# Approximation in VLSI Simulation

Eric Grosse
*AT&T Bell Laboratories*

## 1 Introduction

Over the past decade, one of the most important applications of numerical analysis to the electronics industry has been in VLSI device simulation. Numerical approximation in particular has played a valuable supporting role. Examples include the use of 1) tensor product variation diminishing splines for models of transistor charges and currents and 2) continuation to find a safe operating region avoiding avalanche breakdown. This paper looks at quadratic box splines in three variables for use in Monte Carlo solution of the Boltzman transport equation. The bivariate Zwart-Powell element does not directly generalize, but another particular box spline is constructed.

Section 2 reviews ways that approximation has been used to tie detailed device simulation to higher-level circuit simulation. Section 3 describes a particular form of device simulation, Monte Carlo solution of the Boltzman transport equation. Section 4 presents some initial explorations of the use of box splines, whose general theory has been extensively developed in the approximation literature of the past decade, in the particular case of three variables.

## 2 Approximation as a bridge between levels of simulation

In computational microelectronics as in many other areas of science, there are many levels of detail at which one can undertake numerical simulation. For example, device simulation studies behaviour inside a single transistor, whereas circuit simulation studies behaviour in a circuit of thousands of transistors. There is a need for the results of device simulation to be packaged into a compact model for use in circuit simulation. Traditionally this was done by analytic approximations, adorned by clever adjustment terms as more accuracy was needed. Building these models by hand for each new device or regime of operation requires too much manual effort, by people with specialized skills.

So an early use of approximation in VLSI simulation was tensor product splines. This allows new models to be automatically constructed [8]. In particular, we introduced the use of tensor quadratic splines on a uniform grid (for fast lookup) to model the current $I$ in a transistor as a function of the voltage differences between the source, drain, gate, and bulk contacts:

$$I = f(v_{ds}, v_{gs}, v_{bs}) + \frac{d}{dt} q(v_{ds}, v_{gs}). \tag{2.1}$$

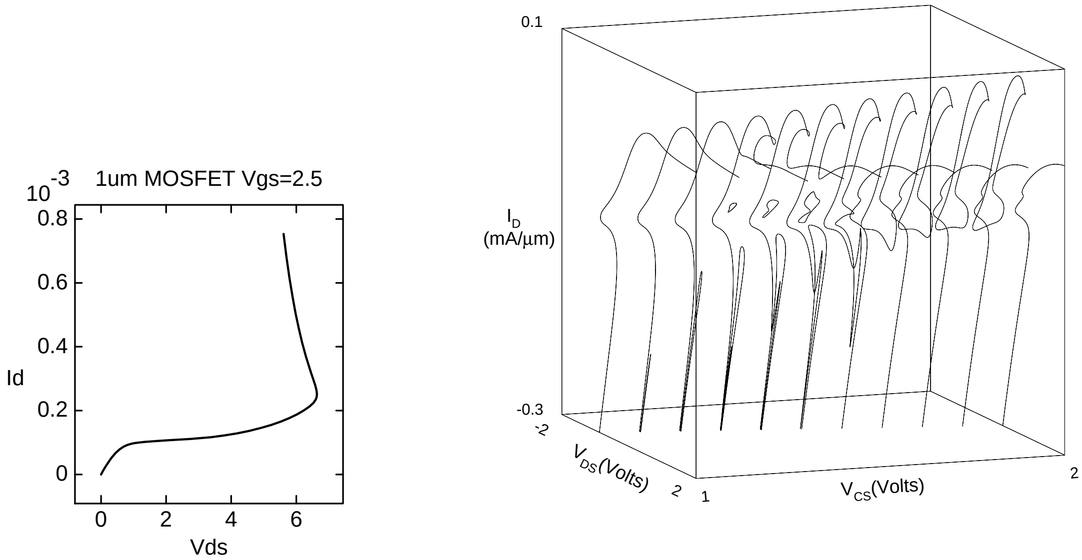


FIGURE 1. Current-voltage curves for MOSFET and CHINT transistors.

