Journal of Computational Mathematics, Vol.21, No.1, 2003, 85–100.

MULTI-SCALE METHODS FOR INVERSE MODELING IN 1-D MOS CAPACITOR*[1)]

Pingwen Zhang[†] Yi Sun[‡] Haiyan Jiang[♯]
(*School of Mathematical Sciences, Peking University, Beijing, 100871, People's Republic of China*)
Wei Yao[♮]
(*China Predictive Technology Laboratory, DigitalDNA Lab.-China, Motorola (China) Electronics Ltd., Beijing, 100022, People's Republic of China*)

Dedicated to the 80th birthday of Professor Zhou Yulin

Abstract

In this paper, we investigate multi-scale methods for the inverse modeling in 1-D Metal-Oxide-Silicon (MOS) capacitor. First, the mathematical model of the device is given and the numerical simulation for the forward problem of the model is implemented using finite element method with adaptive moving mesh. Then numerical analysis of these parameters in the model for the inverse problem is presented. Some matrix analysis tools are applied to explore the parameters' sensitivities. And third, the parameters are extracted using Levenberg-Marquardt optimization method. The essential difficulty arises from the effect of multi-scale physical difference of the parameters. We explore the relationship between the parameters' sensitivities and the sequence for optimization, which can seriously affect the final inverse modeling results. An optimal sequence can efficiently overcome the multi-scale problem of these parameters. Numerical experiments show the efficiency of the proposed methods.

Key words: Inverse problem, MOS capacitor model, Finite element method, Adaptive moving mesh, Levenberg-Marquardt method, Sequence for optimization, Multi-scale methods.

1. Introduction

Metal Oxide Silicon (MOS) transistors are the basic building block of MOS integrated circuits. Very Large Scale Integrated (VLSI) circuits using MOS technology have emerged as the dominant technology in semiconductor industry. The direct techniques for determining the two-dimensional doping profiles, such as scanning capacitance microscopy and dopant selective etching followed by atomic force microscopy (DSE/AFM), however, are less mature at the moment. The device have to be destroyed in these direct techniques. In mathematical theory, present implementation of technique does not determine unique physical solution and requires excessive intervention to achieve acceptable results because of the high nonlinear character of the mathematical model. Therefore much work for the design and analysis of circuits focuses on scientific computing of the numerical simulation and the inverse modeling.

Since the accuracy of a device model in predicting device characteristics is fully dependent on the accuracy of the model parameter values being used, *parameters extraction*, an electronics inverse problem, is one of most important aspects of the semiconductor industry. It

* Received September 30, 2002.

[1)]This project is supported by Motorola (China) Electronics Ltd. and the work of Pingwen Zhang is also partially supported by Special Funds for Major State Basic Research Projects of China G1999032804.

† Email address: pzhang@math.pku.edu.cn

‡ Email address: ysun@pku.edu.cn

♯ Email address: hyjiang@163.com

♮ Email address: r52015@email.sps.mot.com

is to characterize the complicated doping profiles inside the device by curve fitting the model equations to a set of the data measured on its surface, such as the capacitance-voltage (C-V) and/or current-voltage (I-V) data using nonlinear least square optimization techniques. This is still a very challenging problem because of the lack of effective optimization methods or ideas. Especially in practical problems, with ever decreasing device dimensions, the complexity of the models used in circuit simulators have increased significantly and the physical scales of these parameters often are various. Moreover, the effect of multi-scale physical difference of the parameters is so serious that we usually can extract only part of the parameters.

The main purpose of this paper is to propose some numerical methods or strategies for solving the inverse problem of parameters extraction in 1-D MOS capacitor using C-V technique, particularly the difficulty in the multi-scale problem, based on our sensitivity analysis of the parameters.

In section 2, we will formulate the mathematical model of 1-D MOS capacitor. The model is a nonlinear two-point boundary value problem. In section 3 we use finite element method with adaptive moving mesh in the discretization of the forward problem. Then sensitivity analysis of the parameters is given in section 4. And section 5 outlines the algorithm used to solve the inverse problem. Section 6 and 7 present the results and concluding remarks. Furthermore, we will give different results based on two kinds of physical MOS models: *low-frequency capacitance* model and *deep depletion* model.

2. The Mathematical Model of 1-D MOS Capacitor

Figure 1 represents a simple structure of 1-D MOS capacitor, which consists of *Poly-Si* layer (metal), SiO_2 layer (oxide) and *Si* layer (silicon) from the left to the right.

