

INTERPOLATION METHODS IN MODELLING SEMICONDUCTOR DEVICES BY ENHANCED CONTROLLED SOURCES

W.M. Zuberek

Department of Computer Science, Memorial University
St. John's, NL, Canada A1C 5S7

Abstract

Numerical modelling of semiconductor devices, based on detailed descriptions of geometries and electrical properties of materials, is often the most time-consuming part of circuit simulations. To reduce the amount of required calculations, table-driven methods are proposed in which semiconductor devices are represented by (nonlinear) dependent voltage/current sources $Q_{driven}U$ by (multidimensional) arrays of data obtained from measurements, evaluations, or more detailed simulations. The actual values of voltages and currents are determined by interpolation of these discrete data. A comparison of several interpolation methods is illustrated by numerical results obtained from the SPICE-PAC package of simulation routines.

1. INTRODUCTION

Analog circuits typically have fewer components than digital ones, but their performance must be analyzed in greater detail and more accurately. Computer-aided circuit analysis or circuit simulation offers circuit designers the capability to analyze both the functionality and the performance of a design before the expensive and time-consuming fabrication takes place. The SPICE-2G program [6,12,13,16] developed at the University of California at Berkeley, has become one of the most popular "second-generation" circuit simulators.

"Second-generation" circuit simulators are usually general-purpose, batch-oriented programs with "closed" sets of operations and "static" definitions of analyses and parameters [4,14]. This is too restrictive in applications based on repeated circuit analyses, such as interactive simulation or circuit optimization [9,17]. In such cases, a more flexible structure for the simulation program is needed in which different analyses can be performed "on demand", and which provides an access to internal representation of circuit elements in order to modify their values during simulation sessions. The simulator should have a structure of a collection (or a package) of subroutines rather than a program with one, fixed sequence of analyses. SPICE-PAC [17,18] is such a package of simulation subroutines; it uses the same numerical algorithms which have been implemented in SPICE, but its internal structure is redesigned in order to provide a more flexible (user) interface, with better utilization of existing simulation capabilities.

SPICE-PAC, in a way similar to the SPICE program, provides a "closed" set of (parameterized) circuit elements and semiconductor devices. In some cases such a solution is unsatisfactory, and a number of extensions has been provided to enhance the original SPICE facilities [18,19]. For example, it has been observed that most of the time spent on circuit simulation is consumed by evaluations of device models [8] which is attributed to nonlinearities and complicated physical phenomena of modern semiconductor devices, especially those with submicron dimensions [9]. Consequently, even speeding-up the basic numerical algorithms (e.g., by relaxation techniques [14]) cannot reduce the complexity of model evaluations. Several approaches have been used for approximations of device models (e.g., table look-up [3,5,15], spline functions [2,3,7]). Look-up tables generally have data stored in linearly increasing values of the independent variable(s), and then the value of the function can be obtained by simple mappings of independent values into the corresponding indices of the table. Different interpolation schemes can be used to increase accuracy of table-driven approximations [1,10,11,19].

Table-driven approximations of semiconductor devices could be implemented by dependent voltage and current sources (supported by SPICE-like simulators) provided there is a straightforward access to tables of discrete data values and a flexible, user-defined mapping of controlling voltages or currents into the "positions" of controlled data. Also, data interpolation methods and "smoothing" techniques should be available for potential refinements, modifications or extensions. Therefore, an interface has been built into SPICE-PAC to allow external specifications of voltage and current "controlling" functions (SPICE defines linear and nonlinear dependent voltage and current sources, but the only nonlinear function which is supported by SPICE is a polynomial). It is assumed that the external (evaluating) function is in the form of a user-defined subroutine (or a group of subroutines organized in a hierarchical way) which evaluates the controlled voltage or current and its derivative(s) for given set of controlling variables. The subroutine is called by the package for each of the indicated dependent sources when the controlled value is needed. Since the subroutine is supplied by the user, the characteristics of dependent voltages and/or currents can be "shaped" rather arbitrarily. It should be noticed that this facility also provides

an opportunity for SPICE-PAC to simulate devices for which models do not presently exist in SPICE; the voltage and/or current characteristics of new devices can be represented by externally specified functions (in analytic or tabular form).

