the case of table model 2.

The two table models are also equally powerful in the sense that model 1 can exactly model some functions that model 2 cannot, and vice versa. From the definition of $\hat{\gamma}_1$ it is obvious that table model 1 cannot exactly describe multidimensional products of highly nonlinear functions, while table model 2 can. On the other hand, it can be proven by inspection of (2.12) and substitution in (2.9) and (2.8) that the function $f(x_1,x_2)=x_1/x_2+x_2/x_1$ can

be exactly described by table model 1, but not by table model 2.

2.4. Smoothing Shell for Table Models 1 and 2

Because the $\phi^{(k)}$ are C^1, the table models are C^1 within the local descriptions at $x_k \in [x_{k,i_k}, x_{k,i_k+1}]$, $k = 1, \cdots, n$. Continuity of the two table models at the borders between neighboring local descriptions is shown by verifying that for all $j = 1, \cdots, n$:

$$\lim_{x_j \uparrow x_{j,i_j}} \hat{\gamma}\left\{\phi^{(k)}_{i_1,\cdots,i_n}(x_k)\right\} \equiv \lim_{x_j \downarrow x_{j,i_j}} \hat{\gamma}\left\{\phi^{(k)}_{i_1,\cdots,i_n}(x_k)\right\}. \quad (2.15)$$

Therefore, the table models 1 and 2 are C^1 models, except for possible discontinuities in the first partial derivatives at x_{k,i_k} in the x_k-directions. Furthermore, by substitution of $x_j = x_{j,i_j}$, for all $j \neq k$, it is found that $\hat{\gamma}_1$ and $\hat{\gamma}_2$ leave the descriptions along gridlines unaltered, the result being one of the $\phi^{(k)}$. This satisfies condition (2.5). A C^1 generalization of $\hat{\gamma}_1$ and $\hat{\gamma}_2$ was deliberately avoided to minimize model evaluation times. It is better to design a smoothing scheme as a separate shell around the basic table models. This allows one to keep control of the amount of smoothing required for a specific application. The basic idea behind this is the following. If, owing to good heuristics, a C^0 model for Φ is already accurate, including the first partial derivatives, then additional smoothing will make little difference, and we may smooth only close to x_{k,i_k}. If close enough, then the contribution of the smoothing scheme to the average model evaluation time becomes negligible, assuming a more or less uniform distribution of argument values by which the model implementation is called. Ideally, a smoothing shell would yield a C^1 model, while preserving the monotonicity (if present) of the unsmoothed model and its first partial derivatives. Neighboring descriptions should be left unaltered if the behavior across their border is already C^1, to avoid distorting accurate or even exact descriptions. Preservation of symmetry in the controlling variables of the modelling scheme is also desirable.

A particularly well-behaved smoothing scheme is the following one. A function F, representing table model 1 or 2, is transformed into a globally C^1 function $F^{(s)}$ by an operator $\hat{S}$:

$$\hat{S}F(x_1, \cdots, x_n) \triangleq F^{(s)}(x_1, \cdots, x_n) \quad (2.16)$$

with

$$\hat{S} \triangleq \prod_{k=1}^{n} \left[1 + \sum_{i_k} s(x_k, x_{k,i_k}, \epsilon_{k,i_k}) \cdot \left(\lim_{x_k \downarrow x_{k,i_k}} \frac{\partial}{\partial x_k} - \lim_{x_k \uparrow x_{k,i_k}} \frac{\partial}{\partial x_k} \right)\right] \quad (2.17)$$

and

$$s(x_k, x_{k,i_k}, \epsilon_{k,i_k}) \triangleq \begin{cases} \dfrac{(x_k - x_{k,i_k} + \epsilon_{k,i_k})^2}{4\epsilon_{k,i_k}}, & x_{k,i_k} - \epsilon_{k,i_k} \leqslant x_k < x_{k,i_k} \\ \dfrac{(x_k - x_{k,i_k} - \epsilon_{k,i_k})^2}{4\epsilon_{k,i_k}}, & x_{k,i_k} \leqslant x_k < x_{k,i_k} + \epsilon_{k,i_k} \\ 0, & \text{otherwise} \end{cases} \quad (2.18)$$

where $\epsilon_{k,i_k} > 0$ denotes the smoothing range around x_{k,i_k}, using the no-overlap condition:

$$x_{k,i_k+1} - x_{k,i_k} \geqslant \epsilon_{k,i_k+1} + \epsilon_{k,i_k}. \quad (2.19)$$

This scheme has all the above-mentioned desirable properties, except for a general guarantee for monotonicity preservation. Nevertheless, the scheme was designed to preserve monotonicity in most practical cases. For any n-dimensional C^0 piecewise-multilinear function on a hyperrectangular grid, the monotonicity preservation of this function and even of its first partial derivatives can be proven. This can be done in a straightforward manner, by applying the smoothing scheme to a piecewise-multilinear function written as a tensor product of linear B-splines. So the scheme is ideal for smoothing piecewise-linear models, which are special cases of table models 1 and 2. The symmetry arises because all the cross-derivatives that occur in the scheme exist and are C^0 for an F being a piecewise sum of products of C^1 one-dimensional functions in different directions, as generated by $\hat{\gamma}_1$ or $\hat{\gamma}_2$. This makes the order of differentiation irrelevant. Fig. 3 illustrates the effect of the smoothing shell when applied to a two-dimensional piecewise-linear model.