Here $v_{ds} = V_d - v_s$ is the voltage difference between the drain and the source contacts, $f$ is the steady state component, and $q$ is the charge in the device. We chose the spline coefficients by the obvious generalization to three variables of the familiar univariate variation diminishing spline construction. This preserves monotonicity and gives $\|f' - S'\| = O(h^2)$. Most other groups in VLSI that used splines attempted interpolation, possibly with univariate monotonicity constraints and did not notice at first that they had lost trivariate monotonicity, meaning that in their models all voltages could increase and yet the current decrease. This is critical because the Newton optimization methods used to solve the nonlinear circuit equations can easily diverge if the transistor model has even tiny wiggles.

Measurement or simulation gives data for $f$ and $I$ and hence for partial derivatives of $q$, but not for $q$ itself. Estimating path integrals from the discrete data in an attempt to recover $q$ is unsatisfactory because different paths give different results. We used preconditioned conjugate gradients least squares on

$$\begin{bmatrix} B_1' \otimes B_2 \\ B_1 \otimes B_2' \\ 1\,0 \cdots 0 \end{bmatrix} a_q \approx \begin{bmatrix} \partial_1 q \\ \partial_2 q \\ 0 \end{bmatrix} \tag{2.2}$$

to solve this so-called "charge-extraction" problem of estimating spline coefficients $a_q$ for the $q$ model [7]. (The matrices $B_i$ and $B_i'$ are composed of univariate B-spline values and derivatives.)

Besides these examples of approximation being used to link the results of device simulation to the needs of circuit simulation, there are other uses in summarizing device results to aid the circuit designer. When voltages get too large in a transistor, a phenomenon called "avalanche breakdown" occurs. Local polynomial models of the current-voltage curves of a transistor obtained from device simulation can be analyzed to establish the boundary of a safe operating region [9]. Typical curves are shown in Fig. 1. To get the fold line of limit points ($f(z,v,w)=0$ and $\dot{v}=0$ as $w$ varies), where $z$ is the solution of a system of partial differential equations and $v$ and $w$ are scalar parameters, we apply standard continuation to the system

$$f(z,v,w)=0, \tag{2.3}$$
$$\frac{\partial f}{\partial z}=0. \tag{2.4}$$

The Rheinboldt augmenting equation [19] is well suited to device simulators, which can conveniently enforce pure voltage or current boundary conditions as needed. This summary glosses over details involving continuation in the (huge) solution $z$ versus the scalar $I$. Initially, this problem had been posed to us as needing two dimensional continuation, but we showed one dimensional continuation suffices. Even for sweeping out a full surface of current-voltage curves for a MOSFET transistor, one can use univariate sweeps. The more exotic CHINT transistor might be more interesting. Superficially, two dimensional continuation looks like isosurface generation, but the essential difference is cost of function evaluation and availability of derivatives. Each function evaluation here may cost minutes on a supercomputer.

One step of aggregation beyond a transistor model is a "macromodel" of a logic gate made up of a few transistors. For example, a three transistor NAND gate in NMOS technology satisfies the static equation

$$f(w,v_1,0)=f(v_1-w,v_2-w,v_3-w) \tag{2.5}$$

where voltages $v_i$ are on external terminals of the NAND gate but $w$ is only an internal voltage. The global nonlinear system to be solved for circuit voltages can avoid $w$, not only reducing the dimension of the optimisation problem but, more importantly, removing a variable that often leads to poor conditioning of the Jacobian. Note that with a tensor quadratic transistor model, eliminating $w$ requires finding zeros of a piecewise polynomial of degree 6, but with a quadratic box spline model, only quadratic solves are needed.

# 3 VLSI device simulation

As mentioned above, insight of considerable industrial importance has been obtained from numerical study of what goes on inside a transistor when voltages are applied to its contacts. The following short discussion is intended to give a flavour of the application area. For a more extended presentation, see [16] or the excellent papers in [14].