Implementation of such "enhanced" controlled sources requires a number of additional parameters and options which do not exist in the original SPICE input language. Also, enhanced controlled sources are just one example of enhanced circuit simulation, i.e., an "open" simulation environment in which users can replace, enhance or modify existing simulation capabilities, as required by their applications.

2. ENHANCED CONTROLLED SOURCES

The SPICE-like programs support four types of dependent sources (or transducers) characterized by the following equations

$$\mathbf{v}=\mathbf{e}(\mathbf{v}), \mathbf{i}=\mathbf{f}(\mathbf{i}), \mathbf{i}=\mathbf{g}(\mathbf{v}), \mathbf{v}=\mathbf{h}(\mathbf{i})$$

where the functions "e", "f", "g" and "h" must be polynomials, the arguments "v"R and "i" may be multidimensional, and the polynomial functions are specified by a set of coefficients "p0", "p1", ..., "pn" [16]. The general form of SPICE dependent source description is

```
Zname node+ node- POLY(narg) arg1,arg2,...
+ p0,p1,p2,... IC=...
```

where "Zname" is the name of a source, "node+" and "node-" are the positive and negative nodes of the dependent source, "POLY(narg)" must be used for sources with more than one argument (otherwise it is optional), and "narg" is the number of arguments; each argument "arg1", "arg2" ..., is either an independent voltage source name "Vname" (for current controlled sources) or a pair of node numbers "vnode+ vnode-" (for voltage controlled sources), the sequence "p0", "p1" ..., represents consecutive polynomial coefficients, and the last part is an optional specification of initial conditions "IC=...".

Enhanced dependent sources, i.e., sources with "external" functions, can be described in one of two following forms:

```
Zname node+ node- FUN(idf) ARG(narg) arg1,arg2,...
+ DIM(d1,d2,...) p0,p1,... IC=...
```

or

```
Zname node+ node- FUN(idf) ARG(narg) arg1,arg2,...
+ USE(tabname) IC=...
```

where:

"FUN(idf)" defines a numerical identifier "idf" which is passed to the function evaluation routine; usually "idf" denotes the "method" of evaluation, e.g., "idf=1" may indicate unidimensional linear interpolation, "idf=2" unidimensional Lagrange interpolation, etc.; it should be noticed that the interpretation and use of "idf" is completely determined by user's routine(s);

"ARG(narg)" specifies the number of arguments "narg" (with the default value equal of 1); this parameter is mandatory if the number of arguments is greater than 1;

"DIM(d1,d2,...)" is an optional description of the "structure" of coefficients "p0,p1,p2,..."; if the coefficients describe a multidimensional array of data, the dimensions of this array may be indicated in the "DIM" option; when this option is used, it must contain "narg" positive integer values "d1,d2,...,dnarg" which are the consecutive dimensions of the array of coefficients, and then the number of coefficients "p0,p1,p2,..." must be equal to $(d1+d2+\dots+dnarg+d1*d2*\dots*dnarg)$, i.e., "narg" vectors of independent variable values (or vectors of controlling values) and the corresponding array of (dependent) data; the ordering of coefficients must be consistent with their uses since SPICE-PAC simply enters consecutive coefficients into consecutive elements of an internal vector, and then passes this vector to the (external) function evaluation routine;

"USE(tabname)" is a reference to a "TABLE" of shared coefficients which is identified by the name "tabname"; "TABLE" pseudo-element is a new extension of the SPICE-PAC package, introduced in order to use the same coefficients by a number of enhanced controlled sources (in this sense "TABLEs" are similar to "MODELs" for semiconductor devices).

The remaining elements are as before. The list of coefficients "p0,p1,..." may thus be used to specify arbitrary parameters which are passed to the user-defined function evaluation routine together with unique —identification" information of the controlled source.