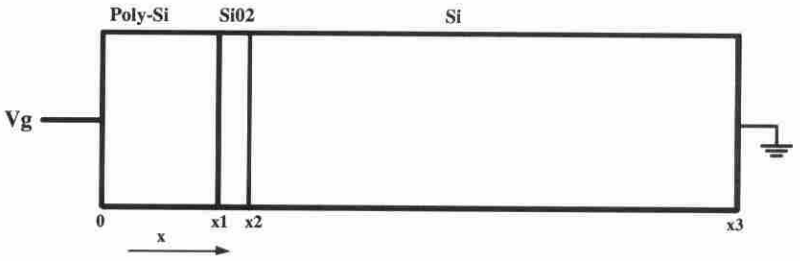

Figure 1: The structure of MOS capacitor.

A fundamental equation in MOS structure is Possion's equation, which relates the charge density ρ to the electrical potential ϕ. The formulation is

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\varepsilon(x)\frac{\mathrm{d}\phi(x)}{\mathrm{d}x}\right) = -\rho(\phi(x), x), \tag{2.1}$$

where x is the depth from the left boundary of the device, $\phi(x)$ is the electrical potential at x, $\rho(x)$ is the charge density, $\varepsilon(x)$ is the material's dielectric number. And $\varepsilon(x)$ in the three layers are constants $\varepsilon_{\mathrm{si}}$, $\varepsilon_{\mathrm{ox}}$ and $\varepsilon_{\mathrm{si}}$ respectively. And the boundary conditions

$$\begin{cases} \phi|_{x=0} = \phi_0 + \phi_{Bpoly} = \phi_0 + \dfrac{\mathrm{kT}}{\mathrm{q}}\ln\left(\dfrac{N_{0poly}}{\mathrm{n_i}}\right) \\ \phi|_{x=x_3} = \phi_B = -\dfrac{\mathrm{kT}}{\mathrm{q}}\ln\left(\dfrac{N_{const}}{\mathrm{n_i}}\right). \end{cases} \tag{2.2}$$

The charge density:

$$\rho(\phi(x), x) = \begin{cases} \mathrm{q}\left[\mathrm{n_i}\exp\left(\dfrac{-\mathrm{q}(\phi(x)-\phi_0)}{\mathrm{kT}}\right) - \mathrm{n_i}\exp\left(\dfrac{\mathrm{q}(\phi(x)-\phi_0)}{\mathrm{kT}}\right) + N_{poly}(x)\right] & \text{in } Poly\text{-}Si \text{ layer} \\ \rho_{\mathrm{fix}} - \mathrm{Q_{it}}\delta(x_2) & \text{in } SiO_2 \text{ layer} \\ \mathrm{q}\left[\mathrm{n_i}\exp\left(\dfrac{-\mathrm{q}\phi(x)}{\mathrm{kT}}\right) - \mathrm{n_i}\exp\left(\dfrac{\mathrm{q}\phi(x)}{\mathrm{kT}}\right) - N_{si}(x)\right] & \text{in } Si \text{ layer,} \end{cases} \tag{2.3}$$

where $N_{poly}(x)$ is the gate doping profile in *Poly-Si* layer. Exponential function is selected to represent the doping profile in the layer.

$$N_{poly}(x) = N_{0poly}\left[1 - \exp\left(\frac{x - \mathrm{t_{poly}}}{\sigma_1}\right)\right] \tag{2.4}$$

and $N_{si}(x)$ is the substrate doping profile in Si layer. Here we use *Gaussian* function to describe the doping profile:

$$N_{si}(x) = N_{0si}\exp\left[-\left(\frac{x - x_0}{\sqrt{2}\sigma_2}\right)^2\right] + N_{const}. \tag{2.5}$$

The other fundamental equations in our problem are about the capacitances in MOS. The MOS gate capacitance C_g equals the series combination of the capacitances of C_{poly}, C_{sio_2} and C_{si}:

$$\frac{1}{C_g} = \frac{1}{C_{poly}} + \frac{1}{C_{sio_2}} + \frac{1}{C_{si}}, \tag{2.6}$$

where

$$\begin{cases} C_{poly} &= \dfrac{\mathrm{d}Q_{poly}}{\mathrm{d}U_{poly}} \\ C_{sio_2} &= \dfrac{\varepsilon_{\mathrm{ox}}}{t_{sio_2}} \\ C_{si} &= \dfrac{\mathrm{d}Q_{si}}{\mathrm{d}U_{si}}, \end{cases} \tag{2.7}$$

in which Q_{poly} and Q_{si} are the electric charge, U_{poly} and U_{si} are the potential differences of the *Poly-Si* layer and Si layer respectively. Here we use the definitions of differential capacitance (the ratio of the variation in charge to the corresponding variation in voltage) to compute C_{poly} and C_{si}, which are more important than those of static capacitance (the ratio of total charge to total voltage) because the capacitance of an MOS structure is a nonlinear function of voltage (see chapter 4 in [3]).

