Table Models for Device Modelling

Peter B. L. Meijer

Philips Research Laboratories
P.O. Box 80.000, 5600 JA Eindhoven, The Netherlands

Abstract - **Device models determine to a large extent the attainable quality of electronic circuit simulation. This paper describes a general approach to the automatic construction of C^1 multidimensional (multivariate) device models from table values. By using suitable heuristics, a compact and accurate C^1 table model for highly nonlinear multidimensional behaviour can be obtained. An example demonstrates the basic ideas.**

Introduction

Device models play a crucial role in electronic circuit simulation. They determine the attainable quality in the prediction and verification of the circuit behaviour for a given network. The demands placed upon the models are strong. One wants short model evaluation times, since nowadays a major fraction of VLSI simulation time is spent evaluating device models. The models must be accurate, because they limit the maximum simulation accuracy. The simulation algorithms (e.g. Newton-Raphson) may pose additional restrictions for rapid convergence, like continuity and monotonicity of the outcomes of the device models as a function of each of the controlling variables. Usually the continuity of the first partial derivatives is also required. Other restrictions follow from charge and energy conservation laws.

The classical approach has been to study the device physics, and to solve the corresponding device equations under simplifying assumptions about the boundary conditions. This physical modelling has the important advantage of a good predictive power for the effects of device modifications. However, for the ever smaller devices in VLSI, a simple geometrical subdivision into a few 1-dimensional subsystems treatable for analysis is no longer accurate. The divide-and-conquer approach starts to fail: e.g. multidimensional interactions, surface states, short-channel effects and narrow width effects can no longer be neglected nor be linearized. Furthermore, for more complicated models it becomes hard, and therefore very time-consuming, to match the subsystem descriptions in a way that fulfills the above-mentioned numerical requirements.

Therefore, some means of automatically generating suitable models for circuit simulation is desirable. The most obvious solution is to use table models that generate smooth curves from their input data. These input data can be obtained from measurements or device simulations.

Most table modelling methods known in the literature apply piecewise polynomial (pp) descriptions [1 – 8,10,12] . Especially the B-spline tensor product approach is becoming popular, partly due to the ease of constructing a smooth monotonic approximation. In spite of the excellent mathematical properties, this approach is still not well suited for accurate semiconductor device modelling purposes, because a strongly exponential behaviour in part of the operating regime is very common (e.g. for MOSFET's and bipolar devices). Accurate pp modelling of these exponential regions leads to excessive memory demands in the multidimensional case. The location of the transition between regions of qualitatively different behaviour is in general unknown and variable-dependent. This means that one should add to the table model at least a rudimentary intelligence that exploits some a priori knowledge about the device data and that applies selection criteria for choosing the most adequate local description. However, pushing the incorporated knowledge too far would again yield a very device dependent modelling scheme, which is one of the disadvantages of physical modelling.

In the following sections a general approach to the modelling of heavily nonlinear multidimensional functions is presented. With an example it is then shown that the approach can yield good approximations to physically meaningful behaviour.

Principles

For most operating points of a semiconductor device, only a few physical effects dominate the device behaviour. Furthermore, points with the same dominating physical effects tend to fall in clusters, i.e. in operating regions. In physical modelling one tries to determine the dominating physical effects and boundary conditions for a particular operating region, and tries to solve for explicit expressions describing the device behaviour, usually with the help of additional simplifications and assumptions. Quite often the modelling problem then reduces to the solution of a set of simple differential equations. Now we reverse this approach. The most frequent (idealized) semiconductor configurations constitute a problem space, from which a limited number of characteristic behavioural expressions arises. We expect that for a non-ideal device the local behaviour can still be adequately described by these expressions. They will be called primary functions, since they constitute a basis for the models to be constructed. We just have to choose the most appropriate primary function (or a combination of them) for a given set of table points. After determining the coefficients of the primary function chosen, we have a local description of the device behaviour. The local descriptions are then somehow put together to obtain a sufficiently smooth result. The choice of the most adequate primary function is made by heuristics, based upon general knowledge of semiconductor physics and a local set of table points.