"QFUN(idf)" and "ARG(narg)" are optional parts but at least one of them must be present in the description of enhanced dependent sources.

Example. Semiconductor diode voltage/current characteristic is approximated by a voltage controlled current source (with node numbers "n1" and "n2") "driven" by a sequence of 20 data points (data points are pairs of j diode voltage, diode current j values with increasing values of diode voltage; it should be noted that parentheses may be used rather arbitrarily for "grouping" the data since they are simply ignored by the SPICE-PAC's input processor):

```
GDIODE1 n1 n2 FUN(1) ARG(1) n1 n2 (-10 -1.0E-11,
+ 0.00 0.000000, 0.05 1.09E-13, 0.10 5.68E-13,
+ 0.15 3.44E-12, 0.20 2.30E-11, 0.25 1.58E-10,
+ 0.30 1.09E-09, 0.35 7.53E-09, 0.40 5.21E-08,
+ 0.45 3.60E-07, 0.50 2.49E-06, 0.55 1.72E-05,
+ 0.60 1.19E-04, 0.65 8.21E-04, 0.70 5.67E-03,
+ 0.75 3.92E-02, 0.80 2.71E-01, 0.85 1.87E+00,
+ 0.90 1.29E+01)
```

The use of "ARG(1)" is optional since the default number of arguments is one. Moreover, it is assumed that an external function identified by "idf=1" performs interpolation of these data, while the organization of data corresponds to a matrix 2 by 20.

If the values of diode voltages and currents are to be used as separate vectors, the same coefficients should be arranged in a different way:

```
GDIODE2 n1 n2 FUN(2) ARG(1) n1 n2 DIM(20)
+ (-10,0,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.40,
+ 0.45,0.50,0.55,0.60,0.65,0.70,0.75,0.80,0.85,
+ 0.90)
+ (-1E-11,0,1.09E-13,5.6E-13,3.44E-12,2.30E-11,
+ 1.58E-10,1.09E-09,7.53E-09,5.21E-08,3.60E-07,
+ 2.49E-06,1.72E-05,1.19E-04,8.21E-04,5.67E-03,
+ 3.92E-02,0.271,1.87,12.9)
```

The use of "DIM" is optional, but when specified it must indicate one argument (diode voltage). Also, it is assumed that this representation is processed by an external function determined by "idf=2". □

Quite often several enhanced dependent sources use the same coefficients for the evaluation function. In such cases it is convenient (and recommended) to define the coefficients as an independent "TABLE" pseudo-element, indicated in dependent source descriptions by the "USE(tabname)" section.

The general form of "TABLE" parameters is

```
.TABLE tabname ARG(narg) DIM(d1,d2,...) p0,p1,p2,...
```

where "tabname" is a (unique) name of this pseudo-element, and the remaining elements are identical as for specification of dependent sources. The number of arguments, "narg", in a "TABLE" parameters (default or explicit) must be equal to the number of arguments in dependent sources indicating "tabname" in "USE(tabname)" sections.

Example. A transfer characteristic of a MOSFET inverter [9] can be approximated by a voltage controlled voltage source:

```
EMOSINV n1 n2 FUN(1) n3 n4 USE(TMOSINV)
```

with the following table of 26 data points:

```
.TABLE TMOSINV (0.0 5.00E+00, 0.5 5.00E+00,
+ 0.8 5.0-E+00, 0.9 4.96E+00, 1.0 4.86E+00,
+ 1.1 4.71E+00, 1.2 4.47E+00, 1.3 4.10E+00,
+ 1.4 3.13E+00, 1.5 1.66E+00, 1.6 5.07E-01,
+ 1.7 3.60E-01, 1.8 2.96E-01, 1.9 2.56E-01,
+ 2.0 2.27E-01, 2.1 2.05E-01, 2.2 1.87E-01,
+ 2.3 1.73E-01, 2.4 1.61E-01, 2.5 1.51E-01,
+ 2.7 1.34E-01, 3.0 1.17E-01, 3.5 9.66E-02,
+ 4.0 8.36E-02, 4.5 7.43E-02, 5.0 6.74E-02)
```