In fact, what we can measure is the gate voltage V_g. It has the relationship with ϕ_0:

$$\phi_0 = V_g - \phi_{ms}, \tag{2.8}$$

in which ϕ_{ms} is the work function difference: $\phi_{ms} = W_f - \chi$, where W_f is the work function, χ is the semiconductor's electron affinity.

Here we give out other known parameters' physical meanings in the above equations:

- k is *Boltzman* constant;
- T is room temperature;
- $\mathrm{Q_{it}}$ is the interface-trapped charge density;
- $\mathrm{n_i}$ is intrinsic carrier concentration;
- q is electron charge;
- ρ_{fix} is fixed-oxide charge density;
- $\delta(x)$ is *Delta* function at x;
- $\mathrm{t_{poly}}$, $\mathrm{t_{si}}$ are thickness of the two layers.

The unknown parameters that are to be extracted are:

$$\left.\begin{array}{l} 1)N_{0poly}; \\ 2)\sigma_1; \end{array}\right\} \text{in the gate doping profile } N_{poly}(x)$$

$$\left.\begin{array}{l} 3)N_{const}; \\ 4)N_{0si}; \\ 5)x_0; \\ 6)\sigma_2; \end{array}\right\} \text{in the substrate doping profile } N_{si}(x)$$

$7)t_{sio_2}$: the thickness of the SiO_2 layer;
$8)W_f$: the work function.

3. The Forward Problem

Our forward problem is that given a gate voltage V_g on the left boundary of MOS, to determine the corresponding gate capacitance C_g of MOS. The first step is to calculate the distribution of the electrical potential ϕ by solving (2.1) and (2.2). The second step is to get C_g with (2.6) and (2.7).

Equations (2.1) and (2.3) show that we face a two-point boundary value problem with very strong nonlinear character. We use finite element method with adaptive moving mesh to solve it.

After finite element discretization, we can get an nonlinear algebraic equation:

$$A\Psi = F(\Psi) \tag{3.1}$$

where Ψ is the vector of ϕ_i for $i = 1, 2, \ldots, n$. Moreover, the coefficient matrix A is diagonally dominant tridiagonal. The classical iterative Newton-Raphson method is used to solve (3.1) (see [5]).

In the second step, we find the fact that the distribution of voltage has a continuous dependence on the boundary value ϕ_0 is useful to get gate capacitance C_g. So by differentiating equations (2.1) and (2.2) with respect to ϕ_0 (for simplification, we use the denotation $V = \phi_0$), a new equation about ϕ_V can be obtained

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}x}\left(\varepsilon(x)\dfrac{\mathrm{d}\phi_V(x)}{\mathrm{d}x}\right) = -\Phi(\phi(x), \phi_V(x), x) \\ \phi_V|_{x=0} = 1 \\ \phi_V|_{x=x_3} = 0, \end{cases} \tag{3.2}$$

where $\Phi(\phi(x), \phi_V(x), x) = \dfrac{\mathrm{d}\rho(\phi(x), x)}{\mathrm{d}V}$, and

$$\frac{\mathrm{d}\rho(\phi(x), x)}{\mathrm{d}V} = \begin{cases} -\dfrac{\mathrm{q}^2\mathrm{n_i}}{\mathrm{kT}}\left[\exp\left(\dfrac{-\mathrm{q}(\phi(x)-\phi_0)}{\mathrm{kT}}\right) + \exp\left(\dfrac{\mathrm{q}(\phi(x)-\phi_0)}{\mathrm{kT}}\right)\right](\phi_V(x)-1) & \text{in } \textit{Poly-Si} \text{ layer} \\ 0 & \text{in } SiO_2 \text{ layer} \\ -\dfrac{\mathrm{q}^2\mathrm{n_i}}{\mathrm{kT}}\left[\exp\left(\dfrac{-\mathrm{q}\phi(x)}{\mathrm{kT}}\right) + \exp\left(\dfrac{\mathrm{q}\phi(x)}{\mathrm{kT}}\right)\right]\phi_V(x) & \text{in } \textit{Si} \text{ layer,} \end{cases} \tag{3.3}$$

Obviously, $\Phi(\phi(x), \phi_V(x), x)$ is a linear function of $\phi_V(x)$. Then after solving this linear two-point boundary value problem, we can get the distribution of ϕ_V.