Methods

We construct an interpolating table model for an n-dimensional table ($n > 1$) defined on a rectangular grid Γ_N defined by

$$\Gamma_{\underline{N}} \triangleq \{ (x_{1,i_1}, \ldots, x_{n,i_n}) \mid k = 1, \ldots, n : i_k = 1, \ldots, N_k \wedge x_{k,1} < \cdots < x_{k,N_k} \}$$

Here x_{k,i_k} is the i_k-th discrete position in the k-th direction. The n-dimensional table has a value $F_{i_1 \ldots i_n}$ associated with each of the grid points $(x_{1,i_1}, \ldots, x_{n,i_n})$.

We decompose the n-dimensional interpolation $F(x_1, \ldots, x_n)$ into a set of 1-dimensional interpolations I_j. I_j gives an interpolated value at x_j using a 1-dimensional table containing N_j points:

$$F(x_1, \ldots, x_n) \triangleq I_n(x_n, \{ (x_{n,i_n}, F_{i_n}(x_1, \ldots, x_{n-1})) \mid i_n = 1, \ldots, N_n \})$$

$$F_{i_{k+1} \ldots i_n}(x_1, \ldots, x_k) \underset{1<k<n}{\triangleq} I_k(x_k, \{ (x_{k,i_k}, F_{i_k \ldots i_n}(x_1, \ldots, x_{k-1})) \mid i_k = 1, \ldots, N_k \})$$

$$F_{i_2 \ldots i_n}(x_1) \triangleq I_1(x_1, \{ (x_{1,i_1}, F_{i_1 \ldots i_n}) \mid i_1 = 1, \ldots, N_1 \})$$

This scheme uses real table points to obtain 1-dimensional interpolations in the x_1-direction. The results act as table points for the 1-dimensional interpolations in the next direction x_2 [11]. We will call such auxiliary table points virtual table points. If the 1-dimensional interpolations in this scheme are exact, i.e. if they match the behaviour of the device exactly, then the n-dimensional interpolation is also exact. This is because the n-dimensional interpolation consists only of 1-dimensional interpolations. So this decomposition does not restrict the attainable accuracy in n-dimensional modelling.

Notice that all interpolations at any particular level in the recursion can be performed in parallel, thus leading to O(n) evaluation times on sufficiently parallel computer architectures (but then the amount of hardware needed grows exponentially with n).

The partial derivatives are given by

$$\frac{\partial F(x_1, \ldots, x_n)}{\partial x_n} = \frac{\partial I_n}{\partial x_n}$$

$$\frac{\partial F(x_1, \ldots, x_n)}{\partial x_k} \overset{k \neq n}{=} \sum_{i_{k+1}=1}^{N_{k+1}} \cdots \sum_{i_n=1}^{N_n} \frac{\partial I_k}{\partial x_k} \cdot \prod_{j=k+1}^{n} \frac{\partial I_j}{\partial F_{i_j \ldots i_n}}$$

To obtain a C^1 result, it is therefore sufficient to make $I_1, \ldots, I_n$ C^1 functions of $x_1, \ldots, x_n$ respectively, and $I_2, \ldots, I_n$ C^1 functions of the virtual table values in their argument lists.

Now we decompose the general 1-dimensional interpolations into a set of piecewise functions. This allows us to apply local selection of the most adequate description, and the resulting local description can probably be simple, because it only has to model a relatively small region. This helps to achieve a computationally efficient model.

Let for $x_1 \leq x \leq x_N$

$$I(x, \{ (x_i, F_i) \mid i = 1, \ldots, N \}) \triangleq \sum_{i=1}^{N} h_i(x).f_i(x)$$

with

$$f_1(x) \triangleq f_2(x) \quad ; \quad f_N(x) \triangleq f_{N-1}(x)$$

$$f_i(x) \underset{1<i<N}{\triangleq} \text{a } C^1 \text{ function on } [x_{i-1}, x_{i+1}] \text{ interpolating } (x_{i-1}, F_{i-1}), (x_i, F_i) \text{ and } (x_{i+1}, F_{i+1})$$

$$h_i(x) \triangleq \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in [x_{i-1}, x_i], \quad i = 2, \ldots, N \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}], \quad i = 1, \ldots, N\text{-}1 \\ 0 & \text{otherwise} \end{cases}$$

