An algorithm for smoothing three-dimensional Monte Carlo ion implantation simulation results

Clemens Heitzinger*, Andreas Hössinger, Siegfried Selberherr

Institute for Microelectronics, Technical University of Vienna, Gußhausstraße 27–29, 1040 Wien, Austria

Abstract

We present an algorithm for smoothing results of three-dimensional Monte Carlo ion implantation simulations and translating them from the grid used for the Monte Carlo simulation to an arbitrary unstructured three-dimensional grid. This algorithm is important for joining various simulations of semiconductor manufacturing process steps, where data have to be smoothed or transferred from one grid to another. Furthermore different grids must be used since using ortho-grids is mandatory because of performance reasons for certain Monte Carlo simulation methods. The algorithm is based on approximations by generalized Bernstein polynomials. This approach was put on a mathematically sound basis by proving several properties of these polynomials. It does not suffer from the ill effects of least squares fits of polynomials of fixed degree as known from the popular response surface method. The smoothing algorithm which works very fast is described and in order to show its applicability, the results of smoothing a three-dimensional real world implantation example are given and compared with those of a least squares fit of a multivariate polynomial of degree 2, which yielded unusable results.
© 2003 IMACS. Published by Elsevier B.V. All rights reserved.

Keywords: Monte Carlo simulation; Ion implantation; Bernstein polynomials; Smoothing operator

1. Introduction

After a Monte Carlo simulation of ion implantation on an ortho-grid, the question arises how to translate the resulting values, i.e., concentrations, to an unstructured grid. In the Monte Carlo simulation an ortho-grid is commonly used in order to achieve workable simulation times, since calculating point locations, i.e., tracing the position of ions, dominates performance. For other, subsequent simulations via, e.g., the finite element method, it is mandatory to use different, unstructured grids. Furthermore, the resulting values have to be smoothed in order to provide suitable input for the simulation of subsequent process steps like diffusion.

---

* Corresponding author. Tel.: +43-1-58801-36035; fax: +43-1-58801-36099.
E-mail address: clemens.heitzinger@tuwien.ac.at (C. Heitzinger).

Thus an algorithm for smoothing Monte Carlo ion implantation results has to meet the following demands: it has to work with unstructured target grids, it must provide suitable smoothing, and since the number of grid points in the target grid is usually large, it must not be computationally expensive.

One simple approach is to perform a least squares fit of a multivariate polynomial of fixed degree, usually two, and to hope that this polynomial is a suitable approximation providing proper smoothing. This is known as the response surface methodology (RSM) [5] approach and has been used to a great extent in TCAD applications, but it does often not work satisfactorily (c.f. Fig. 4). In order to solve this problem, generalizations of Bernstein polynomials were devised and their properties proven. Hence a fast algorithm based on these polynomials was developed and applied to a real world example. The RSM approach will be compared to the proposed algorithm since least squares fits are a popular method: RSM has been used extensively in TCAD applications, e.g. in [2,4,6,9,12,14,16, 19,20].

Although it can be argued that the RSM approximation is based on a truncated Taylor series expansion $f(\boldsymbol{r} + \boldsymbol{a}) = \sum_{k=0}^{\infty} \left((1/k!)(\boldsymbol{a} \cdot \nabla_{\boldsymbol{r'}})^k f(\boldsymbol{r'}) \right) \big|_{\boldsymbol{r'}=\boldsymbol{r}}$ for a multivariate function f, it is important to note that this is a local approximation and quite different from a least squares fit for several points. In the Taylor series expansion convergence occurs when the number of terms and thus the degree of the polynomial increases, whereas in the RSM approach the degree of the approximating polynomial is fixed to an arbitrary low value. Increasing the degree is possible of course, but the choice is still arbitrary and the number of coefficients and thus the number of points required for the least squares fit increases abundantly.

Furthermore, the RSM suffers from the fact that a polynomial of fixed degree cannot preserve the global properties of the original function: the set of all polynomials of a certain fixed maximal degree is not dense in $C(X)$, $X \subset \mathbb{R}^p$ compact.

Although the RSM approach can be improved by transforming the variables before fitting the polynomials, it has to be known a priori which transformations are useful and should be considered. If this knowledge is available, it can of course be applied to other approximation approaches as well.

Finally, an advantage of the RSM approach is the simple structure of the approximations: it is easy to deal with polynomials of degree 2. However, in the algorithm proposed in the following no polynomials have to be constructed explicitly and the computational effort for doing least squares fits is eliminated as well.