2.5. Algorithms for Choosing Local One-Dimensional Descriptions

To study the expressive power of the table models, it was assumed that we could obtain exact descriptions along the gridlines. This is not realistic, of course. However, we can come close by using suitable heuristics, that choose the most adequate local description from a basis of C^1 primary functions as in [15], in short $\phi^{(k)}_{i_1,\cdots,i_n}(\cdot) \in \{\text{primary}(\cdot)\}$. In this paper only the following two primary functions are used:

$$\begin{aligned} \text{primary}_1(x) &\triangleq a_0 + a_1 x + a_2 x^2 \\ \text{primary}_2(x) &\triangleq b_0 + b_1 \cdot b_2^x \end{aligned} \quad (2.20)$$

The choice of the local descriptions and the processing of coefficients to interpolate between $(x_{k,i_k}, F_{i_1,\cdots,i_n})$ and $(x_{k,i_k+1}, F_{i_1,\cdots,i_{k-1},i_k+1,ik+1,\cdots,i_n})$ occurs at the start-up of the circuit simulation, which makes the increased memory usage only temporary.

One set of heuristics chooses a primary function for interpolating three table points, and the coefficients of this primary function are then calculated. The heuristics are

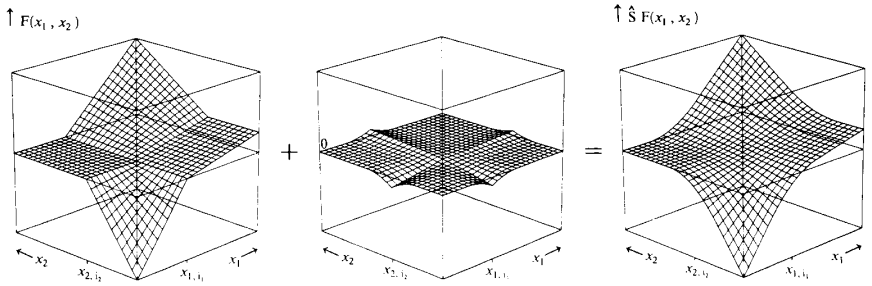


Fig. 3. Smoothing scheme applied to piecewise-linear model.

applied twice, once to the table points at x_{k,i_k-1}, x_{k,i_k} and x_{k,i_k+1}, and once to the table points at x_{k,i_k}, x_{k,i_k+1} and x_{k,i_k+2}. If the table values are strictly monotonic, the quadratic polynomial is normally chosen, unless this would cause a nonmonotonicity between x_{k,i_k} and x_{k,i_k+1}, in which case the exponential primary function is used. For nonmonotonic table values the quadratic polynomial is used. There are some exceptional cases, concerning nearly identical table values, for which a linear interpolation is chosen between x_{k,i_k} and x_{k,i_k+1}. Fig. 4 shows several cases for the heuristics, involving the table points at x_{k,i_k+2}.

Subsequently another set of heuristics decides which of the two interpolating primary functions, resulting from the above-mentioned two sets of three table points each, will ultimately be used to interpolate the table points at x_{k,i_k} and x_{k,i_k+1}. If the primary functions are of the same type, a new combination of coefficients may be formed to obtain an averaged result. Otherwise, the polynomial function has priority over the exponential function. As a result, the use of four table points in the heuristics helps to discriminate between leveling off as part of monotonic behavior and leveling off as part of a nonmonotonicity in the table values. In the latter case there is no reason to use an exponential form.

The physical motivation for the heuristics is found in the observation that, for many semiconductor devices, a sudden flattening of a device characteristic is caused by a sudden lack (depletion) of charge carriers somewhere in the device. Then the thermal generation of charge carriers may become the dominant physical effect for that part of the device characteristic. The thermal generation is governed by Boltzmann statistics, which lead to exponential dependencies on the applied voltages. Also, because many device characteristics for currents and charges are monotonic as a function of the applied voltages, a nonmonotonic approximating function for monotonic table values suggests that the wrong approximating function was chosen.

There is no compelling reason to use only the two primary functions (2.20). For example, for modeling JFET's it can be advantageous to include a square-root-like primary function to model the effects of depletion layer growth. Then it may be possible to obtain a further reduction in the number of table points needed for a given accuracy. Such an extension may not be worth the effort,

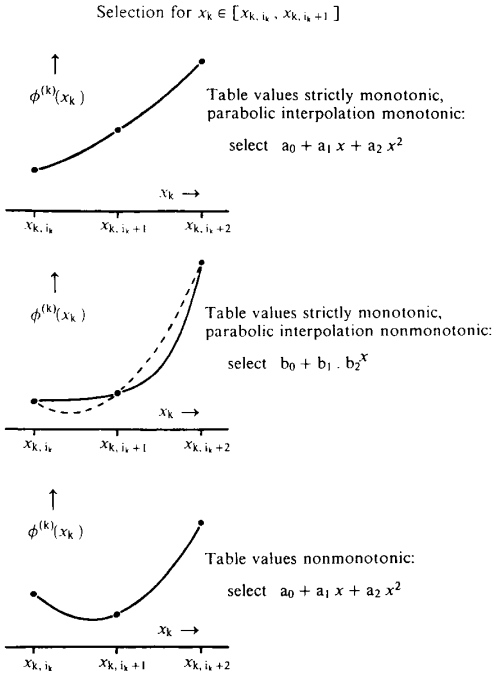


Fig. 4. Some typical heuristics for primary function selection.

however, because much of the square root behavior can be approximated with adequate storage efficiency by polynomial pieces. Furthermore, taking a square root on a computer is usually a rather costly (time-consuming) operation.