The hat functions $h_i(x)$ induce continuity of the derivative with respect to x, provided each piecewise function $f_i(x)$ is itselve C^1 within the range of non-zero values of its corresponding hat function, and interpolates its three supporting (virtual) table points. Furthermore, the hat functions do not influence the interpolation result if neighbouring (overlapping) piecewise functions are the same, e.g. when exact, at their overlap. So this decomposition does not restrict the attainable accuracy in the 1-dimensional modelling of C^1 behaviour. This also means that all C^1 n-dimensional models (e.g. C^1 tensor products of B-splines) are covered by the present level of decomposition into C^1 1-dimensional piecewise functions, provided the table has been generated by these models. Interpolation is general as long as we do not specify where the table to be interpolated comes from. It may have been generated by an approximation scheme that itself used another table to approximate. We need not exclude the possibility of modifying the original set of table points.

So far we have an n-dimensional description that is guaranteed C^1 without having specified the precise shape of the C^1 piecewise functions. The large freedom of choice in selecting the piecewise functions allows us to apply many physically meaningful heuristics. One such a heuristic could be, that if we know that the device behaviour is strictly monotonic as a function of each of the controlling variables, we should use a (strictly monotonic) exponential interpolation when a polynomial interpolation would yield non-monotonic results. The physical motivation lies in the fact that a sudden 'flattening' of the semiconductor device characteristic as a function of a controlling voltage corresponds to some kind of saturation or pinch-off, which is often related to a depletion of charge carriers somewhere in the device. The remaining carriers are generated by processes governed by Boltzmann statistics, which lead to exponential dependencies on the controlling voltages. In the following example we illustrate the results obtained using this simple heuristic, that chooses the most adequate piecewise function using a basis consisting of only two primary functions, a polynomial function and an exponential function.

Example

We consider an example that illustrates the concepts of semi-physical table modelling. We generate a table with data from a simple physical MOST model, and compare the results of the above interpolation scheme with the original physical model. The drain-to-source current behaviour of the 2-dimensional C^2 physical model GLASMOST [9] is used as the reference model, with a

threshold of 1V and a thermal voltage of 30mV. $\mu C_{ox} W/L = 2.789 \times 10^{-4}\ AV^{-2}$. 16 table points per dimension are used, at voltages -3, -2, -1, 0, 0.5, 0.75, 0.875, 1, 1.125, 1.25, 1.5, 2, 3, 4, 5 and 6V. The non-equidistant points give some extra weight to the transition region near the 1V threshold. For simplicity of discussion, we assume that real and virtual table points are strictly monotonic as a function of the controlling variables. (In general, monotonicity of the virtual table points is not guaranteed and needs special treatment, but in this example the monotonicity assumption can be made valid.) The heuristic chooses whether to use a polynomial or an exponential interpolation based upon a measure of the curvature κ

$$\kappa_i \triangleq \frac{F_{i+1} - F_i}{F_i - F_{i-1}} \cdot \frac{x_i - x_{i-1}}{x_{i+1} - x_i} > 0$$

Let

$$f_i(x) \triangleq w_i(\kappa_i)\, P_i(x) + (1 - w_i(\kappa_i))\, E_i(x)$$

$$P_i(x) \triangleq a_{0i} + a_{1i}\, x + a_{2i}\, x^2$$

$$E_i(x) \triangleq b_{0i} + b_{1i} \cdot b_{2i}^{x}$$

with $P_i(x)$ and $E_i(x)$ each interpolating (x_{i-1}, F_{i-1}), (x_i, F_i) and (x_{i+1}, F_{i+1}). $w_i(\kappa_i)$ is a C^1 pp weight function of κ_i. $w_i(\kappa_i) = 0$ if the extremal of $P_i(x)$ lies in the range $[x_{i-1}, x_{i+1}]$. $w_i(\kappa_i) = 1$ if κ_i is close to 1.