2. Properties of multivariate Bernstein polynomials

The Weierstraß Approximation Theorem states that continuous functions on compact intervals can be arbitrarily well approximated by polynomials. One constructive way to obtain such polynomials are Bernstein polynomials which were first introduced by Sergei N. Bernstein in the univariate case. A generalization to multidimensional intervals and its properties is presented in this section. Generalizations to multidimensional simplices using barycentric coordinates and other properties of Bernstein polynomials can be found e.g. in [1,3,7,13,15,17,18].

In order to keep the formulae simple only functions defined on the multidimensional intervals $[0, 1]^N$, i.e., the unit cubes in $\mathbb{R}^N$, are considered. Using affine transformations it is straightforward to apply the formulae and results to arbitrary intervals.

Definition 2.1 (Multivariate Bernstein polynomials). Let $n_1, \ldots, n_m \in \mathbb{N}$ and f be a function of m variables. The polynomials

$$B_{f,n_1,\ldots,n_m}(x_1, \ldots, x_m) := \sum_{\substack{0 \le k_j \le n_j \\ j \in \{1,\ldots,m\}}} f\left(\frac{k_1}{n_1}, \ldots, \frac{k_m}{n_m}\right) \prod_{j=1}^{m} \left(\binom{n_j}{k_j} x_j^{k_j}(1 - x_j)^{n_j - k_j}\right)$$

are called the multivariate Bernstein polynomials of f.

We note that $B_{f,n_1,\ldots,n_m}$ is a linear operator.

Lemma 2.2. *For all $x \in \mathbb{R}$*

$$\sum_{k=0}^{n} (k - nx)^2 \binom{n}{k} x^k(1 - x)^{n-k} = nx(1 - x).$$

For all $x \in [0, 1]$ we have $x(1 - x) \le 1/4$ and hence

$$0 \le \sum_{k=0}^{n} (k - nx)^2 \binom{n}{k} x^k(1 - x)^{n-k} \le \frac{n}{4}.$$

Theorem 2.3 (Uniform convergence). *Let $f : [0, 1]^m \to \mathbb{R}$ be a continuous function. Then the multivariate Bernstein polynomials $B_{f,n_1,\ldots,n_m}$ converge uniformly to f for $n_1, \ldots, n_m \to \infty$.*

Proof. We first note that because of the uniform continuity of f on $I := [0, 1]^m$ we have

$$\forall \varepsilon > 0 : \exists \delta > 0 : \forall x, x' \in I : \|x - x'\| < \delta \Rightarrow \|f(x) - f(x')\| < \tfrac{1}{2}\varepsilon.$$

Given an $\varepsilon > 0$, we can find such a δ. In order to simplify notation we set

$$b_j := \binom{n_j}{k_j} x_j^{k_j}(1 - x_j)^{n_j - k_j}$$

and

$$k := \left(\frac{k_1}{n_1}, \ldots, \frac{k_m}{n_m}\right).$$

x always lies in I. We have to estimate

$$B_{f,n_1,\ldots,n_m}(x) - f(x) = \sum_{\substack{0 \le k_j \le n_j \\ j \in \{1,\ldots,m\}}} (f(k) - f(x))b_1, \ldots, b_m$$

and to that end we split the sum into two parts, namely

$$S_1 := \sum{}' (f(k) - f(x))b_1, \ldots, b_m,$$

where $\sum'$ means summation over all k_j with $0 \leq k_j \leq n_j$ (where $j \in \{1, \ldots, m\}$) and $\|k - x\|_2 \geq \delta$, and

$$S_2 := \sum{}'' (f(k) - f(x))b_1, \ldots, b_m,$$

where $\sum''$ means summation over the remaining terms. For S_2 we have

$$|S_2| \leq \sum{}'' |f(k) - f(x)|b_1, \ldots, b_m < \frac{\varepsilon}{2} \sum{}'' b_1, \ldots, b_m \leq \frac{\varepsilon}{2} \sum_{\substack{0 \leq k_j \leq n_j \\ j \in \{1, \ldots, m\}}} b_1, \ldots, b_m = \frac{\varepsilon}{2}.$$

We will now estimate S_1. In the sum S_1 the inequality $\|k - x\|_2 \geq \delta$ holds, i.e.,

$$\left(\frac{k_1}{n_1} - x_1\right)^2 + \cdots + \left(\frac{k_m}{n_m} - x_m\right)^2 \geq \delta^2.$$