The same data are used in one of the following examples. □bu

3. UNIVARIATE INTERPOLATION

Functions that are especially attractive for approximation, are polynomials because they can be evaluated, differentiated and integrated easily using just the basic arithmetic operations of addition, subtraction and multiplication. The simplest case of piecewise linear univariate approximation is known as "Qbroken line" approximation. However, broken lines are neither smooth nor very efficient approximators. Piecewise polynomials with higher order pieces must be used for a smoother approximation. It has been found that the simple schemes, i.e., quadratic and cubic, yield good results [8]. Higher order polynomials tend to produce "wiggles" and "oscillations" [7]. Akima's cubic interpolation [1] was developed in order to remove such wiggles in the interpolant. In local approximation methods, the "i"-th piece depends only on information from, or near the approximated interval. Usually, local interpolants do not guarantee continuous derivatives, but they can be evaluated directly from the data (and the data can be dynamically refined if better accuracy is needed). Typical examples of global piecewise polynomial interpolants are splines and B-splines.

An excellent discussion of piecewise approximation is given in [7]. The IMSL Math Library [20] provides a consistent implementation of methods presented in [7].

In the first example, a diode voltage/current characteristic represented by a TABLE TAB is approximated by voltage controlled current sources with four different (external) interpolation methods, piecewise linear (or "broken line") interpolant (FUN(1)), piecewise quadratic (local) interpolant (FUN(2)), Akima cubic interpolant (FUN(5)), and B-spline (order 3) interpolant (FUN(6)). The results are compared with the original SPICE (default) diode model, and the differences V(3,2), V(4,2), etc., are shown together with the original diode output V(2).

It can be observed, that the local quadratic interpolant (V(4,2)) is as good as the Akima interpolant (V(5,2)), while the B-spline (V(6,2)) is not better than the linear interpolant (V(3,2)).

The second example uses the same four interpolation methods for a DC transfer curve of a MOSFET inverter in which an enhancement MOSFET uses a depletion MOSFET as the load [9]. The transfer curve is represented by table-driven voltage controlled voltage sources with external interpolation routines. The interpolation results are compared with the "real" inverter implemented by level 3 SPICE models.

Again, the local quadratic interpolant (V(5,2)) is very similar to the Akima interpolant (V(8,2)). The B-spline interpolant (V(9,2)) is worse in the initial part of the curve. The linear interpolant (V(4,2)) is surprisingly good. Graphical representation of all four approximated transfer curves is shown in Fig.1; the differences between interpolants are clearly visible.

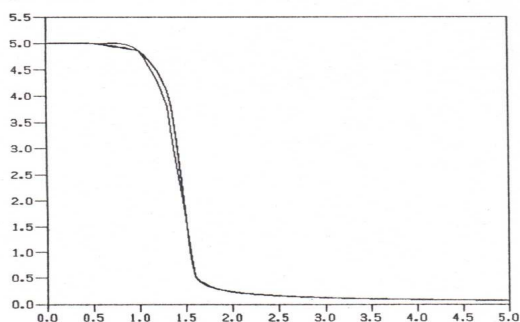


Fig.1. Approximated DC transfer curves.

4. MULTIVARIATE INTERPOLATION

The simplest method of obtaining multivariate interpolation and approximation is to take univariate methods and apply tensor products; many two- and three-dimensional spline interpolation routines in the IMSL Library use this approach [7,20]. For simple multivariate polynomials (e.g., quadratic and cubic), local methods interpolate the data with corresponding patches of surfaces, which - in general case - does not guarantee the continuity.