Most of these numerical simulations depend on the so-called "drift-diffusion" partial differential equations. The exact form of these equations need not concern us here; what is relevant is that they contain mobility, diffusion, and lifetime coefficients which are difficult to measure directly but which can, in principle, be extracted from Monte Carlo simulation of the more fundamental Boltzman transport equation. This simulation is much like the galaxy simulations familiar in scientific computing, now with electrons and holes following equations of motion,

$$\frac{d\mathbf{r}}{dt} = \text{const}\nabla_{\mathbf{k}} E_\nu(\mathbf{k}) \tag{3.1}$$

$$\frac{d\mathbf{k}}{dt} = \text{const}\mathbf{F}(\mathbf{r}), \tag{3.2}$$

where $\mathbf{r}$ represents position, $\mathbf{k}$ momentum, $E_\nu$ energy, and $\mathbf{F}$ the electric field. Unlike galaxy simulation, there are special scattering events, dominantly interactions of the electrons and holes with the silicon lattice.

A model for kinetic energy is a key part of the physics. We wish to approximate $E(\mathbf{k})$, the energy corresponding to a momentum vector $\mathbf{k}$. Without loss of generality in typical semiconductors such as silicon and gallium arsenide, we can assume $\mathbf{k}$ lies in the positive octant. Actually, the asymmetric unit is 1/48 of the total but, to achieve simpler and faster code, we are willing to store a larger table for 1/8 of the whole.

One can start with a simple parabolic model, $E(\mathbf{k}) = \text{const}\|\mathbf{k}\|_2^2$, but "hot electron" effects important for transistor reliability demand a full band model. At any particular $\mathbf{k}$, $E$ can be computed by solving modest (order 100, say) eigenproblems [3].

Also we need to invert $E$. After a scattering event, we are given $\bar{E}$ and we want a random $\mathbf{k}$ chosen from a distribution over the isosurface $E(\mathbf{k}) = \bar{E}$. In simple models, the distribution is uniform; in more advanced models, there may also be a density-of-states factor.

Finally, besides being able to evaluate $E$ and its inverse, we wish to compute segments of carrier trajectories by evaluating $\int_t^{t+\delta t} \nabla_{\mathbf{k}} E(\mathbf{k})$. With a uniform grid, one can quickly pick a $\delta t$ to hit the grid cell boundary, and can integrate exactly the quadratic element. All of these computational requirements point toward the desirability of a piecewise quadratic model.

Present technology in the semiconductor simulation field is either to take $E(\mathbf{k})$ as a (single global) quadratic, or to tabulate it on a very fine mesh and solve the inversion problem by exhaustive search. The former is fast but seriously in error for large electric fields; the latter is too slow (a typical simulation takes days on a supercomputer) [12].

# 4 Quadratic box splines in three variables

Consider direction vectors $\Xi = \{\xi_1, \ldots, \xi_n\}$ in $\mathcal{Z}^s$. The box B-spline $M_\Xi$ is a piecewise polynomial of degree $n - s$ defined [1] by

$$\int_{\mathbb{R}^s} M_\Xi \phi = \int_{[0,1]^n} \phi(\Xi t)\, dt \tag{4.1}$$

for all continuous test functions $\phi$. The fundamental recurrence relation is

$$M_\Xi(x) = \frac{1}{n-s} \sum_{1 \le i \le s} [t_i M_{\Xi\setminus\{\xi_i\}}(x) + (1 - t_i) M_{\Xi\setminus\{\xi_i\}}(x - \xi_i)] \tag{4.2}$$

```
double m1234(double *x)
{
   int l, gx = x[0]+2, gy = x[1]+1;
   double m, u = x[0]-(gx-2), v = x[1]-(gy-1);
   if(u>=v)
      if(u<=1-v) l=0; else l=2;
    else
      if(u<=1-v) l=1; else l=3;
   switch(l + 4*(gx + 4*gy)){
   case 22:
   case 23: m = u*u+2*u*v-2*u+v*v-2*v+1; break;
   case 24: m = 2*v*v; break;
   case 25: m = -u*u+2*u*v+v*v; break;
   case 26: m = -u*u-2*u*v+2*u+v*v+2*v-1; break;
   case 27: m = -2*u*u+2*u+2*v-1; break;
   case 29:
   case 31: m = u*u-2*u*v+v*v; break;
   case 36: m = u*u+2*u*v-v*v; break;
   case 37: m = 2*u*u; break;
   case 38: m = 2*u-2*v*v+2*v-1; break;
   case 39: m = u*u-2*u*v+2*u-v*v+2*v-1; break;
   case 40: case 41: case 42:
   case 43: m = -2*u*u+2*u-2*v*v+2*v+1; break;
   case 44: m = u*u-2*u*v-2*u-v*v+2*v+1; break;
   case 45: m = -2*u-2*v*v+2*v+1; break;
   case 46: m = 2*(u*u-2*u+1); break;
   case 47: m = u*u+2*u*v-4*u-v*v+2; break;
   case 52:
   case 54: m = u*u-2*u*v+v*v; break;
   case 56: m = -2*u*u+2*u-2*v+1; break;
   case 57: m = -u*u-2*u*v+2*u+v*v-2*v+1; break;
   case 58: m = -u*u+2*u*v+v*v-4*v+2; break;
   case 59: m = 2*(v*v-2*v+1); break;
   case 60:
   case 61: m = u*u+2*u*v-2*u+v*v-2*v+1; break;
   default: m = 0;
   }
   return m/4;
}
```