Hence at least one of the summands on the left hand side is greater equal δ^2/m. Without loss of generality we can assume this is the case for the first summand:

$$1 \leq \frac{m}{\delta^2} \frac{(k_1 - n_1 x)^2}{n_1^2}.$$

Thus, using Lemma 2.2,

$$\sum{}' b_1, \ldots, b_m \leq \sum{}' \frac{m}{\delta^2 n_1^2} (k_1 - n_1 x)^2 b_1, \ldots, b_m \leq \frac{m}{\delta^2 n_1^2} \sum_{\substack{0 \leq k_j \leq n_j \\ j \in \{1, \ldots, m\}}} (k_1 - n_1 x)^2 b_1, \ldots, b_m$$

$$= \frac{m}{\delta^2 n_1^2} \sum_{k_1 = 0}^{n_1} (k_1 - n_1 x)^2 \binom{n_1}{k_1} x_1^{k_1} (1 - x_1)^{n_1 - k_1} \leq \frac{m}{\delta^2 n_1^2} \frac{n}{4} = \frac{m}{4\delta^2 n_1}.$$

We can now estimate S_1. Since f is continuous on a compact set $M := \max_{x \in I} |f(x)|$ exists.

$$|S_1| \leq \sum{}' |f(k) - f(x)|b_1, \ldots, b_m \leq 2M \sum{}' b_1, \ldots, b_m \leq \frac{2Mm}{4\delta^2 n_1} = \frac{Mm}{2\delta^2 n_1}$$

For n_1 large enough we have $Mm/2\delta^2 n_1 < \varepsilon/2$ and thus

$$|B_{f, n_1, \ldots, n_m}(x) - f(x)| \leq |S_1| + |S_2| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon,$$

which completes the proof. $\qquad\square$

Corollary 2.4. *The set of all polynomials is dense in $C([0, 1]^m)$.*

By presupposing more knowledge about the rate of change of the function, namely a Lipschitz condition, an error bound is obtained.

Theorem 2.5 (Error bound for Lipschitz condition). *If $f : I := [0, 1]^m \to \mathbb{R}$ is a continuous function satisfying the Lipschitz condition*

$$\| f(x) - f(y) \|_2 < L \| x - y \|_2$$

on I, then the inequality

$$\| B_{f,n_1,\ldots,n_m}(x) - f(x) \|_2 < \frac{L}{2} \left(\sum_{j=1}^{m} \frac{1}{n_j} \right)^{1/2}$$

holds.

Proof. Abbreviating notation we set

$$k := \left(\frac{k_1}{n_1}, \ldots, \frac{k_m}{n_m} \right).$$

We will use the Lipschitz condition, the Cauchy–Schwarz inequality, and Lemma 2.2.

$$\| B_{f,n_1,\ldots,n_m}(x) - f(x) \|_2^2 \leq \left(\sum_{k_1,\ldots,k_m} \| f(k) - f(x) \|_2 b_1, \ldots, b_m \right)^2 < \left(L \sum \| k - x \|_2 b_1, \ldots, b_m \right)^2$$

$$\leq L^2 \left(\sum \| k - x \|_2^2 b_1, \ldots, b_m \right) \left(\sum b_1, \ldots, b_m \right) = L^2 \sum \left(\left(\frac{k_1}{n_1} - x_1 \right)^2 \right.$$

$$\left. + \cdots + \left(\frac{k_m}{n_m} - x_m \right)^2 \right) b_1, \ldots, b_m = L^2 \sum_{j=1}^{m} \frac{x_j(1 - x_j)}{n_j} \leq L^2 \sum_{j=1}^{m} \frac{1}{4n_j}$$

This completes the proof. □

Theorem 2.6 (Asymptotic formula). *Let $I := [0, 1]^m$, let $f : I \to \mathbb{R}$ be a C^2 function, and let $x \in I$, then*

$$\lim_{n \to \infty} n(B_{f,n,\ldots,n}(x) - f(x)) = \sum_{j=1}^{m} \frac{x_j(1 - x_j)}{2} \frac{\partial^2 f(x)}{\partial x_j^2} \leq \frac{1}{8} \sum_{j=1}^{m} \frac{\partial^2 f(x)}{\partial x_j^2}.$$