Piecewise-linear models can easily be obtained from table models 1 and 2 by simplifying the heuristics for choosing $\phi^{(k)}$ to the fixed choice of linearly interpolating pieces, and by disabling the smoothing operator, taking $\epsilon_{k,i_k} < 0$.

III. Ebers–Moll and GLASMOST Reconstructability

As stated before, the eigenfunctions of a table model should cover the multidimensional expressions of some basic physical device models. Also, the heuristics should be smart enough to find the exact behavior along the gridlines for these physical models. If these conditions are met, then the behavior of these physical models is reconstructed

exactly by the table models. Such a table model will probably also be suitable for modeling the less ideal but still strongly related behavior of real devices. Consider, therefore, the two-dimensional expressions for the dc currents $I_{C,B,E}(V_{BE}, V_{BC})$ of the Ebers–Moll bipolar transistor model and the dc currents $I_{ds}(V_{gd}, V_{gs})$ of the GLASMOST MOSFET model.

Ebers–Moll bipolar n-p-n transistor model [10]:

- Collector current I_C:

$$I_s[e^{qV_{BE}/k_BT}-1]-\frac{I_S}{\alpha_I}[e^{qV_{BC}/k_BT}-1].$$

- Base current I_B:

$$I_s\left(\frac{1}{\alpha_N}-1\right)[e^{qV_{BE}/k_BT}-1]+I_S\left(\frac{1}{\alpha_I}-1\right)[e^{qV_{BC}/k_BT}-1].$$

- Emitter current I_E:

$$I_s[e^{qV_{BC}/k_BT}-1]-\frac{I_S}{\alpha_N}[e^{qV_{BE}/k_BT}-1]. \quad (3.1)$$

GLASMOST n-channel MOSFET transistor model [19]:

- $V_{gs} \geq V_0 \wedge V_{gd} \geq V_0$, I_{ds} in linear region:

$$\frac{1}{2}\frac{W\mu C_{ox}}{L}[V_{gs}-V_{gd}][V_{gs}+V_{gd}+2V_{th}-2V_0].$$

- $V_{gs} \geq V_0 \wedge V_{gd} < V_0$, I_{ds} for drain saturation:

$$\frac{W\mu C_{ox}}{L}\Big[[V_{gs}-V_0][V_{gs}-V_0+2V_{th}] - V_{th}^2[e^{(V_{gd}-V_0)/V_{th}}-1]\Big].$$

- $V_{gs} < V_0 \wedge V_{gd} \geq V_0$, I_{ds} for source saturation:

$$\frac{W\mu C_{ox}}{L}\Big[V_{th}^2[e^{(V_{gs}-V_0)/V_{th}}-1] - [V_{gd}-V_0][V_{gd}-V_0+2V_{th}]\Big].$$

- $V_{gs} < V_0 \wedge V_{gd} < V_0$, I_{ds} in subthreshold region:

$$\frac{W\mu C_{ox}}{L}V_{th}^2[e^{(V_{gs}-V_0)/V_{th}}-e^{(V_{gd}-V_0)/V_{th}}]. \quad (3.2)$$

By inspection of (2.11) and (2.13), one may verify that for $n>1$ all of these current expressions can be written in the form (2.11) as well as (2.13). The same applies to the Ebers–Moll p-n-p transistor model and the GLASMOST p-channel transistor model. This means that the functions of Ebers–Moll and GLASMOST are in the intersection of the sets of eigenfunctions of $\hat{\gamma}_1$ and $\hat{\gamma}_2$. So both table models give the exact model behavior if the behavior along the gridlines is exact. Furthermore, for a single running variable, the behavior in each current expression is described by one of the primary functions among which the heuristics choose: it is either a quadratic polynomial or an exponential form. Therefore, using the table models and heuristics described in this paper, exact reconstruction of the behavior of the $I_{C,B,E}(V_{BE}, V_{BC})$ and $I_{ds}(V_{gd}, V_{gs})$ expressions is obtained. The only provision is that the voltage steps are not too small, to insure that the exponential behavior is not interpreted as polynomial behavior. However this just helps to reduce the table sizes. Only near the transition between different regions one may expect errors, because the heuristics then operate upon data from different expressions. However, for GLASMOST it can be verified that the heuristics described in the previous section give exact results even around the transition at the threshold voltage V_0, provided V_0 is a grid position and the b_2-value farthest from 1 is chosen when selecting among exponential functions as given in (2.20).

