Automatic Differentiation Technique in Device Model Parameter Extraction

Binjie Cheng, Zhibiao Shao, Liping Wang, Tiantong Tang and Zhong Yu

Dept. of Electronic Engineering, School of Electronic and Information Engineering,
Xi'an Jiaotong University, Xi'an, Shaanxi Province, Post Code 710049,
P. R. China, E-mail: bj-cheng@263.net

ABSTRACT

Automatic Differentiation (AD), based on the nonstandard analysis theory, is a new technique in computer numerical analysis. An AD based algorithm for device model parameter extraction is presented. Using this algorithm, the constrained parameter extraction for a surface potential based MOSFET drain current model was done, and the average relative error between calculated and measured current is less than 2%.

Keywords: Automatic Differentiation; Parameter Extraction; Object Oriented Programming; MOSFET Model.

1 INTRODUCE

Semiconductor device modeling and parameter extraction is one of key works in Electronic Design Automatic (EDA) area. The device models used in EDA usually have analytic expression for acceptable computational efficiency. When the device dimensions are scaled down to submicrometer regime, it is impossible to establish analytic model strictly based on device physics, therefor, some empirical parameters are introduced to describe narrow and short channel effects. These empirical parameters often have ambiguous physical meanings, and is difficult to extracted by experiment method. So they have to be extracted by optimization methods.

In general, for a problem that can be describe by analytic expression (e.g., the device model parameter extraction), the differentiation-dependent methods are more effective than some nonnumerical optimization methods such as genetic algorithms. However, the objective functions in practice are complex, and the analytic expressions of their gradients are rather cumbersome, to say nothing of their Hessian. In convention, it is implemented by numerical differentiation. This brings errors inevitably, and will affect the convergence of optimization.

Automatic Differentiation (AD) is a promising method in the field of computer numerical analysis. It is based on the nonstandard analysis and pronumber algebra, which has the distinctive properties that transforms the differentiation to arithmetic operation [1]. Using AD, the gradient and Hessian matrix of the objective function can be accurately obtained without derivation. In this paper, an algorithm base on AD technology for constrained optimization in device model parameter extraction is presented. Using this algorithm, the constrained parameter extraction for a surface potential based MOSFET drain current model was done, and encouraging result is obtained.

2 PRINCIPLES OF AD

For any smooth function, $f : R^{v} \blacklozenge\ R$, we can define its "*prolongation*" as

$$\hat{f} = \left\langle f,\ f,\quad f,\qquad f \cdots \right\rangle \tag{1}$$

where the first member of $\hat{f}$ is the function itself, the second is its gradient f, the third is its Hessian f, and so forth. Let $\underline{x}$ be a point in R^{v} space. Evaluating a prolongation at this point yields a so-called "*pronumber*"

$$\hat{a} = \hat{f}(\underline{x}) = \left\langle f(\underline{x}),\ f(\underline{x}),\quad f(\underline{x}),\qquad f(\underline{x}), \cdots \right\rangle = \left\langle a, \underline{a}, \underline{\underline{a}}, \cdots \right\rangle \tag{2}$$

We can define some operations on the prolongation as real numbers, under the rule that the corresponding operations of the smooth function and its derivatives are correct. Let the symbol ⊗ represent a kind of binary arithmetic operator of smooth functions, such as addition and multiplication, then the operation of prolongation should be defined as

$$\hat{f}\quad \hat{g} = f\ \hat{}\ g = \left\langle f\quad g,\ (f\quad g),\quad (f\quad g), \cdots \right\rangle \tag{3}$$

Correspondingly, we can define the arithmetic operations on the pronumbers. It is proved that pronumbers and their operations follow the commutative law, associative law and distributive law as real numbers. In practice, the pronumber algebra must be truncated to finite order. In the truncated algebra, operations as above can also be defined and they follow the three laws as well.