According to Gauss's law in electronics,

$$Q = \varepsilon E = \varepsilon \frac{\mathrm{d}U}{\mathrm{d}x}, \tag{3.4}$$

in which E is electric-field intensity, U is the potential difference on the device, from (2.7) we can compute the capacitance as follows:

$$C_{poly} = \frac{\mathrm{d}Q_{poly}}{\mathrm{d}U_{poly}} = \frac{\mathrm{d}Q_{poly}/\mathrm{d}V}{\mathrm{d}U_{poly}/\mathrm{d}V} = \frac{\varepsilon_{si}\mathrm{d}\left(\dfrac{\mathrm{d}U_{poly}}{\mathrm{d}x}\right)\Big/\mathrm{d}V}{\mathrm{d}U_{poly}/\mathrm{d}V} = \frac{\varepsilon_{si}\mathrm{d}\left(\dfrac{\mathrm{d}U_{poly}}{\mathrm{d}V}\right)\Big/\mathrm{d}x}{\mathrm{d}U_{poly}/\mathrm{d}V}$$

and

$$\mathrm{d}U_{poly}/\mathrm{d}V = \mathrm{d}(\phi(x)|_{x=x_0} - \phi(x)|_{x=x_1})/\mathrm{d}V = \phi_V(x)|_{x=x_0} - \phi_V(x)|_{x=x_1},$$

then we have

$$C_{poly} = \frac{\varepsilon_{si}\left(\left.\dfrac{\mathrm{d}\phi_V}{dx}\right|_{x=x_0} - \left.\dfrac{\mathrm{d}\phi_V}{\mathrm{d}x}\right|_{x=x_1}\right)}{\phi_V(x)|_{x=x_0} - \phi_V(x)|_{x=x_1}} = \frac{\displaystyle\int_{x_1}^{x_0} (-\Phi(\phi(x), \phi_V(x), x))\mathrm{d}x}{1 - \phi_V(x_1)}, \tag{3.5}$$

similarly,

$$C_{si} = -\frac{\displaystyle\int_{x_3}^{x_2} (-\Phi(\phi(x), \phi_V(x), x))\mathrm{d}x}{\phi_V(x_2) - \phi_V(x_3)}, \tag{3.6}$$

where the negative sign before the fraction in the right-hand side comes from the fact that Q_{si} is negative charge. And $C_{sio_2} = \varepsilon_{\mathrm{ox}}/t_{sio_2}$ can be obtained easily if t_{sio_2} is known. So with (2.6) we can compute gate capacitance C_g. Hence, we finish the forward problem.

During the inversion, calculation of the forward problem is carried out thousands of time, therefore, it is imperative that the forward model be both computationally efficient and accurate. As we know, there is a sharp layer of electrical potential in the SiO_2 layer, see Figure 2. If we use the uniform mesh, it must cost more to resolve in the SiO_2 layer. Furthermore, the mesh will not be smooth if more meshes are added in the SiO_2 layer and it will lose accuracy. We have a good idea to create an adaptive mesh for the finite element method. There will be the same accuracy for the uniform mesh only using 1/10 mesh for adaptive mesh. This method can increase the accuracy of the numerical approximations and also decrease the computational cost, see [6].

In Figure 3, we first compute a standard solution by setting uniform mesh $n_1 = 25000$, $n_2 = 2500$, $n_3 = 25000$, which are the numbers of grids in three layers respectively; 200 points of V_g changes from -2V to 2V.

1) $N_{0poly} = 1 \times 10^{21}\mathrm{cm}^{-3}$; 2) $\sigma_1 = 0.002\mu$m;
3) $N_{const} = 0.1 \times 10^{17}\mathrm{cm}^{-3}$; 4) $N_{0si} = 5 \times 10^{17}\mathrm{cm}^{-3}$;
5) $x_0 = 0.0\mu$m; 6) $\sigma_2 = 0.2\mu$m;
7) $t_{sio_2} = 25$Å; 8) $W_f = 4.2$V.