In the following example, the DC transfer curve analysis is performed for an enhancement MOSFET with a depletion MOSFET as the load. The depletion MOSFET is represented by a current controlled voltage source with a single voltage-current characteristic (.TABLE TABMOSD), while a two-argument voltage controlled current source QdrivenU by a rectangular table of data (.TABLE TABMOSH with dimensions 11 [gate-source voltages] times 13 [drain-source voltages]) is used to approximate the enhancement MOSFET. The original enhancement MOSFET characteristics are shown in Fig.2 and Fig.3.

The circuit is composed of a "reference" subcircuit which contains the original MOSFETs, and three sections with controlled sources approximating MOSFET's characteristics. One section uses linear interpolation of

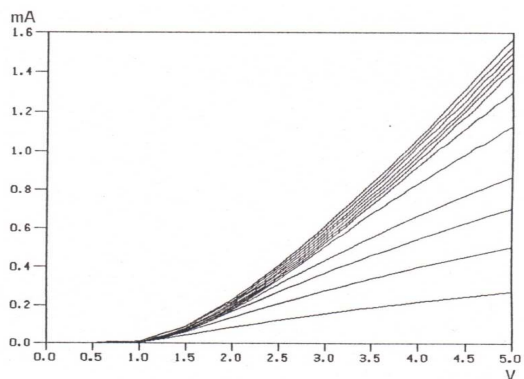


Fig.2. MOSFET drain-source voltage as parameter.

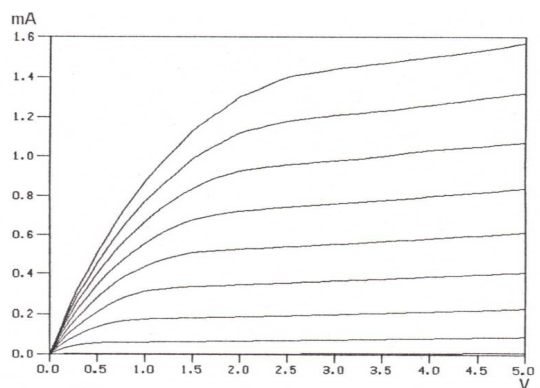


Fig.3. MOSFET gate-source voltage as parameter.

the data (FUN(3)), the second uses local quadratic two-dimensional interpolant (FUN(8)), while two-dimensional tensor-product B-spline interpolation is used in the third section (FUN(7)). The results show the original output voltage V(2), and differences between the original voltage and interpolated results (V(5,2), V(6,2) and V(7,2), respectively).

The results of local quadratic interpolation and tensor-product B-splines are very similar, and the differences seem to be rather insignificant. Linear interpolation is clearly inferior.

5. CONCLUDING REMARKS

Enhanced dependent voltage and current sources provide an efficient and flexible alternative to time-consuming evaluations of complex device models. Look-up tables of table-driven dependent sources may be generated directly from device measurements, fitting devices that do not closely match analytical models. Another benefit of look-up tables is that multidimensional tables can usually be represented by "irregular" data structures in which relatively small highly nonlinear regions are described by dense collections of data while more linear regions need only a few data samples. Consequently, multidimensional tables can be stored rather efficiently and still represent

the data with reasonable accuracy; optimized representations of piecewise interpolants with two or three quadratic pieces and three to five characteristic currents or voltages per device may be sufficient in many applications [2,8]. Moreover, for local methods, the data tables can be "dynamically refined" if more accurate approximations are needed.

An interface for external specification of dependent voltage and current source functions is an example of enhanced circuit simulation. Some further examples of such enhancements may include new models of semiconductor devices (e.g., GaAs models) "new" circuit elements (e.g., operational amplifiers or other "standard blocks") as well as new algorithms (e.g., relaxation methods).

Acknowledgement

This research was partially supported by Northern Telecom Canada through Memorial University Grant, and by Natural Sciences and Engineering Research Council of Canada through Operating Grant A8222.