There are "zero" and "unit" in this algebra:

$$\hat{0} = \left\langle 0, \underline{0}, \underline{\underline{0}}, \cdots \right\rangle \qquad \hat{1} = \left\langle 1, \underline{0}, \underline{\underline{0}}, \cdots \right\rangle \tag{4}$$

Any real r can be mapped into this algebra as

$$\hat{r} = \left\langle r, \underline{0}, \underline{\underline{0}}, \cdots \right\rangle \tag{5}$$

Note that ordering can also be introduced in this algebra. For a pronumber whose first component is 0 and not equal to $\hat{0}$, $(0,\underline{a},\underline{\underline{a}}\cdots)$, has the following property:

$$\hat{0} < (0,\underline{a},\underline{\underline{a}}\cdots) < \hat{r} \text{ or } -\hat{r} < (0,\underline{a},\underline{\underline{a}}\cdots) < \hat{0} \text{ for } r > 0 \tag{6}$$

Such pronumbers are called *infinitesimal*, or *differential*. According to the multiplication of pronumbers, power higher than n of an infinitesimal is zero:

$$(0,\underline{a},\underline{\underline{a}}\cdots)^m = \hat{0} \quad \text{for} \quad m > n \tag{7}$$

This is an important property of infinitesimal, which is called *nilpotent*. It is this that the pornumber algebra is "nonstandard". It is notable that infinitesimal elements have no reciprocal, therefore the pronumber algebra is not a complete space, much less a field, but a subspace of $^*R$ in nonstandard analysis.

Differentiation can also be defined as arithmetic operation in this algebra. The fundamental functions, such as exponent, logarithm, trigonometric functions, can be transformed into the pronumber algebra, either by series expansion or by the differentiation rule of composite function.

The pronumber algebra provides the mathematical principle for AD technique. Its special properties are: (1) a pronumber consists of not only the value of a function but also its derivatives of orders as high as desired; (2) differentiation is defined as arithmetic operation of pronumbers.

3 AD PACKAGE IN VISUAL C++

The object-oriented languages, with excellent properties such as data hiding, inheritance and extensibility, present a natural environment for implementation of AD [2]. An AD package is developed in Visual C++. In this package we defined a class, *ADS*, for pronumbers. The private attributes of *ADS* are *double, dbVecto*r and *dbMatrix. dbVector* and *dbMatrix* are classes that we defined for vector and matrix, which include some special operations needed. Some operators as +, -, *, /, and functions as *sqrt*, *exp*, *log*, *pow*, *sin*, *cos*, *tan*, *asin*, *acos* and *atan* are overloaded, so that the application programmers can use these familiar symbols not only on real numbers but also on pronumbers or between the two types. There is an important function, *set_variable*, which is used to set the variables of the system to *ADS* type, that is:

$$\hat{x}_j = \langle x_j, \underline{e}_j, \underline{\underline{0}} \rangle \tag{8}$$

where $\underline{e}_j$ represents a vector whose components are δ_{kj}, and $\underline{\underline{0}}$ represents a matrix whose components are all zero. Moreover, for facilitation, several constructors are overloaded.

This package is widely applicable because of the unique virtues of AD and the object oriented properties of Visual C++. The only requirement for the user is to write the following

#include "ads.h"

in his application program and change some quantities to *ADS* type. It is noticeable that this implementation of AD is very terse, facilitate, and user-friendly.

4 EXTRACTION ALGORITHM BASED ON AD

Here we present an extraction algorithm based on AD. Because the gradient and Hessian matrix of the objective function can be accurately obtained using AD technology, this algorithm is directly based on Newton method. In practical, the Newton method must be modified in order to insure convergence. The former modifications are usually concerned about how to make Hessian matrix positive-definite [3]. However, Hessian matrix being positive-definite is the necessary condition for global convergence. This is not always necessary since there is often a feasible region in practical extraction problem. So it is unnecessary to modify Newton method for global convergence. Our modification principle is to find steepest decent direction at every iterative point. In our algorithm, the iteration is

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left[\ ^2 f(\mathbf{x}^k) + \lambda^k \mathbf{I} \right]^{-1} \ f(\mathbf{x}^k) \tag{9}$$

where $\mathbf{I}$ is the unit diagonal matrix, λ is a factor to be determined by linear search. This is similar to L-M method.