FIGURE 2. A C function for the Zwart-Powell bivariate box spline.

where $x = \sum_i t_i \xi_i$. We assume here that $\Xi$ has full rank and that the lower order box splines $M_{\Xi \setminus \{\xi_i\}}$ are continuous at their points of evaluation. The recurrence may be started from square full-rank $\Xi$, for which the box spline is the constant

$$M_\Xi(\Xi t) = \frac{1}{|\det \Xi|} \tag{4.3}$$

for $t \in [0,1)^s$ and 0 elsewhere.

Let $r$ be the redundancy of $\Xi$: if any $r$ vectors are removed from $\Xi$ the rest still span $\mathbb{R}^s$. Then $M_\Xi \in C^{r-1}$. Let $S$ be the integer translates of $M_\Xi$; all polynomials of degree $r$ but not all of degree $r+1$ are in $S$. Define a scaled version: $S_h = \text{span}\{M_\Xi(\frac{\cdot}{h} - \mathbf{j}); \mathbf{j} \in \mathcal{Z}^s\}$; then for $f \in C^\infty$, $\|f - S_h\|_\infty = O(h^{r+1})$.

The general theory applies in arbitrary dimension $s$, but explicit examples in the literature have mainly assumed $s = 2$. The directions

$$\Xi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{4.4}$$

give the familiar piecewise linear hat function. The direction vectors for a tensor product spline are along coordinate directions; for example, the bilinear B-spline comes from

$$\Xi = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{4.5}$$

Adding a diagonal gives a three direction mesh studied in [2]. For dim $S$ on bivariate general $n$-direction meshes, see [15]; for general dimension $s$ as well, see [10], which also gives a way to compute linear dependencies of the box splines.

The four-direction mesh in two variables

$$\Xi = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \tag{4.6}$$

is particularly interesting. A locally supported $C^1$ quadratic element on this mesh is nonzero on an octagon contained in $[-1,2] \times [0,3]$. For more details, see [22], [17], [20] and [6]. The explicit polynomial pieces of $M_\Xi$ are given in [6]. These polynomials can be derived using the recurrence relation with the aid of a symbolic algebra system such as Reduce [13], yielding a C function to compute this "Zwart-Powell" element such as the one displayed in Fig. 2.

This construction in two variables is so very appealing that it came as a keen disappointment to discover that there is no close analogue in three variables.

**Theorem 4.1** *Let $\Gamma$ be the three-dimensional tetrahedral mesh defined by adding diagonal planes to a cubic lattice. There is no compactly supported $C^1$ total-degree piecewise quadratic on $\Gamma$.*

**Proof** If there were such a compact $Q$, there would be some "lower left" lattice cube on which $Q$ is not identically zero, but $Q = \partial_n Q = 0$