Proof. We define the vector h through $h_j := k_j/n - x_j$, where the k_j are the integers over which we sum in $B_{f,n,\ldots,n}$. Using Taylor's theorem we see

$$f \left(\frac{k_1}{n}, \ldots, \frac{k_m}{n} \right) = f(x) + \sum_{j=1}^{m} \left(\frac{k_j}{n} - x_j \right) \frac{\partial f(x)}{\partial x_j}$$

$$+ \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\frac{k_i}{n} - x_i \right) \left(\frac{k_j}{n} - x_j \right) \frac{\partial^2 f(x_0)}{\partial x_i \partial x_j} + \| h \|^2 \rho(h),$$

where $\lim_{h \to 0} \rho(h) = 0$. Summing this equation like the sum in $B_{f,n,\dots,n}$ we obtain

$$B_{f,n,\dots,n} = f(x) + \frac{1}{2} \sum_{i=1}^{m} \frac{x_j(1 - x_j)}{n} \frac{\partial^2 f(x_0)}{\partial x_j^2} + \sum_{k_1,\dots,k_m} \|h\|^2 \rho(h) b_1, \dots, b_m$$

since many terms vanish or can be summed because of Lemma 2.2. Noting $\lim_{h \to 0} \rho(h) = 0$ we can apply the same technique as in the proof of Theorem 2.3 for estimating the last sum in the last equation, i.e., splitting the sum into two parts for $\|h\| \geq \delta$ and $\|h\| < \delta$. Hence we see that for all ε this sum is less equal ε/n for all sufficiently large n, which yields the claim. $\square$

This asymptotic formula gives information about the rate of convergence, and states that it depends only on the partial derivatives $\partial^2 f(x)/\partial x_j^2$. This is noteworthy, since it is often the case that the smoother a function is and the more is known about its higher derivatives, the more properties can be proven, but in this case only the second order derivatives play a role.

The following theorem states that the total variation of the Bernstein polynomial of a function of one variable is less equal than the total variation of the function itself. Thus the Bernstein approximation operator has a smoothing effect.

Theorem 2.7 (Total variation). *Let $V(f, [a, b])$ be the total variation of f over $[a, b]$ and let $f : [0, 1] \to \mathbb{R}$ be a continuous function. Then*

$$V(B_{f,n}, [0, 1]) \leq V(f, [0, 1]),$$

where the equality sign holds if and only if the function f is monotone.

This means the approximation is smoother than the original function regarding the amount of total variation. Proofs of this theorem can be found in [15,18], where the case of equality is discussed.

Not only is the total variation reduced by the Bernstein operator, but it also has the following variation diminishing property.

Theorem 2.8 (Variation diminishing property). *Let $Z(f, (a, b))$ be the number of real zeros of f in the interval (a, b) and let $f : [0, 1] \to \mathbb{R}$ be a continuous function. Then*

$$Z(B_{f,n}, (0, 1)) \leq v(f),$$

where $v(f)$ is the number of changes of sign of f in $[0, 1]$.

This last theorem is the reason for the excellent smoothing properties of polynomials of Bernstein type. It states that Bernstein polynomials should be used whenever a polynomial approximation is needed which does not oscillate more often about any straight line than the function to be approximated [18].

Concerning the numerical aspect, an implementation for univariate Bernstein polynomials was presented in [21]. The higher the degree of the approximation polynomial, the more care has to be taken in their numerical evaluation. In the cases needed for our applications, this is not an issue.

3. The algorithm

The algorithm works by constructing approximating multivariate Bernstein polynomials in the neighborhood of the points of the unstructured, new grid. Let A be the initial isotropic homogeneous grid, where values are associated with the volume cells, as is usually the case in Monte Carlo simulations of ion implantations, and B an arbitrary grid where values are associated with the grid points. This grid is to be used in following simulations and hence it is determined by their demands. It is often an anisotropic inhomogeneous one.

For each point of grid B, m^d neighboring points are used for constructing an approximation value for the point considered (c.f. Fig. 1), where $m \geq 3$, m odd, and d is the dimension. $m = 5$ was chosen in the example below and provides good smoothing results. At the boundary the values of grid A are extended constantly. Thus m^d points are used for constructing a multivariate Bernstein polynomial which is evaluated at the point in the middle in question. Note that it is not necessary to calculate the polynomial explicitly, since each polynomial is later evaluated at one point only. Additionally, it is not necessary to use an affine transformation by assuming that the convex hull of the neighboring points is $[0, 1]^d$ and the middle point has coordinates $(1/2, \ldots, 1/2)$.