IV. Experimental Results

Three-dimensional versions of table models 1 and 2 and the smoothing shell have been programmed in Fortran, including their first partial derivatives. The models have been implemented in the circuit simulation program PHILPAC (a proprietary Philips package, comparable to SPICE). The three-dimensional table models can represent any four-terminal element by modeling the three independent terminal currents and charges as a function of the three independent voltages, i.e., taking a reference node and applying charge conservation. In general this leads to six tables per device, but provisions were made to use fewer tables whenever there is a smaller number of independent variables. For example, dc simulations using Ebers–Moll data could be done with only two current tables, e.g., for collector and base current, because of a zero substrate current. Transient simulations with GLASMOST data require three tables per transistor type (n-channel or p-channel). The three tables consist of one table for the drain (or source) current, because of zero gate and bulk currents, and two tables for the terminal charges, because of a zero bulk charge.

The table model implementations also include multiplication parameters in their parameter sets. These can be used to model a number of devices having different geometries, while using the current and charge tables of only a single device. This is done by multiplying the table model outcomes by a geometry-dependent parameter value, which may be different for currents and charges. In general, such a simple geometry scaling will be accurate only for a limited range of device geometries, and additional sets of tables are needed to model devices having geometries beyond this range.

Three-dimensional implementations of table models 1 and 2 were used in all experiments, even though for Ebers–Moll and GLASMOST a two-dimensional table model would have sufficed. Table values were obtained from the Ebers–Moll and GLASMOST models, for which the following model parameter values were used:

Ebers–Moll:	GLASMOST:
$k_BT/q = 30$ mV	$V_{th} = 30$ mV
$I_s = 2\times10^{-14}$ A	$\mu = 0.0675$ m$^2\cdot$V$^{-1}\cdot$s^{-1}
$\alpha_N = 0.985$	$C_{ox} = 1.38\times10^{-3}$ F$\cdot$m^{-2}
$\alpha_I = 0.72$	$V_0 = 1$ V
	$W = 20$ μm
	$L = 2$ μm

In all experiments the following nonequidistant grid was used: table points were positioned at integer-valued volt-

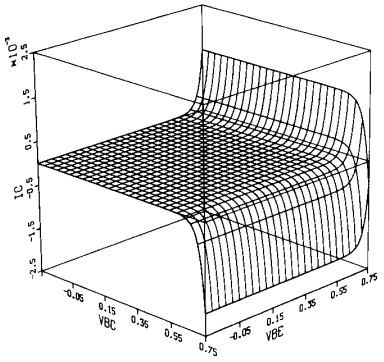


Fig. 5. Table model 1 for Ebers–Moll, $I_C(V_{BE}, V_{BC})$.

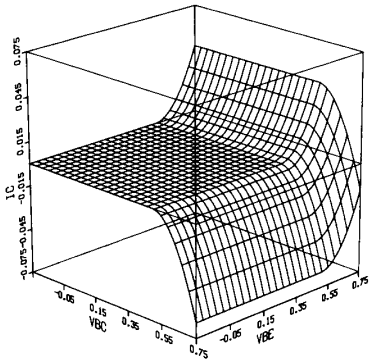


Fig. 6. QVD splines for Ebers–Moll, $I_C(V_{BE}, V_{BC})$.

ages, with additional table points at 0.5, 0.75, 1.25, and 1.5 V for V_{BE}, V_{BC} (n-p-n Ebers–Moll) and for V_{gd}, V_{gs} (n-channel GLASMOST). However, for the separate modeling of p-channel MOSFET behavior, the negated set of n-channel grid positions was used, in accordance with the negative value of the threshold voltage for the p-channel MOSFET. The p-channel MOSFET behavior was derived from the n-channel GLASMOST model by sign changes in currents, charges, and applied voltages. The smoothing ϵ's in (2.17) were 1 mV or less in all cases, which proved sufficient to obtain good convergence.

For comparison purposes, experiments were also performed with the C^1 three-dimensional tensor products of quadratic variation diminishing (QVD) splines [5], [8]. The set of knots was chosen identical to the non-equidistant grid used for table models 1 and 2. Sample points with corresponding table values were positioned midway between successive knots, as required for Schoenberg's QVD splines. The set of table values for QVD splines is, therefore, generally not the same as the set of table values for table models 1 and 2.

4.1. Ebers-Moll Bipolar Table Modeling Example

Fig. 5 shows the behavior of table model 1 for modeling the Ebers–Moll collector current. Exactly the same plot arises when using table model 2, or when using the

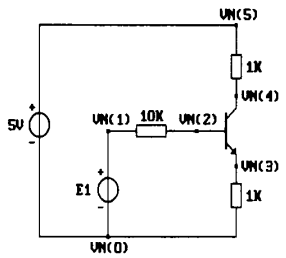


Fig. 7. Circuit schematic of a simple bipolar inverter.

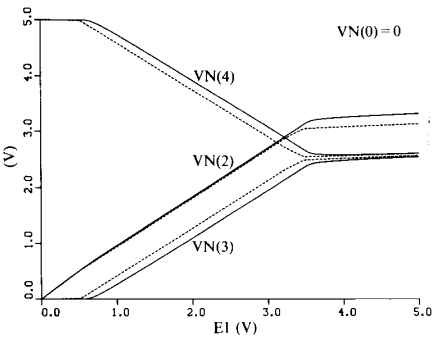


Fig. 8. DC curves of the bipolar inverter, using table model 1 (continuous lines) and QVD splines (dashed).