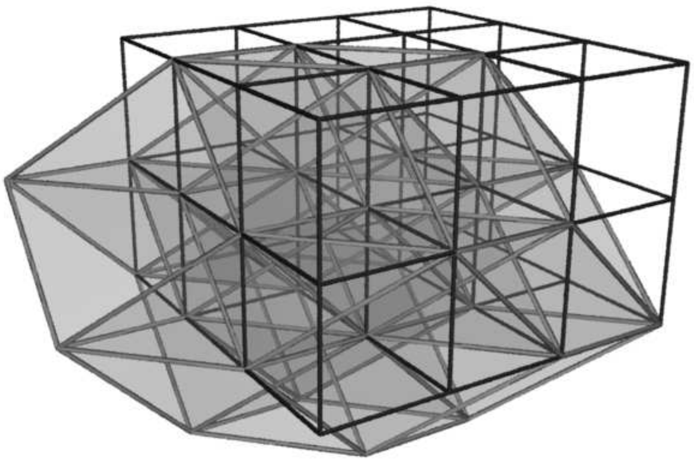

FIGURE 3. Breakplanes of a $C^1$ quadratic box spline, determined by pairs of columns of $\Xi$.

on three adjoining faces, say the bottom and two sides. $\Gamma$ cuts the cube into 24 tetrahedra. Impose $C^1$ zero boundary condition on the bottom face of the cube. Then a little algebra shows that the quadratics in the four tetrahedra on the bottom face of the cube must be $\alpha z^2$ where $0 \le z \le \frac{1}{2}$ is the vertical coordinate. Similarly, the quadratics on the four tetrahedra touching the first side face must be $\beta y^2$ and the second side face $\gamma z^2$. Now imposing continuity on the interior faces of these tetrahedra forces $\alpha = \beta = \gamma = 0$. Imposing $C^1$ continuity on the remaining interior faces forces the quadratics on the remaining tetrahedra to be 0. $\square$

Another way of grasping this result is to see that a naive generalization of the Zwart-Powell element tries to build a box spline on

$$\Xi = \begin{bmatrix} 1 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{4.7}$$

But if columns 2 and 3 are removed, the rest do not span $\mathbb{R}^3$. So this box spline can only be $C^0$. The point of the theorem is that the failure is not an infelicity of the box spline definition but an inflexibility of trivariate quadratics.

Although our preferred mesh does not support the $C^1$ quadratic we desire, clearly we can take some other mesh for which the box spline construction is valid. Using Reduce, an exhaustive listing was made of all $\Xi$ with entries $-1$, 0, 1 meeting the redundancy criterion and having minimum volume $\sqrt{\det \Xi\Xi^T}$. Among these,

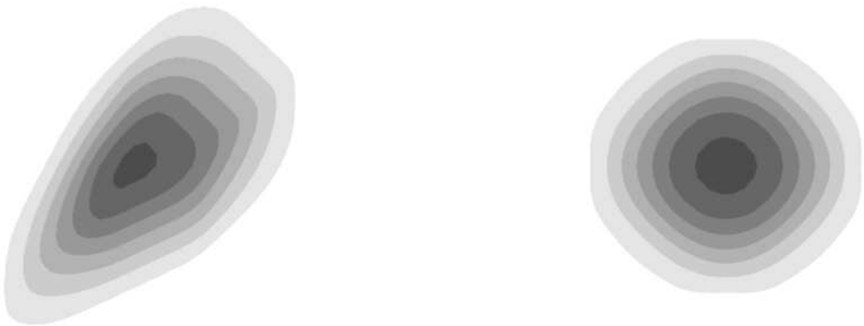

FIGURE 4. Contours of a planar slice through a trivariate quadratic box spline, and through averages of permutations.

one was arbitrarily chosen with a single $-1$, namely

$$\Xi = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & -1 \end{bmatrix}. \tag{4.8}$$

The support of this box spline is shown in Fig. 3, where the coordinate grid is shown for orientation and transparent planes

$$\Gamma_{loc} = \{\Xi t : \text{all but two } t_\xi \in \{0,1\}\} \tag{4.9}$$

are drawn. Constructing the explicit quadratic pieces in each region is straightforward symbolic algebra using the recurrence relation. The "$M_{\Xi\setminus\{\xi_i\}}$ are continuous" restriction of Eqn. 4.2 prohibits evaluating on $\Gamma_{loc}$. We can easily meet this constraint choosing sample points as fractions with prime denominators. Examination of contours of the box spline confirms that it is $C^1$, but as can be seen already from $\Gamma_{loc}$, it lacks the spherical shape of the Zwart-Powell box spline.