Generally, the model parameter extraction is a kind of constrained optimization problem as

$$\text{(IP)} \quad \min\{f(\mathbf{x}) | \mathbf{g}(\mathbf{x}) ? \ 0\} \tag{10}$$

The Rockafellar multiplier is applied to deal with this problem. The improved Lagrange objective function is

$$\tilde{F}(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2p_k} \sum_{i=1}^{m} \left\{ [\max(0, \eta_i - p_k g_i(\mathbf{x}))]^2 - \eta_i^2 \right\} \quad \{p_k\} - +\times \tag{11}$$

where p_k is the k-th penalty factor, $|\zeta$ is Lagrange multiplier_For model parameters, the constraints are generally written as

$$\mathbf{m}_2 \ ? \ \mathbf{c}(\mathbf{x}) \ ? \ \mathbf{m}_1 \tag{12}$$

where $\mathbf{m}_1$ and $\mathbf{m}_2$ stand for the lower and the upper bounds given on the model parameters. For this "box constraint", $\mathbf{g}(\mathbf{x})$ in eq.(10) can be constructed as

$$\mathbf{g}(\mathbf{x})=\left(\frac{\mathbf{m}_2-\mathbf{m}_1}{2}\right)^2-\left(\mathbf{c}(\mathbf{x})-\frac{\mathbf{m}_1+\mathbf{m}_2}{2}\right)^2 ? 0 \qquad (13)$$

The algorithm proceeds as follows:

1. Define the model parameters to be extracted as ADS variable $\hat{\mathbf{x}}$ and initialize them; Initialize Lagrange multiplier; Set constraint condition, incremental series of the penalty factor $\{p_k\}$, improved Lagrange objective $\hat{\tilde{F}}$ _tolerance ε and maximum iteration times n.
2. Using pronumbers' algebraic laws, get $\hat{\tilde{F}}=\tilde{F}(\hat{\mathbf{x}}^k)$. And then get $\tilde{F}(\mathbf{x}^k)$, $\tilde{F}(\mathbf{x}^k)$ and $^2\tilde{F}(\mathbf{x}^k)$ according to the definition of pronumbers.
3. Determine the new Lagrange multiplier as follow: $|Ç^{k+1}=\max(\mathbf{0},|Ç^k-p_k\mathbf{g}(\mathbf{x}^k))$.
4. Let $\mathbf{x}^*=\mathbf{x}^k-[\,^2\tilde{F}(\mathbf{x}^k)+\lambda\mathbf{I}]^{-1}\ \tilde{F}(\mathbf{x}^k)$, and using linear search method, solve $\lambda=\arg\min_{\lambda}\tilde{F}(\mathbf{x}^*)=\arg\min_{\lambda}\tilde{F}(\lambda,\mathbf{x}^k)$
5. $\mathbf{x}^{k+1}\quad\mathbf{x}^*$. If $\tilde{F}(\mathbf{x}^{k+1})\varepsilon$, terminate the calculation; Else, $k\quad k+1$.
6. If $k>n$, exit; Else, transform $\mathbf{x}^k$ into ADS variable $\hat{\mathbf{x}}^k$, and go to step 2

It is notable that AD technique is merged in this algorithm. This algorithm is implemented by Object Oriented Programming technique.

5 TEST MODEL & PARAMETER EXTRACTION RESULT

In order to prove the effect of this AD-based algorithm, a constrained parameter extraction for a surface-potential based MOSFET drain current model was done. For this kind of models, iterative solution of surface potential is needed, therefor the parameter extraction of this kind of models is more difficult than of threshold based model. However, the surface-potential based model is continuous inherently and it constructs compact model easily.

5.1 Surface-Potential based model

This model is based on the fundamental equation of surface potential, in order to consider the short channel effects such as velocity saturation, drain induced barrier lowering and the source/drain resistance, some empirical parameters are applied. The drain current expression is

$$I_{ds}=C_{ox}\frac{W-\Delta W}{L_{eff}}\frac{\mu_{eff}}{1+0.1r_d q_i}\left\{\left(V_g+0.1a_d\big/\sqrt{V_g}\right)\sqrt{V_{ds}}-V_{fb}\right\}(\varphi_d-\varphi_s)$$
$$-0.5\left(\varphi_d^{\,2}-\varphi_s^{\,2}\right)-\frac{2}{3}\gamma\left(\varphi_d^{\,3/2}-\varphi_s^{\,3/2}\right)+\phi_t(\varphi_d-\varphi_s)+\phi_t\gamma\left(\sqrt{\phi_d}-\sqrt{\phi_s}\right)\} \qquad (14)$$