Thus for three dimensions and setting $n := m - 1$, the values of the points of grid B are

$$B_{f,n,n,n}(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}) = \frac{1}{8^n} \sum_{k_1=0}^{n} \sum_{k_2=0}^{n} \sum_{k_3=0}^{n} f_{k_1,k_2,k_3} \binom{n}{k_1} \binom{n}{k_2} \binom{n}{k_3},$$

where f_{k_1,k_2,k_3} are the values of the corresponding cell of grid A and $f_{0,0,0}$ has coordinates $(0, 0, 0)$ and $f_{n,n,n}$ has coordinates $(1, 1, 1)$.

One of the benefits of this algorithm is that it can be implemented in a straightforward manner in languages like C and Fortran using the expression for $B_{f,n,n,n}(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2})$ given above. In order to minimize computation time, the values of the binomial coefficients can be pre-calculated and stored in arrays.
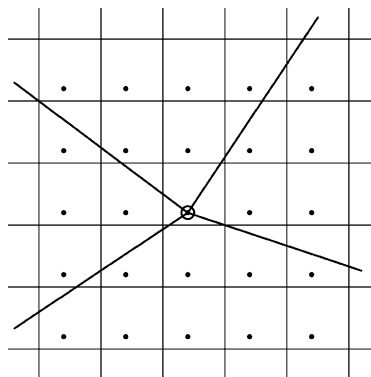

Fig. 1. This figure illustrates how the calculations for one point of the unstructured target grid B are performed in a two-dimensional example. The thin orthogonal lines confine the cells of the initial grid A, the four sloped lines denote the unstructured grid, and the point in the middle is the one currently considered. The 5^2 points show which values are used for determining the approximating polynomial.

Furthermore, it is fast so that it can be used for grids containing hundreds of thousands of points. Due to the theorems given above, its smoothing and approximating properties are outstanding. Thus it is faster, easier to implement, and approximates and smoothes better than the RSM approach of fitting polynomials of fixed degree.

4. A three-dimensional example

The example is a three-dimensional CMOS structure which consists of poly-silicon in the upper part, of silicon dioxide in the middle part, and silicon in the lower part. A boron dose of 10^{13} cm^{-2} with an energy of 15 keV was implanted in a Monte Carlo simulation [10,11] using an isotropic homogeneous grid. The resulting concentration of boron interstitial atoms in (cm^{-3}) is shown in Figs. 2–6. The new anisotropic inhomogeneous grid with 78,651 grid points was generated by DELINK [8] and is additionally shown in Fig. 2. In Figs. 2, 3, 5 and 6 the new algorithm was applied on 5×5 grids, whereas in Fig. 4 least squares fits of polynomials of degree 2 on grids of the same size were performed.

Obviously the result in Fig. 4 is inferior to the result yielded by the algorithm described in the previous section. In order to interpret the failure of the RSM method, it is important to note that the shape of the RSM polynomials of degree 2 does not allow enough change to adapt to the points to be approximated. Because of the inherent noise in the Monte Carlo simulation result, the shapes of the RSM approximations vary strongly between neighboring elements. Furthermore, because of the limited choice of approximating polynomials, the noise may even be amplified.

The new algorithm provides very good smoothing and yields concentration values at the grid points that can serve as input to subsequent simulation steps without problems. In this example the computation
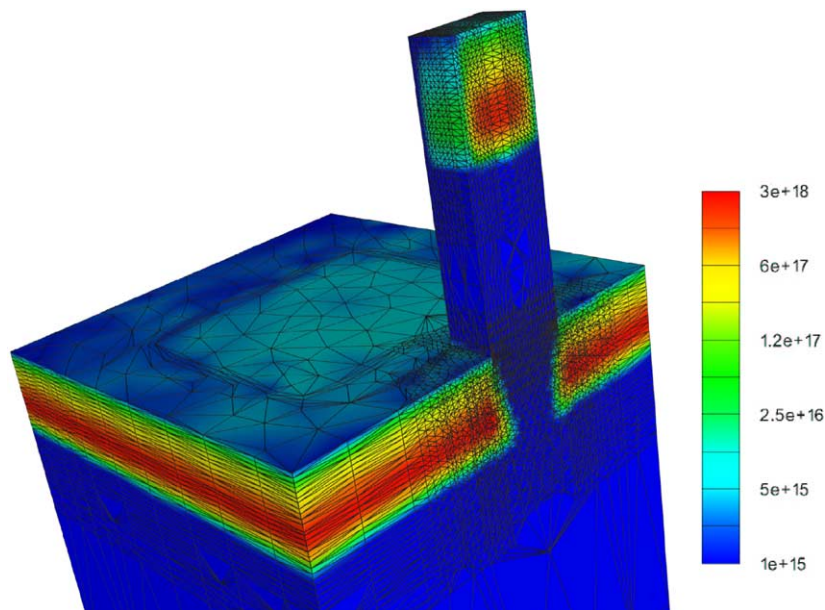