Ebers–Moll model itself, because of the exact reconstruction of Ebers–Moll behavior by both table models. Fig. 6 shows the behavior obtained using QVD splines. The collector currents modeled by the latter become more than an order of magnitude too large in the plotted voltage range. The reason is that most of the "tail" of a quadratic B-spline, associated with a particular sample point, decreases far too slowly to follow the exponential decrease in current with decreasing voltage. Significant consequences for circuit behavior can be demonstrated with the simple bipolar inverter shown in Fig. 7. Fig. 8 shows the PHILPAC simulation results for the dc transfer characteristics of this inverter, using table model 1 (solid lines) and QVD splines (dashed lines) with Ebers–Moll data for the current tables. Results obtained with table model 2 and the Ebers–Moll model itself are not shown, because they give the same characteristics as table model 1. It is worth noticing that the dashed lines predict the beginning of circuit response to occur just beyond $E1 = 0.5$ V. This erroneous result is a consequence of the sample points at $V_{BE} = 0.875$ V for large negative V_{BC}. The large associated current values are weighted by B-splines having their lowest voltage knots at $V_{BE} = 0.5$ V. Because this weight becomes nonnegligible soon beyond $V_{BE} = 0.5$ V, significant circuit currents are predicted just beyond $E1 = 0.5$ V. The table model 1 characteristics reflect the correct bipolar turn-on voltage of approximately 0.7 V. This example clearly demonstrates the importance of accurate highly nonlinear modeling. Table models 1 and 2 are capable of doing that, whereas QVD splines would require a much denser grid to yield accurate results.

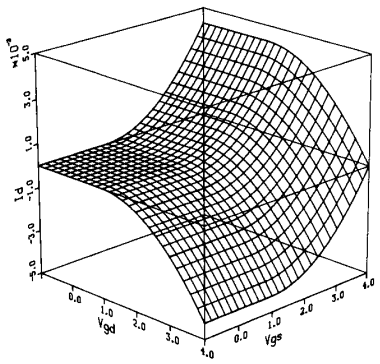


Fig. 9. Table model 1 for GLASMOST, $I_d(V_{gd}, V_{gs})$.

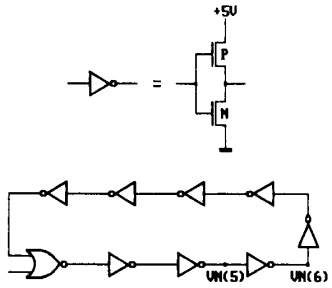


Fig. 10. A 9-stage CMOS ring oscillator circuit.

4.2. GLASMOST MOSFET Table Modeling Example

Fig. 9 shows the behavior of table model 1 for modeling the GLASMOST drain(source) current. Again, the same plot arises when using table model 2, or when using the GLASMOST model itself, because of the exact reconstruction of GLASMOST dc current behavior by both table models. The table model results are also exact near the transition at V_0, because $V_0 = 1$ V is a grid position. If circuit performance depends critically on the exponential subthreshold behavior of a MOSFET [19], [22], table models 1 and 2 will demonstrate major advantages over QVD splines, just as with bipolar modeling. However, in order to show that table models 1 and 2 also yield excellent results with rather subthreshold-insensitive digital circuits, simulations were performed for the 9-stage CMOS ring oscillator shown in Fig. 10. GLASMOST data were used for the current and charge tables. The PHILPAC simulation results for the dc transfer characteristic of a single CMOS inverter stage are shown in Fig. 11. QVD splines now give results (dashed line) that are close to the results obtained with the exact modeling by table model 1 (solid line). Results for table model 2 and GLASMOST itself are not shown, because their results coincide with the results for table model 1, due to the exact reconstruction of the dc current behavior. The good results obtained with QVD splines are not surprising, because the drain–source currents of one MOSFET operating within, or near, its subthreshold region, are too small to cause a significant drain–source voltage drop across the other, fully open, MOSFET. Only rather large currents, to which any exponential contribution in (3.2) can only make negligible contributions, will cause a significant drain–source voltage drop across a fully open MOSFET. The dc behavior of the inverter is, therefore, dominated by the polynomial parts of (3.2), which can be modeled accurately with QVD splines.

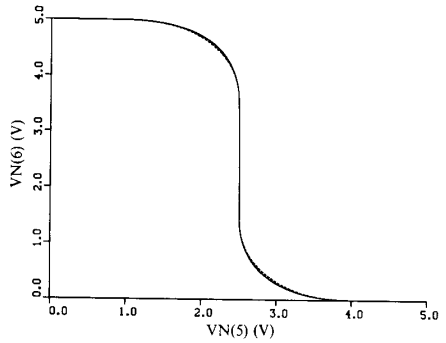


Fig. 11. DC curve of a single stage of the ring oscillator, using table model 1 (continuous line) and QVD splines (dashed).