Figure 4 shows the absolute errors compared with the standard solution when we use less grids in uniform meshes. And Figure 5 show a good result about adaptive mesh. Here we use only 1000 points as the total number of all grids in three layers with adaptive technique. The solution with the technique, which is denoted by real line, attains much higher precision than that one using uniform mesh with 21000 points as total. The high efficiency of the adaptive mesh will save much computational time in the inverse problem while keeping high precision.

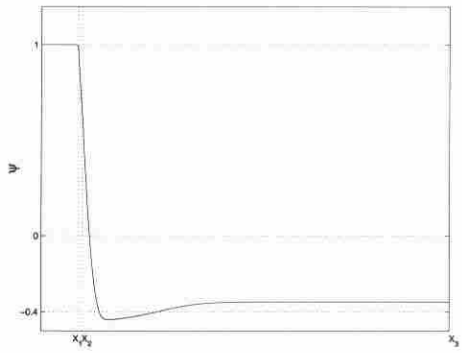

Figure 2: The distribution of electrical potential for $\phi|_{x=0} = 1.0$V.

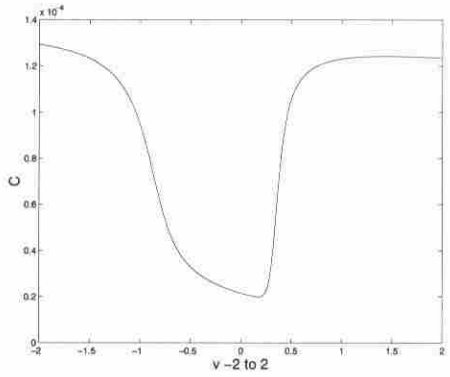

Figure 3: Forward problem: C-V curve.

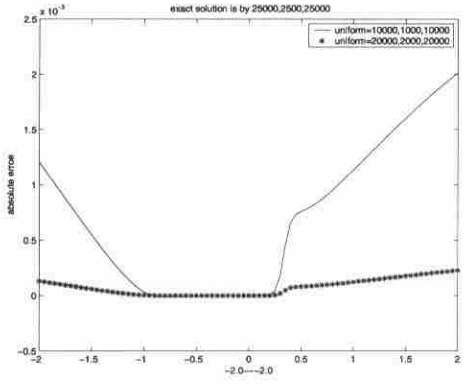

Figure 4: The error of the C-V using uniform mesh.

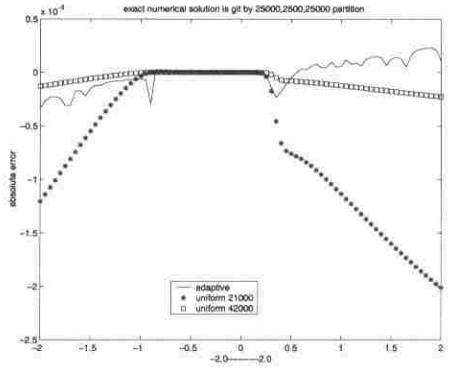

Figure 5: The error of the C-V using adaptive mesh.

4. Sensitivity Analysis

In this section, we do some numerical analysis about the eight parameters' sensitivities. These information will be very important to the inverse problem.

First, we use Figures 6-13 to show outwardly the sensitivities of the parameters. Figures 6 and 7 show that for the 1st and 2nd parameters N_{0poly} and σ_1, the shifts of the C-V curves mainly arise on the interval $[0, 2]$. Figures 8 shows that for the 3rd parameter N_{const}, the C-V curve changes very little on the whole interval $[-2, 2]$. Figures 9-11 show that for the 4th, 5th and 6th parameters N_{0si}, x_0 and σ_2, the shifts of the C-V curves mainly arise on the interval $[-1, 1]$. Figure 12 shows that the C-V curve changes on its two wings for the 7th parameter t_{sio_2}. Figure 13 shows that for the 8th parameter W_f, the change in the C-V curve is a horizontal parallel shift along the voltage axis. If W_f increases, the shift is to the right of the standard curve $W_f = 4.2$V. A decreasing W_f causes a shift to the left of the standard curve.