Fig. 2. A front view of the sample Monte Carlo ion implantation after smoothing using the new algorithm. The unstructured destination grid with 78,651 points is shown as well.
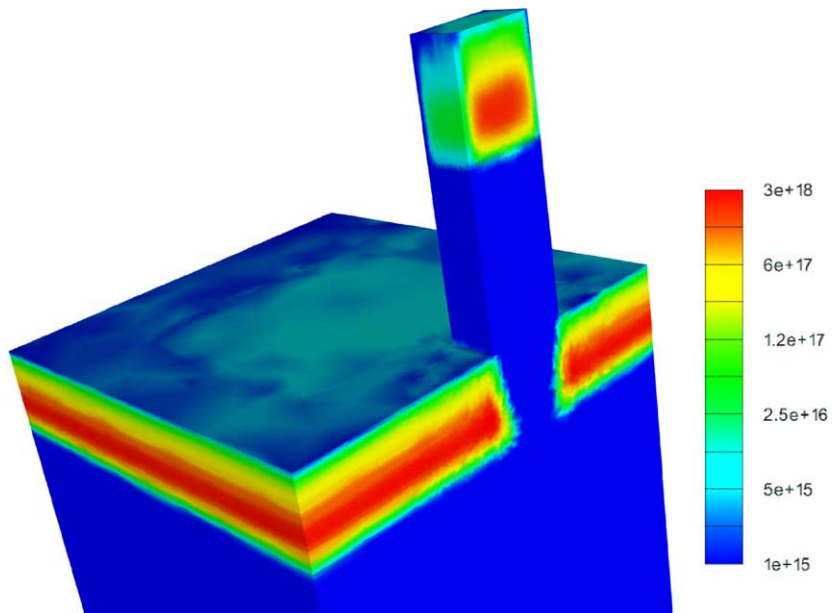
Fig. 3. A front view of the sample Monte Carlo implantation after smoothing using the new algorithm.
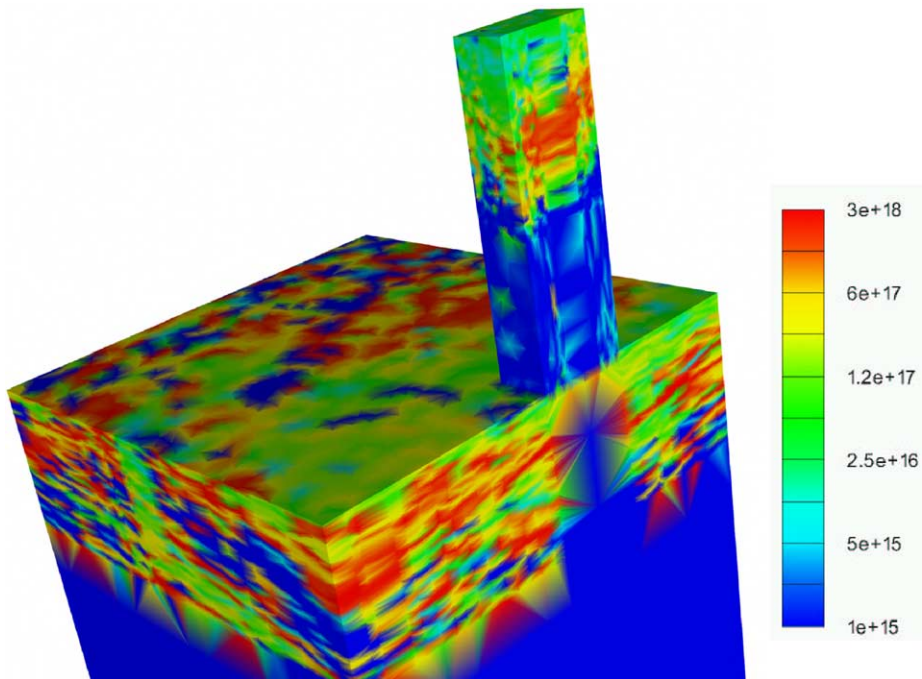