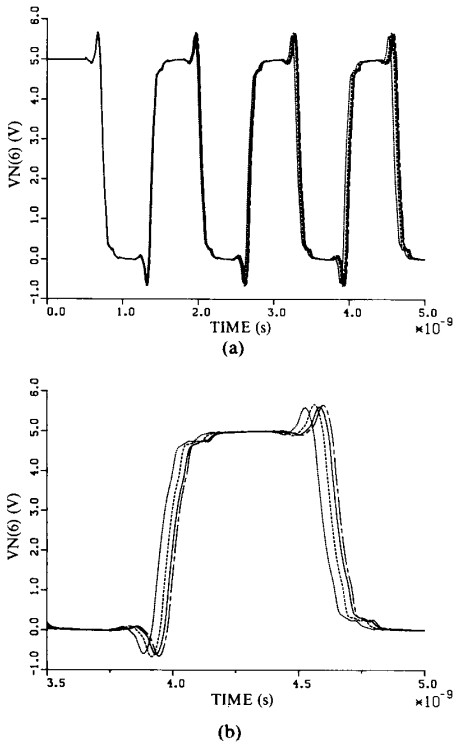


Fig. 12. (a) Transient simulations of the ring oscillator, using GLASMOST (continuous line), table model 1 (dashed), table model 2 (chain-dashed), and QVD splines (dotted). (b) Enlargement of the rightmost part of (a).

Fig. 12 shows the results of transient simulations, in which the oscillator was enabled after 0.5 ns. The step-size for time integration had an upper bound determined by the output points, which were requested every 5 ps. The timing of a ring oscillator strongly depends on the charge model. The GLASMOST charge model behavior is *not*

reconstructed exactly by table models 1 and 2. This explains why the transient response obtained with table model 1 (dashed line) and table model 2 (chain-dashed line) differs from the behavior obtained with the GLASMOST model (solid line). Still, the timing obtained with these table models is more accurate than the timing obtained with QVD splines (dotted line). Nevertheless, the results obtained with QVD splines are also good. Part of the explanation lies in the fact that the voltage dependencies of the GLASMOST charges rapidly become almost linear when entering the drain saturation and source saturation regions. Purely linear behavior is reconstructed exactly by QVD splines [5], [8], as well as by table models 1 and 2. The ring oscillator example demonstrates that table models 1 and 2 can also be attractive in problem cases that are known to be well suited to piecewise-polynomial modeling. An important advantage is that one no longer needs to assess whether a problem is well suited to piecewise-polynomial modeling.

4.3. Other Experimental Observations

The model evaluation times obtained for the three-dimensional implementations of table models 1 and 2 were within a few percent identical. The Fortran implementation of these table models for an IBM3090 scalar processor gave evaluation times per device characteristic between 31 and 53 μs (dependent on the MOSFET operating region). These times include the calculation of the three partial derivatives and the subroutine call overhead, and involve the use of a nonequidistant grid. This outperforms advanced physical CAD device models, for which evaluation times of 50 μs or more are not uncommon. The table model implementations can also compete in speed with (similarly optimized) implementations of conventional table models. For comparison, when using three-dimensional tensor products of QVD splines, evaluation times of 26 μs were obtained. Comparable model evaluation times of 37 μs were also obtained in [8], using QVD splines on a Cray-1A. The development time for adapting table models 1 and 2 to a new device is also negligible, involving a few seconds of preprocessing, using the same set of heuristics for bipolar as well as for MOSFET devices.

The increased memory usage of table models 1 and 2 at runtime, due to the preprocessing for primary function selection, can be seen as a disadvantage. For every table point and for each dimension, the primary function type is stored along with three double-word coefficients, leading to an approximate tenfold increase in storage for the three-dimensional table models *after* preprocessing. Without this preprocessing, simulation times would be much longer. QVD splines do not require such preprocessing. Nevertheless, QVD splines would require a far greater increase in storage in order to accurately model bipolar and MOSFET subthreshold dc current behavior. For example, considering the thermal voltage as a characteristic voltage step for a change in the exponential current behavior, a grid spacing of a few tens of millivolts would be more appropriate for QVD splines. This would give an order of magnitude increase over the number of sample point positions per dimension used in this paper. A tenfold increase in the number of sample point positions per dimension already implies a hundredfold increase in the number of table points in two dimensions. Moreover, substantially increasing the number of table points involves many additional measurements or device simulations, which may be much more costly than just the associated increase in memory usage. In practice, computer storage limitations have not yet presented any restrictions to the use of table models 1 and 2.

The Ebers–Moll and GLASMOST examples prove the power and usefulness of table models 1 and 2. Very accurate approximations have also been obtained for other, less ideal, device characteristics, using the same grid positions as described in this paper. Nevertheless, table model 1 will often be preferred over table model 2. Model 2 has extra conditions (2.10) on the table values. Furthermore, nonmonotonic functions $\phi^{(k)}$ occurring in (2.9) can lead to very large nonmonotonicities, because the values of the weights $w^{(k)}$ calculated from (2.9) may reach far beyond the range $[0,1]$, especially for nearly identical table values. Even if all interpolating $\phi^{(k)}_{i_1,\cdots,i_n}(x_k)$ are (forced to be) monotonic within their corresponding intervals $x_k \in [x_{k,i_k}, x_{k,i_k+1}]$, giving $w^{(k)}_{i_1,\cdots,i_n}(x_k) \in [0,1]$, it appears that nonlinear functions $w^{(k)}$ can lead rather easily to unwanted nonmonotonicities between gridlines. Present experience indicates that table model 1 is more robust for many practical situations.