where w is width of channel, C_{ox} is gate-oxide capacitance, V_g is gate voltage, V_{ds} is drain voltage, V_{fb} is flat band voltage, $r=\sqrt{2q\varepsilon_{si}N_b}\big/C_{ox}$ (N_b is substrate concentration), ϕ_t is thermal voltage, and φ_s and φ_d is source/drain surface voltage:

$$\varphi_s=V_g-V_{fb}-\gamma\sqrt{\varphi_s+\phi_t\exp((\varphi_s-2\phi_F)/\phi_t)} \qquad (15)$$

$$\varphi_d=V_g-V_{fb}-\gamma\sqrt{\varphi_d+\phi_t\exp((\varphi_d-2\phi_F-V_{ds})/\phi_t)} \qquad (16)$$

where $\phi_F=\phi_t\ln(N_{sur}/n_i)$, N_{sur} is surface concentration. Source/drain surface potential must be solved by an iterative procedure.

L_{eff} is the effective channel length

$$L_{eff}=L-\Delta l-0.1a_l\sqrt{V_{ds}/\varphi_d} \qquad (17)$$

q_i is a factor considering source/drain resistance

$$q_i=(V_g-V_T)\sqrt{(\varphi_d-\varphi_s)} \qquad (18)$$

where $V_T=V_{fb}+2\phi_F+\gamma\sqrt{2\phi_F}$.

μ_{eff} is effective mobility

$$\mu_{eff}=\mu\big/\left(1+0.1a_r(V_g-V_T)+\sqrt{E_{nv}E_{lv}/E_{\max}}\right) \qquad (19)$$

where E_n and E_{lv} are longitudinal and transverse average electrostatic field respectively.

There are eleven parameters: ΔW, ΔL, μ, V_{fb}, N_{sur}, N_b, $E_{\max}$, a_l, a_d, a_r and r_d. Among these, V_{fb}, N_{sur} and N_b appear in the implicit equation and I-V equation simultaneously.

5.2 Result of Parameter Extraction

The experimental data of $0.8\mu m$ NMOSFET is used to extract model parameter. Its fundamental physics parameters are $L = 0.8\mu m$ _$W = 20\mu m$ and $T_{ox} = 0.022\mu m$. The objective function is of least-square type, and the constraint conditions are as follow (using the same units as in Table 1):

$$0.95 \; ? \; \mu \; ? \; 0.45$$
$$4.0 \; ? \; E_{\max} > 0$$
$$0.25 \; ? \; \Delta L > 0$$
$$a_l > 0, a_d > 0, a_r > 0, r_d > 0$$
$$1.0 \; ? \; V_T \; ? \; 0.6$$

(20)

Table 1: The result of the parameter extraction

k	Value of objective function	λ	ΔL (μm)	μ (10^3cm^2/v-s)	V_{fb} (V)	N_{sur} (10^{16}cm^{-3})	N_b (10^{16}cm^{-3})	$E_{\max}$ (10^3v/cm)	a_l	a_d	a_r	r_d	V_T (V)
0	257.005	0	0.15	0.8	-0.2	10	1	2.5	1	1	1	1	0.9538
1	5.92987	65.248	0.1857	0.6552	0.037	10.334	0.9299	1.6946	1.3487	0.5713	0.3812	0.6368	1.1807
2	2.3011	19.039	0.1861	0.6588	-0.16	10.182	0.9071	1.6924	1.3536	0.27	0.3865	0.4904	0.9825
3	1.2313	1.7167	0.1849	0.6069	-0.4	9.3463	1.3815	1.6213	1.2791	0.2411	0.4149	0.3314	0.8062
4	1.0037	0.1913	0.1387	0.6242	-0.57	8.7884	1.6045	1.6178	1.7311	0.095	0.6319	0.3116	0.6617
5	0.8109	0.7639	0.1427	0.5997	-0.65	6.6503	2.0628	1.5395	1.8188	0.099	0.5732	0.2536	0.6197
6	0.7338	0.8328	0.1541	0.5533	-0.72	6.1633	2.228	1.5135	1.9021	0.0721	0.3653	0.2157	0.5652
7	0.692	0.2492	0.1654	0.551	-0.84	8.8475	2.9521	1.4821	1.9248	0.0709	0.4074	0.1926	0.5504
8	0.5951	0.0688	0.1822	0.5537	-0.82	8.5795	3.2609	1.4861	1.7051	0.1126	0.4424	0.1834	0.5885
9	0.5217	0.06888	0.1787	0.5781	-0.87	6.371	4.1015	1.6406	1.594	0.1422	0.7255	0.1904	0.5961
10	0.51	0.7639	0.1777	0.593	-0.86	6.1365	4.191	1.5415	1.588	0.1469	0.7665	0.1801	0.6132