Fig. 4. A front view of the sample Monte Carlo ion implantation after extracting values using least squares fits of multivariate polynomials of degree 2.
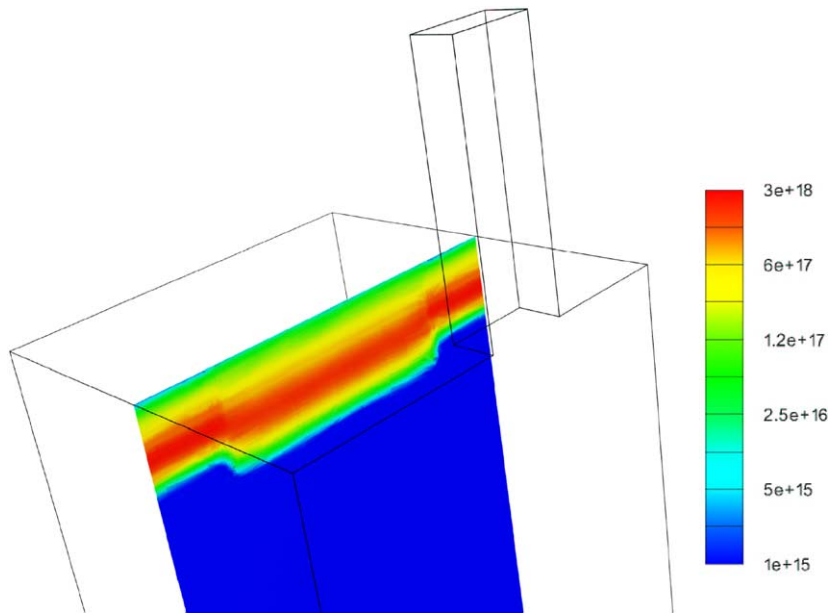
Fig. 5. A cut parallel to the front side of the sample Monte Carlo ion implantation after smoothing using the new algorithm.
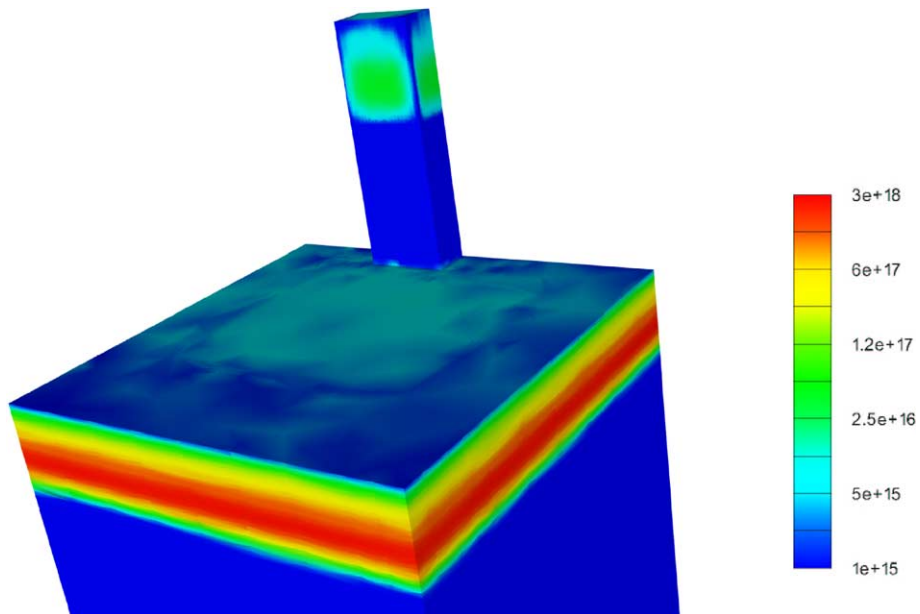

Fig. 6. A back view of the sample Monte Carlo ion implantation after smoothing using the new algorithm.

time on an Intel Pentium III processor at 1 GHz is 2.417 ms per point using the RSM method and 0.858 ms per point using the new algorithm.

5. Conclusion

In summary, the properties of polynomials of fixed degree arising from least square fits were compared to those of multivariate Bernstein polynomials. The Bernstein polynomials fulfill the requirements for approximations needed for smoothing Monte Carlo simulation results and translating them from ion implantation ortho-grids to arbitrary, unstructured grids.

The polynomials and the algorithm devised provide the following benefits. First, they converge uniformly when the number of base points goes to infinity. Second, an asymptotic formula gives information about their rate of convergence. Third, total variation is decreased and the approximations do not oscillate more often about any straight line than the original function. This assures suitable smoothing. Fourth, the algorithm works very fast and is easy to implement using the specialized formula given, since the calculation of the actual approximating polynomials is avoided.