V. Conclusions

The two table models presented in this paper combine several advantages of physical models and piecewise-polynomial table models, while avoiding several disadvantages. A good set of heuristics and primary functions makes it possible to obtain at least *some* indication of the likely physical background of the parameters. Automatic highly nonlinear modeling has become practical, as is demonstrated by the exact modeling of the dc expressions of a basic bipolar and a basic MOSFET model. The table models are also general purpose models, as shown by the fact that no change is needed to reconstruct the behavior of the two device models exactly, not even a change in the heuristics. The table models, as implemented in the circuit simulator PHILPAC, may well be the first table models capable of accurate automatic MOSFET subthreshold and bipolar modeling. Furthermore, the method evaluation times are short, as is the model development time. No time-consuming optimization procedures are needed to obtain accurate results. The smoothing shell guarantees C^1 smoothness. Also, monotonicity in the table values is usually maintained in the table model when using suitable local heuristics. If the heuristics always choose linear pieces, monotonicity preservation is guaranteed, even after application of the smoothing shell. In that case, the table models have some of the major attractive properties of tensor products of QVD splines, namely C^1 smoothness

and monotonicity preservation. On the other hand, suitable heuristics and primary functions can make the table models very compact, requiring few device measurements or device simulations, and little disk storage. Therefore, some finetuning will always be necessary to obtain the best compromise between conflicting model properties for a particular application. The modeling framework presented in this paper may aid in structuring this difficult work.

Appendix
Eigenfunctions of Table Models 1 and 2

Because the general proofs for the exact reconstructability of (2.11) and (2.13) are somewhat laborious in terms of the amount of calculus, only an outline of the major steps will be given. Basically, the proofs are obtained by substitution of (2.11) in (2.7), and (2.9) and (2.13) in (2.8). In the one-dimensional case, any one-dimensional function is an eigenfunction of $\hat{\gamma}_1$ and $\hat{\gamma}_2$, because all behavior occurs along a (single) gridline.

For $\hat{\gamma}_1$ induction on n can be applied. First the inductive proof is delivered for a single term from G_1 in (2.11). The vector τ as an index to g is then irrelevant for this part of the proof, because τ is just a uniqueness identifier for g in the case of multiple terms. Only the effect of τ on the form of a G_1-term needs to be considered. For a term containing one of the factors $g_\tau^{(1)}(x_1), \cdots, g_\tau^{(m)}(x_m)$, the equation obtained by substitution for $n = m$ can be separated from the equation resulting for $n = m+1$. Then, by using the induction proposition that this term is an eigenfunction for $n = m$, it follows that a term containing the same factor is an eigenfunction for all $n \geqslant m$. This means that if such a term is an eigenfunction for a particular dimension n, it automatically becomes an eigenfunction for all higher dimensions n as well. To prove the induction proposition, it is sufficient to prove that terms containing one of the factors $g_\tau^{(1)}(x_1), \cdots, g_\tau^{(m-1)}(x_{m-1})$ are eigenfunctions for $n = m-1$ (and hence, for $n = m$ and higher), *and* to prove that a term containing $g_\tau^{(m)}(x_m)$ is an eigenfunction for $n = m$. Continuing this partitioning recursively, it becomes sufficient to prove that terms containing one of the factors $g_\tau^{(1)}(x_1), \cdots, g_\tau^{(m)}(x_m)$ are eigenfunctions for $n = 1, \cdots, m$, respectively. This can be proven for any m without induction. Finally it is shown by substitution that any linear combination of eigenfunctions of $\hat{\gamma}_1$ is again an eigenfunction of $\hat{\gamma}_1$, thus concluding the proof that G_1 describes eigenfunctions of $\hat{\gamma}_1$.

For $\hat{\gamma}_2$ it can be verified by substitution that the operation of $\hat{\gamma}_2$ on G_2 in (2.13) yields a function of the same form as G_2, but with potentially different coefficients c'_τ. However, both expressions must interpolate the 2^n table points located at the vertices of a hyperrectangle, because $\hat{\gamma}_2$ does not modify the exact descriptions along the gridlines. Therefore, the 2^n coefficients become the unknowns of 2^n equations for interpolation, giving a unique solution (meaning $c'_\tau = c_\tau$) if the determinant of the set of equations is nonzero. So it just has to be shown that (2.10) is sufficient to obtain a nonzero determinant. The conditions (2.10) imply

$$g^{(k)}(x_{k,i_k}) \neq g^{(k)}(x_{k,i_k+1}). \tag{A1}$$

Introducing the notation:

$$a_{k,0} \triangleq g^{(k)}(x_{k,i_k});\ a_{k,1} \triangleq g^{(k)}(x_{k,i_k+1}) \tag{A2}$$

the determinant of size $2^n \times 2^n$ becomes

$$\begin{vmatrix} \prod_{k=1}^{n} a_{k,\tau'_k}^{\tau_k}\Big|_{\substack{\tau=0\\ \tau'=0}} & \cdots & \prod_{k=1}^{n} a_{k,\tau'_k}^{\tau_k}\Big|_{\substack{\tau=1\\ \tau'=0}} \\ \vdots & & \vdots \\ \prod_{k=1}^{n} a_{k,\tau'_k}^{\tau_k}\Big|_{\substack{\tau=0\\ \tau'=1}} & \cdots & \prod_{k=1}^{n} a_{k,\tau'_k}^{\tau_k}\Big|_{\substack{\tau=1\\ \tau'=1}} \end{vmatrix}$$

$$= \det\left[\bigotimes_{k=1}^{n}\begin{pmatrix} 1 & a_{k,0} \\ 1 & a_{k,1} \end{pmatrix}\right] = \prod_{k=1}^{n}(a_{k,1} - a_{k,0})^{2^{n-1}} \tag{A3}$$

which is obviously nonzero under conditions (A1). A treatment of Kronecker products can be found in [14].