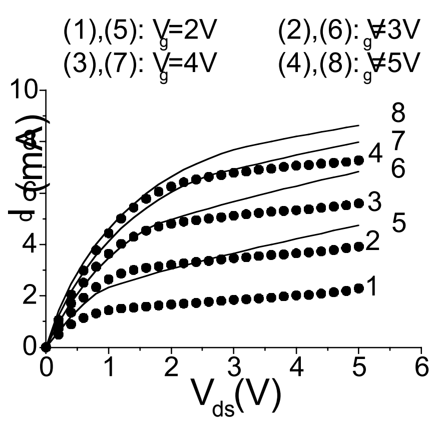

Fig.1: The initial curve of model and measured data
(Note: the dots in Fig.1 are measured data)

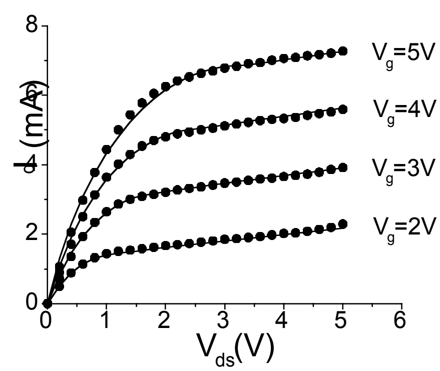

Fig.2: The optimized curve of model and measured data
(Note: the dots in Fig.2 are measured data)

It is shown that the convergence of this algorithm is fast when the value of objective function is large; the algorithm still has the tendency to converge when the parameters return to feasible region. We define the average relative error of the parameter extraction as

$$error = \sum_{i=1}^{N} \left(\left| I^i_{mea} - I^i_{\mathrm{mod}} \right| / I^i_{mea} \right) / N$$

(22)

where N is the number of the iteration times, I_{mea} is the measured current, I_{mod} is the current calculated from the model. For the present extraction, the average relative error is about 1.97%.

6 CONCLUSION

AD is a promising technique based on the nonstandard analysis and pronumber algebra. An AD based constrained optimization algorithm for device model parameter extraction is presented. In this algorithm, a modified Newton method is adopted, the constrained conditions are dealt with by Rockafellar multiplier, and the AD technology are merged in the optimization procedure entirely. The distinctive features of this algorithm are: (1) Using AD technology, the gradient and Hessian matrix of the objective function can be accurately obtained without formulae derivation; (2) Taking advantage of the Hessian information, this algorithm can achieve rather fast convergency even when the initial values are not well; (3) The algorithm can extract all parameters in a run without manual intervention, and the only requirement is to give initial values and constraint conditions of the parameters. By this algorithm, a constrained parameter extraction for a surface-potential-based MOSFET drain current model was done. And the result is encouraging.

REFERENCES

[1] Robinson A, “Nonstandard Analysis,” In: Proceedings of the Royal Academy of Sciences Ser.A. Amsterdam: North-Holland, 64, 432~440, 1961.

[2] Michelotti L, In: Proceedings of the 1989 IEEE Particle Accelerator Conference, 839~841, 1989.

[3] P. E. Gill, W. Murry, M. H. Wright, "Practical Optimization," Academic Press, New York, 1981.