Finally, the new algorithm and its RSM counterpart were compared in a real world Monte Carlo ion implantation example, and the new algorithm was found to yield superior results, which can immediately be used for further simulations.

Acknowledgements

References

[1] J.A. Adell, J. de la Cal, M. San Miguel, On the property of monotonic convergence for multivariate Bernstein-type operators, J. Approximation Theory 80 (1995) 132–137.
[2] A.R. Alvarez, B.L. Abdi, D.L. Young, H.D. Weed, J. Teplik, E.R. Herald, Application of statistical design and response surface methods to computer-aided vlsi device design, IEEE Trans. Computer-Aided Design 7 (2) (1988) 272–288.
[3] I.S. Berezin, N.P. Zhidkov, Computing Methods, vol. 1, Pergamon Press, 1965.
[4] D.S. Boning, P.K. Mozumder, Doe/opt: a system for design of experiments, response surface modeling, and optimization using process and device simulation, IEEE Trans. Semiconductor Manufacturing 7 (2) (1994) 233–244.
[5] G.E.P. Box, N.R. Draper, Empirical Model-Building and Response Surfaces, Wiley, New York, 1987.
[6] J.A. Cornell, How to Apply Response Surface Methodology, Second ed., American Society for Quality Control, Milwaukee, Wisconsin, 1990.
[7] Y.Y. Feng, J. Kozak, Asymptotic expansion formula for Bernstein polynomials defined on a simplex, Constructive Approximation 8 (1992) 49–58.
[8] P. Fleischmann, Mesh Generation for Technology CAD in Three Dimensions, Dissertation, Technische Universität Wien, 1999, http://www.iue.tuwien.ac.at/phd/fleischmann.
[9] G.J. Gaston, A.J. Walton, The integration of simulation and response surface methodology for the optimization of ic processes, IEEE Trans. Semiconductor Manufacturing 7 (1) (1994) 22–33.
[10] A. Hössinger, E. Langer, S. Selberherr, Parallelization of a Monte Carlo ion implantation simulator, IEEE Trans. Computer-Aided Design Integrated Circuits Syst. 19 (5) (2000) 560–567.

[11] A. Hössinger, S. Selberherr, M. Kimura, I. Nomachi, S. Kusanagi, Three-dimensional Monte-Carlo ion implantation simulation for molecular ions, in: Proceedings of the Electrochemical Society, vol. 99 (2), 1999, pp. 18–25.

[12] W. Kanert, N. Krischke, K. Wiesinger, Optimization of dmos transistors for smart power technologies by simulation and response surface methods, in: S. Selberherr, H. Stippel, E. Strasser (Eds.), Simulation of Semiconductor Devices and Processes, vol. 5, Springer, Wien, 1993, pp. 213–216.

[13] G.G. Lorentz, Bernstein Polynomials, Second ed., Chelsea Publishing Company, New York, 1986.

[14] H. Masuda, F. Otsuka, Y. Aoki, S. Sato, S. Shimada, Response surface methods for submicron MOSFETs characterization with variable transformation technology, IEICE Trans. E 74 (6) (1991) 1621–1633.

[15] T. Popoviciu, Sur l'approximation des fonctions convexes d'ordre supérieur, Mathematica 10 (1935) 49–54.

[16] H. Sato, K. Tsuneno, K. Aoyama, T. Nakamura, H. Kunitomo, H. Masuda, A new hierarchical rsm for tcad-based device design to predict cmos development, in: Proceedings of the IEEE 1995 International Conference on Microelectronic Test Structures, vol. 8, 1995, pp. 299–302.

[17] R. Schnabl, Eine verallgemeinerung der Bernsteinpolynome, Math. Ann. 179 (1968) 74–82.

[18] I.J. Schoenberg, On variation diminishing approximation methods, in: R.E. Langer (Ed.), On Numerical Approximation, The University of Wisconsin Press, Madison, 1959, pp. 249–274.

[19] V. Senez, T. Hoffmann, A. Tixier, Calibration of a two-dimensional numerical model for the optimization of locos-type isolations by response surface methodology, IEEE Trans. Semiconductor Manufacturing 13 (4) (2000) 416–426.

[20] H. Stippel, Ph. Lindorfer, J. Weintraub, Response surface method for statistical process and device simulation, in: First International Workshop on Statistical Metrology, Honolulu, 1996.

[21] Y.-F. Tsai, R.T. Farouki, Algorithm 812: Bpoly: an object-oriented library of numerical algorithms for polynomials in Bernstein form, ACM Trans. Mathematical Software 27 (2) (2001) 267–296.