References

[1] Ph. E. Allen and K. S. Yoon, "A table look-up MOSFET model for analog applications," in *Proc. Int. Conf. Computer-Aided Design*, Santa Clara, CA, pp. 124–127, Nov. 1988.

[2] J. A. Barby, "Multidimensional splines for modeling FET nonlinearities," Thesis UW/ICR 86-01, Univ. of Waterloo, Canada, 1986.

[3] J. A. Barby, J. Vlach, and K. Singhal, "Polynomial splines for MOSFET model approximation," *IEEE Trans. Computer-Aided Design*, vol. 7, pp. 557–566, May 1988.

[4] G. Bischoff and J. P. Krusius, "Technology independent device modeling for simulation of integrated circuits for FET technologies," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 99–109, Jan. 1985.

[5] C. de Boor, *A Practical Guide to Splines*. Berlin, Germany: Springer-Verlag, 1978.

[6] J. L. Burns, "Empirical MOSFET models for circuit simulation," Mem. No. UCB/ERL M84/43, Univ. Calif. Berkeley, May 1984.

[7] L. O. Chua and A.-C. Deng, "Canonical piecewise-linear modeling," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 511–525, May 1986.

[8] W. M. Coughran, E. Grosse and D. J. Rose, "Variation diminishing splines in simulation," *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 696–705, Apr. 1986.

[9] D. Divekar, D. Ryan, J. Chan, and J. Deutsch, "Fast and accurate table look-up MOSFET model for circuit simulation," in *Proc. Custom Integrated Circuits Conf.*, Rochester, NY, pp. 621–623, May 1986.

[10] J. J. Ebers and J. L. Moll, "Large-signal behavior of junction transistors," *Proc. IRE*, vol. 42, pp. 1761–1772, Dec. 1954.

[11] A. R. Forrest, "On Coons and other methods for the representation of curved surfaces," *Comput. Graph. Image Proc.*, vol. 1, pp. 341–359, 1972.

[12] J. L. Huertas and A. Rueda, "Sectionwise piecewise polynomial functions: Applications to the analysis and synthesis of nonlinear n-Port networks," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 897–906, Oct. 1984.

[13] Y.-H. Jun and S.-B. Park, "Piecewise polynomial models for MOSFET dc characteristics with continuous first order derivative," in *Proc. Int. Symp. on Circuits and Systems*, Espoo, Finland, pp. 2589–2592, June 1988.

[14] M. Marcus and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities* vol. 14, Boston, MA: Prindle, Weber, and Schmidt, 1964.

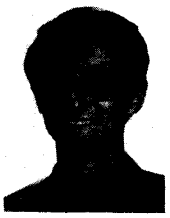
[15] P. B. L. Meijer, "Table models for device modelling," in *Proc. Int. Symp. on Circuits and Systems*, Espoo, Finland, pp. 2593–2596, June 1988.

[16] K. J. Narendra, D. Agnew and M. S. Nakhla, "Two-dimensional table look-up MOSFET model," in *Proc. Int. Conf. on Computer-Aided Design*, Santa Clara, CA, pp. 201–203, Sept. 1983.

[17] K.-G. Rauh, "A table model for circuit simulation," in *Proc. Twelfth European Solid-State Circuits Conf.*, Delft, The Netherlands, pp. 211–213, 1986.

[18] A. R. Rofougaran, B. Furman, and A. A. Abidi, "Accurate analog modeling of short channel FETs based on table lookup," in *Proc. Int. Symp. on Circuits and Systems*, Espoo, Finland, pp. 413–416, June 1988.

[19] M. F. Sevat, "GLASMOST: A MOSFET model of high numerical quality," in *Proc. Int. Symp. on Circuits and Systems*, Espoo, Finland, pp. 2597–2600, June 1988.
[20] T. Shima, "Device and circuit simulator integration techniques," in W. L. Engl, ed., *Process and Device Modeling*. Amsterdam, The Netherlands: North-Holland, 1986, ch. 14, pp. 433–459.
[21] P. Subramaniam, "Table models for timing simulation," *Proc. Custom Integrated Circuits Conf.*, Rochester, NY, pp. 310–314, May 1984.
[22] G. T. Wright, "Physical and CAD models for the implanted-channel VLSI MOSFET," *IEEE Trans. Electron Devices*, vol. ED-34, pp. 823–833, Apr. 1987.



Peter B. L. Meijer received the master's degree in physics from the Delft University of Technology, Delft, The Netherlands, in 1985.

Since 1985, he has been with the group CAD for VLSI Circuits at the Philips Research Laboratories in Eindhoven. His current research interests include device and circuit modeling, deterministic chaos, neural networks, and self-organizing systems.