Although not a formal mathematical constraint, past experience leads us to ask for an element with support and shape that are not too unsymmetric. A sum of six copies of $M_\Xi$ based on permutations of the rows of $\Xi$ works well. Contours of $M_\Xi$ and $\sum M_\Xi$ are shown in Fig. 4. Even more circular contours may be obtained by averaging over reflections as well, but in the absence of a mathematical criterion, deciding whether or not this additional step is worthwhile is a matter of taste.

In the particular application of box splines to Monte Carlo modelling we need to deal with the explicit quadratic pieces. The faces of $\Gamma_{loc}$ are obtained by taking cross products of all pairs of columns of $\Xi$ and considering planes orthogonal to these normals and passing through lattice points. There are 15 cutting planes

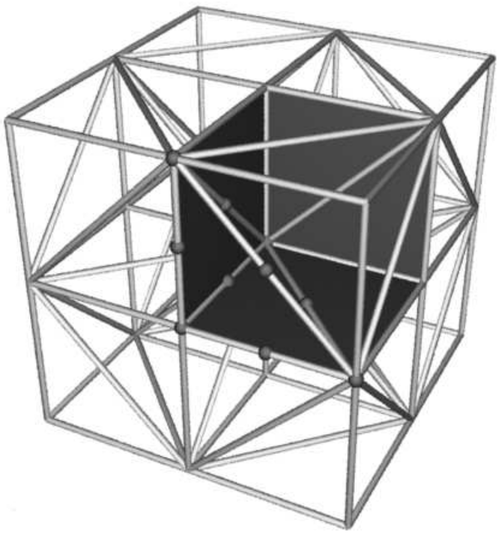

FIGURE 5. If $C^1$ is not necessary, a scheme based on five tetrahedra per cube may be used. Function values at vertices and midpoints of edges determine $C^0$ quadratics.

besides the ones that lie on lattice cube faces. To find the boundary of a particular piece, take the polar dual [18, pp.283-284] of these cutting planes, obtaining 15 points in three dimensions, and then compute the convex hull of the points to determine which planes are redundant. $M_\Xi$ has 84 pieces per cube, with volumes ranging from 0.033 down to 0.002. One can build a code analogous to Fig. 2 for this trivariate quadratic box spline, with the switch statement replaced by a decision tree. To put this construction in perspective, the best existing alternative is the Worsey-Piper $C^1$ quadratic [21], which generalizes the Powell-Sabin split from triangles to tetrahedra but requires an oblique lattice and generates 144 pieces per cell.

## 5 Future directions

Symmetrizing the shape of the box spline by averaging led to too many quadratic pieces for our application. An alternative that ought to be investigated is using an affine grid to improve the shape of a single box spline, or possibly to try other $\Xi$. Once a satisfactory $M_\Xi$ is chosen, a quasi-interpolant can be readily constructed by the Neumann series approach of [5].

A motivation for seeking total-degree quadratic rather than tensor-quadratic is that zero-finding along a ray in general orientation ought to be much faster. Also, isosurface generation ought to be much faster and compact. However, detailed algorithms for these tasks remain to be developed. To render images of the isosurface of the box spline, we use [11] applied to individual quadratic pieces.

As a stopgap while the box spline approach continues to be explored, we have used $C^0$ quadratics on a uniform grid with each cube divided into five tetrahedra, as shown in Fig. 5, with a standard ten-node quadratic finite element. This trivial construction achieves the same $O(h^3)$ convergence as box splines but gives up the smooth gradients we would have liked for better trajectories and isosurfaces. Nevertheless, the physics program based on this approach achieved an overall factor of two speedup over the existing design based on tricubic interpolation.

Kent Smith [4] has suggested using a tetrahedral grid conforming to isosurfaces. This scheme has the disadvantage of more expensive grid search for random probes and difficult construction of $C^1$ elements, but is uniquely well suited to the inversion problem.