The results are shown in figures 1-3. Figure 1 shows the monotonic interpolation of table points representing the drain-to-source current as a function of gate-to-drain voltage and gate-to-source voltage. Figure 2 shows the absolute error normalized to the largest current in this range. Figure 3 shows the true relative error, but to avoid dividing by zero the relative error was artificially forced to zero near $V_{ds} = 0$ (within 1mV). The interpolation causes zero-valued errors at the table points. In most of each of the four operating regions (the subthreshold, linear, drain saturation and source saturation region) the errors are also zero. This is because the decomposed 1-dimensional behaviour of GLASMOST is always one of the two primary functions our heuristic can select. Only at the transition between different operating regions the heuristic is unable to make the right choice, which causes non-zero errors. Still, the normalized absolute error stays under 0.02%. Even the relative error stays under 50%, whereas a B-spline approach would yield huge relative errors. Of course this is an idealized example, but it illustrates that the scheme applies quite well to physically sensible descriptions.

Conclusions

A general C^1 decomposition into 1-dimensional piecewise functions can be advantageous in modelling heavily nonlinear multidimensional behaviour. It opens the way for heuristic rule-based modelling to bring physical knowledge into table modelling. Thus it becomes possible to achieve greater accuracy with less table points. However, the evaluation times on sequential machines can become large due to the $2(2^{2n} - 1)/3$ piecewise functions $f_i(x)$ involved in a model evaluation. Transitions in qualitative behaviour are detected automatically by the heuristics, and although monotonicity is not guaranteed, suitable heuristics can generate monotonic results in most practical cases. Furthermore, due to the generality of the approach, we can see it as a first step towards the automatic construction of very compact physical models. If neighbouring local descriptions are almost the same, then a single compact description may adequately model the whole or a major part of an operating region. In such a case we only need to save a few coefficients for the description of the behaviour in this region and for a description of the position and shape of the region. The heuristics used for primary function selection may also be applied as an analytical tool to study the behaviour of a device, because they indicate in which region which physical effect dominates.

References

[1] Barby, J.A., Multidimensional Splines for Modeling FET Nonlinearities, Thesis UW/ICR 86-01, University of Waterloo, Canada (1986).

[2] Bischoff, G. and Krusius, J.P., Technology Independent Device Modeling for Simulation of Integrated Circuits for FET Technologies, IEEE Trans. on Computer-Aided Design 4 (1985), 99-109.

[3] Boor, C. de, A Practical Guide to Splines, Springer-Verlag, Berlin (1978).

[4] Burns, J. L., Empirical MOSFET Models for Circuit Simulation, Memorandum No. UCB/ERL M84/43, University of California, Berkeley (1984).

[5] Coughran, W. M., Grosse, E. and Rose, D. J., Variation Diminishing Splines in Simulation, SIAM J. Sci. Stat. Comput. 7 (1986), 696-705.

[6] Divekar, D., Ryan, D., Chan J. and Deutsch, J., Fast and Accurate Table Look-Up MOSFET Model for Circuit Simulation, Custom Integrated Circuits Conference, IEEE (1986), 621-623.

[7] Narendra K. J., Agnew D. and Nakhla, M. S., Two-Dimensional Table Look-Up MOSFET Model, Int. Conf. on Computer-Aided Design, IEEE (1983), 201-203.

[8] Rauh, K. -G., A Table Model for Circuit Simulation, Twelfth Europian Solid-State Circuits Conference (1986), 211-213.

[9] Sevat, M. F., GLASMOST: A MOSFET Model of High Numerical Quality, Int. Symp. on Circuits and Systems, IEEE (1988).

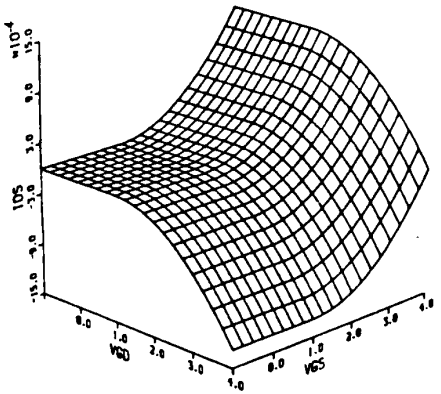
[10] Shima, T., Device and Circuit Simulator Integration Techniques, Chapter 14 in Engl, W. L. (ed.), Process and Device Modeling, North-Holland (1986), 433-459.