Now we give more refined numerical analysis as below with some matrix analysis tools. Related details of these tools can be found in any books on numerical algebra such as [4]. For simplification, we use x_j $(j = 1, \ldots, 8)$ to represent the eight parameters respectively, and

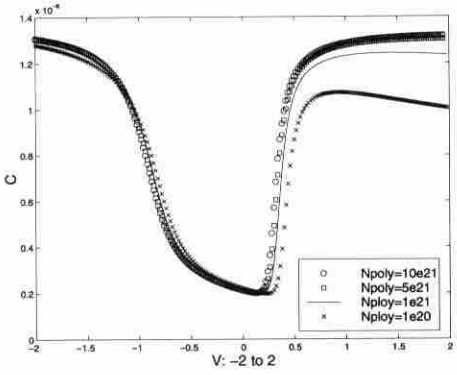

Figure 6: N_{0poly} = 1e20, 1e21, 5e21, 1e22 cm^{-3}.

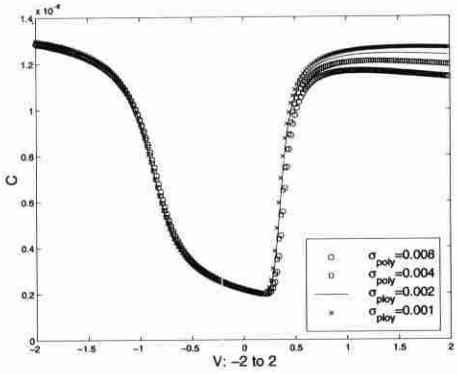

Figure 7: σ_1 = 0.001, 0.002, 0.004, 0.008 μm.

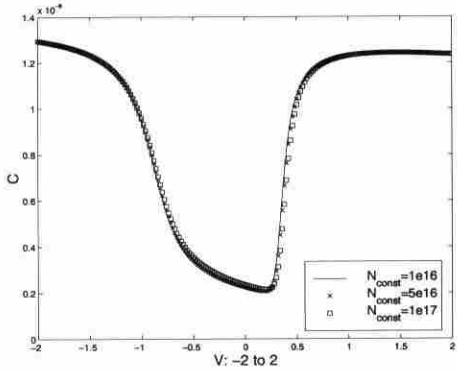

Figure 8: N_{const} = 1e16, 5e16, 1e17 cm^{-3}.

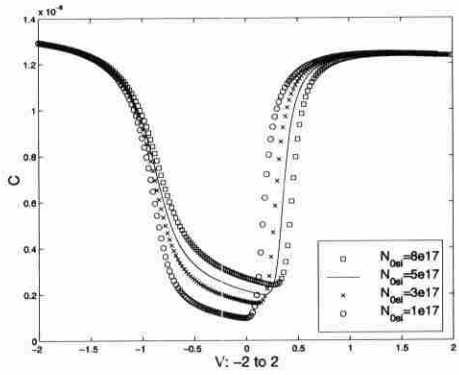

Figure 9: N_{0si} = 1e17, 3e17, 5e17, 8e17 cm^{-3}.

simplify the forward problem into:

$$\mathbf{C_g} = \mathbf{C}(x_1, x_2, \ldots, x_8, \mathbf{V_g}), \tag{4.1}$$

where $\mathbf{C_g}, \mathbf{C}, \mathbf{V_g} \in \mathbb{R}^m$. These m different V_g values are evenly selected from the V_g range $[-2, 2]$. They can be expressed as $Vg_i = -2 + 4i/m \ \ (i = 1, \ldots, m)$. The sensitivity of x_j will be obtained by analysing the jacobian of $\mathbf{C}$.

The jacobian is

$$J = \begin{pmatrix} \frac{\partial C_1}{\partial x_1} & \frac{\partial C_1}{\partial x_2} & \cdots & \frac{\partial C_1}{\partial x_8} \\ \frac{\partial C_2}{\partial x_1} & \frac{\partial C_2}{\partial x_2} & \cdots & \frac{\partial C_2}{\partial x_8} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial C_{m-1}}{\partial x_1} & \frac{\partial C_{m-1}}{\partial x_2} & \cdots & \frac{\partial C_{m-1}}{\partial x_8} \\ \frac{\partial C_m}{\partial x_1} & \frac{\partial C_m}{\partial x_2} & \cdots & \frac{\partial C_m}{\partial x_8} \end{pmatrix} \in \mathbb{R}^{m \times 8}, \tag{4.2}$$

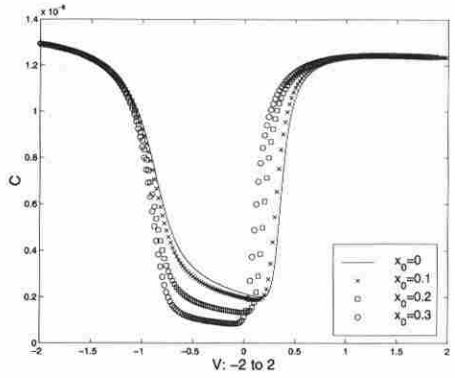

Figure 10: $x_0 = 0.0,\ 0.1,\ 0.2,\ 0.3\ \mu$m.