In closing, I would like to thank Carl de Boor, Klaus Höllig, and Sherman Riemenschneider for an early look at their outstanding book on box splines (forthcoming). Jeff Bude, Kent Smith, and Franco Venturi patiently described aspects of the Boltzman transport problem to me; Bill Coughran and Mark Pinto explained the simulation context. Steve Fortune suggested the polar dual. Jorg Peters made helpful comments on a draft of this paper.

## References

[1] C. de Boor and K. Höllig, *B-splines from parallelepipeds*, J. d'Analyse Mathématique, 42 (1982), pp. 99–115.

[2] ——, *Bivariate box splines and smooth pp functions on a three-direction mesh*, J. Computational and Applied Mathematics, 9 (1983), pp. 13–28.

[3] D. Brust, *Electronic spectra of crystalline germanium and silicon*, Physical Review, 134 (1964), pp. A1337–1347.

[4] J. Bude, E. Grosse, and R. K. Smith, *Phase-space simplex Monte Carlo for semiconductor transport*, in preparation, (1993).

[5] C. K. Chui and H. Diamond, *A natural formulation of quasi-interpolation by multivariate splines*, Proc. Amer. Math. Soc., 99 (1987), pp. 643–646.

[6] C. K. Chui and R.-H. Wang, *On a bivariate B-spline basis*, Tech. Rep. 7, Texas A&M University, Center for Approximation Theory, 1981.

[7] W. M. Coughran, Jr., W. Fichtner, and E. Grosse, *Extracting transistor charges from device simulations by gradient fitting*, IEEE Transactions on Computer Aided Design, 8 (1989), pp. 380–394.

[8] W. M. Coughran, Jr., E. Grosse, and D. J. Rose, *Variation diminishing splines in simulation*, SIAM J. on Scientific and Statistical Computation, 7 (1986), pp. 696–705.

[9] W. M. Coughran, Jr., E. H. Grosse, and M. R. Pinto, *Computing folds and bifurcations in current-voltage characteristics of semiconductor devices*, in Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits: NUPAD IV. Technical Digest, IEEE, 1992, pp. 149–153.

[10] W. Dahmen and C. A. Micchelli, *Some results on box splines*, Bulletin (New Series) of the American Mathematical Society, 11 (1984), pp. 147–150.

[11] T. Duff, *Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry*, Computer Graphics (SIGGRAPH '92 Proceedings), 26 (1992), pp. 131–138.

[12] M. V. Fischetti and S. E. Laux, *Monte Carlo analysis of electron transport in small semiconductor devices including band-structure and space-charge effects*, Physical Review B, 38 (1988), pp. 9721–9745.

[13] A. C. Hearn, *REDUCE User's Manual*, The Rand Corporation, Los Angeles, 1987.

[14] K. Hess, ed., *Monte Carlo Device Simulation: Full Band and Beyond*, Kluwer Academic, 1991.

[15] D. Lee, *A note on bivariate box splines on a $k$-direction mesh*, J. Computational and Applied Mathematics, 15 (1986), pp. 117–122.

[16] M. Lundstrom, *Fundamentals of Carrier Transport*, Addison-Wesley Publishing, 1990.

[17] M. J. D. Powell, *Piecewise quadratic surface fitting for contour plotting*, in Software for Numerical Analysis, D. J. Evans, ed., Academic Press, 1974, pp. 253–271.

[18] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, 1985.

[19] W. C. Rheinboldt, *Numerical Analysis of Parametrized Nonlinear Equations*, Wiley-Interscience, 1986.

[20] L. L. Schumaker, *On spaces of piecewise polynomials in two variables*, in Approximation Theory and Spline Functions, S. P. Singh et al., eds., Reidel Publishing, 1984, pp. 151–197.

[21] A. J. Worsey and B. Piper, *A trivariate Powell-Sabin interpolant*, Computer Aided Geometric Design, 5 (1988), pp. 177–186.

[22] P. B. Zwart, *Multivariate splines with nondegenerate partitions*, SIAM J. Numerical Analysis, 10 (1973), pp. 665–673.