REFERENCES

- [1] H. Akima, A new method of interpolation and smooth curve fitting based on local procedures; *Journal of the ACM*, vol.17, pp.589-602, 1970.
- [2] J. Barby, J. Vlach, K. Singhal, Optimized polynomial splines for FET models; *Proc. Int. Symp. on Circuits and Systems*, Montreal, Canada, pp.1159-1162, 1984.
- [3] J. Barby, J. Vlach, K. Singhal, Tableau formulation for optimized polynomial spline MOSFET model approximation; *Proc. Int. Symp. on Circuits and Systems*, San Jose CA, pp.1137-1138, 1986.
- [4] R. Beresford, J. Domitrowich, Survey of circuit simulators; *VLSI Systems Design*, vol.8, no.8, pp.70-81, 1987.
- [5] J.L. Burns, A.R. Newton, D.O. Pederson, Active device table look-up models for circuit simulation; *Proc. Int. Symp. on Circuits and Systems*, Newport Beach CA, pp.250-253, 1983.
- [6] E. Cohen, Program reference for SPICE 2; Memorandum UCB/ERL M592, University of California, Berkeley CA 94720, 1976.
- [7] C. de Boor, **A practical guide to splines**; Springer Verlag 1978.
- [8] U. Feldman, K-G. Rauh, K. Steger, Circuit simulation on vector processors; *Proc. European Conf. on VLSI in Computers*, Hamburg, West Germany, pp.246-249, 1987.
- [9] L.A. Glasser, D.W. Dobberpuhl, **The design and analysis of VLSI circuits**; Addison-Wesley 1985.
- [10] A.J. Macleod, A comparison of algorithms for polynomial interpolation; *Journal of Computational and Applied Mathematics*, vol.8, pp.275-277, 1982.
- [11] J.M. McNamee, Comparison of spline and Lagrangian interpolation; *Journal of Computational and Applied Mathematics*, vol.16, pp.237-240, 1986.
- [12] A.R. Newton, D.O. Pederson, A.L. Sangiovanni-Vincentelli, C.H. Sequin, Design aids for VLSI: the Berkeley perspective; *IEEE Trans. Circuits and Systems*, vol.28, no.7, pp.666-680, 1981.
- [13] D.O. Pederson, A historical review of circuit simulation; *IEEE Trans. Circuits and Systems*, vol.31, no.1, pp.103-111, 1984.
- [14] A.E. Ruehli, G.S. Ditlow, Circuit analysis, logic simulation, and design verification for VLSI; *Proc. of the IEEE*, vol.71, no.1, pp.34-48, 1983.
- [15] T. Shima, H. Yamada, R.L.M. Dang, Table look-up MOSFET modeling system using a 2-D device simulator and monotonic piecewise cubic interpolation; *IEEE Trans. CAD of Integrated Circuits and Systems*, vol.2, no.2, pp.121-126, 1983.
- [16] A. Vladimirescu, K. Zhang, A.R. Newton, D.O. Pederson, A.L. Sangiovanni-Vincentelli, SPICE Version 2G - User's Guide (10 Aug. 1981); Department of Electrical Engineering and Computer Sciences, University of California, Berkeley CA 94720, 1981.
- [17] W.M. Zuberek, SPICE-PAC, a package of subroutines for interactive simulation and optimization of circuits; *Proc. IEEE Int. Conf. on Computer Design*, Port Chester NY, pp.492-496, 1984.
- [18] W.M. Zuberek, P. Gillard, Enhanced circuit simulation in the SPICE-PAC package; *Proc. 28-th Midwest Symp. on Circuits and Systems*, Louisville KY, pp.182-185, 1985.
- [19] W.M. Zuberek, M.S. Zuberek, Enhanced dependent sources in the SPICE-PAC package of simulation subroutines; Technical Report #8701, Department of Computer Science, Memorial University of Newfoundland, St. John's, Newfoundland, Canada A1C 5S7, 1987.
- [20] IMSL Math/Library - FORTRAN Subroutines for Mathematical Applications, vol.2, chapter 3: Interpolation and Approximation; MAL-USM-UNBND-1.0, version 1.0, 1987.