[11] Steffensen, J. F., Interpolation, Chelsea Publishing Company (1950), 203-204.

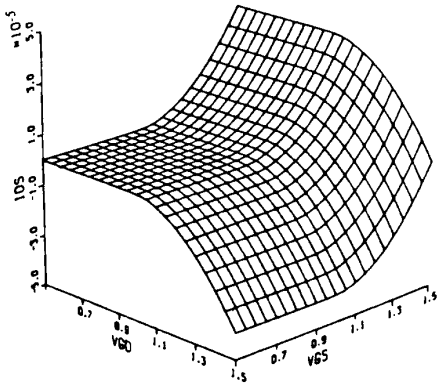
[12] Subramaniam, P., Table Models for Timing Simulation, Custom Integrated Circuits Conference, IEEE (1984), 310-314.

Figure 1. Table model for $I_{ds}(V_{gd}, V_{gs})$

$IDS = I_{interpolation}$



$V_{gd}, V_{gs} \in [-1, 4]$ V

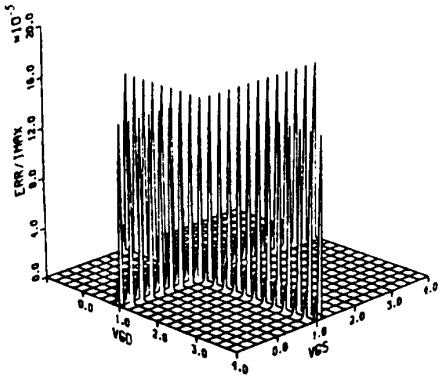


Enlargement: $V_{gd}, V_{gs} \in [0.5, 1.5]$ V

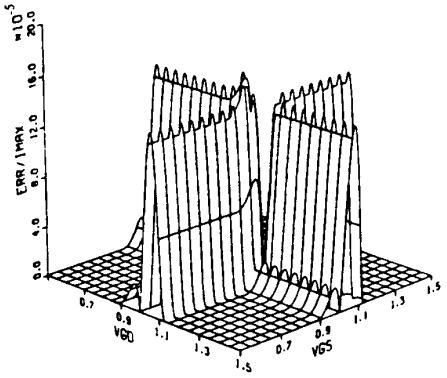
Figure 2. Normalized absolute error:

$$ERR/IMAX = \left| \frac{I_{interpolation} - I_{GLASMOST}}{I_{maximum}} \right|$$

with $I_{maximum} = 12.8 \times 10^{-4}$ A



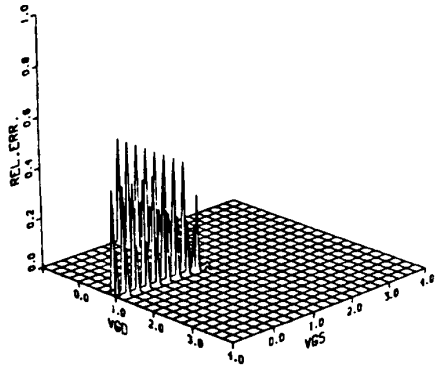
$V_{gd}, V_{gs} \in [-1, 4]$ V



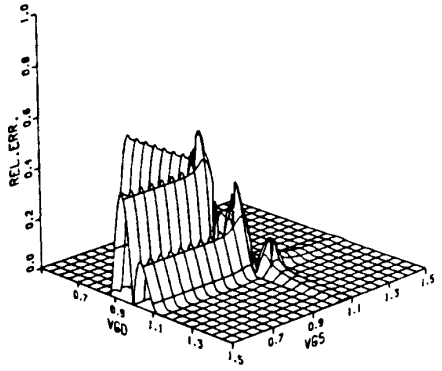
Enlargement: $V_{gd}, V_{gs} \in [0.5, 1.5]$ V

Figure 3. Relative error:

$$REL.ERR. = \left| \frac{I_{interpolation} - I_{GLASMOST}}{I_{GLASMOST}} \right|$$



$V_{gd}, V_{gs} \in [-1, 4]$ V



Enlargement: $V_{gd}, V_{gs} \in [0.5, 1.5]$ V