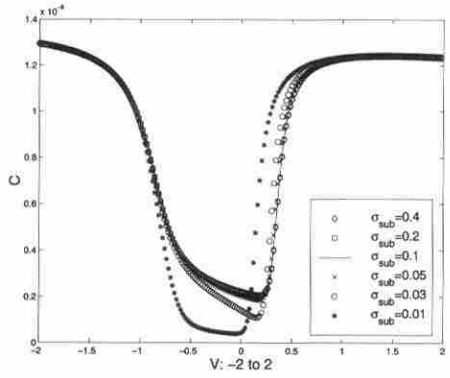

Figure 11: $\sigma_2 = 0.01, 0.03, 0.05, 0.1, 0.2, 0.4$ μm.

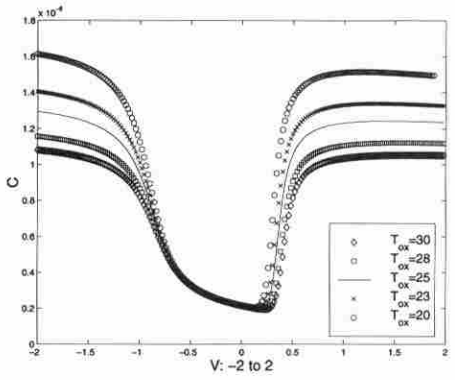

Figure 12: $t_{sio_2} = 20,\ 23,\ 25,\ 28,\ 30$ Å.

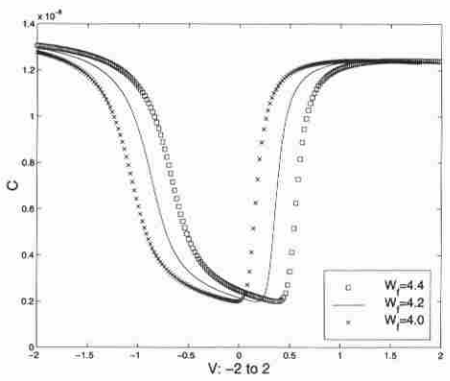

Figure 13: $W_f = 4.0,\ 4.2,\ 4.4$ V.

it is an m by 8 rectangular matrix. And we define the *sensitivity* of each parameter. It can be described by the values S_j that is the L_2 norm of every corresponding column in the jacobian:

$$S_j := \left[\sum_{i=1}^{m}\left(\frac{\partial C_i}{\partial x_j}\right)^2\right]^{\frac{1}{2}} \qquad j = 1, \ldots, 8, \tag{4.3}$$

where the derivate $\frac{\partial C_i}{\partial x_j}$ is obtained as follows:

$$\frac{\partial C_i}{\partial x_j} = \frac{C_i(x_j + \Delta x_j) - C_i(x_j)}{\Delta x_j} \qquad i = 1, \ldots, m;\quad j = 1, \ldots, 8, \tag{4.4}$$

where $\Delta x_j = \delta_j \cdot x_j$. To make the results comparable, we let all (V_g, C_g) values lose their units and take all $\delta_j = 1\%$ except for the 5th parameter ($x_5 = 0.0$). We make $\Delta x_5 = 1\% \cdot t_{si} = 0.01$. Here we let $m = 200$ and get the results as below:

Table 1. The sensitivities of the eight parameters (*low-frequency capacitance* model).

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
0.3304	0.2915	0.0292	1.4264	1.6993	0.0628	8.9043	41.0280

From the Table 1, we can see that the 3rd parameter N_{const} has the lest sensitivity in the eight ones. Then the next one is the 6th parameter σ_2. After them are the 2nd σ_1 and 1st

N_{0poly} parameters. The other ones 4th N_{0si}, 5th x_0, 7th t_{sio_2} and 8th W_f have comparatively large sensitivities.

We can use an inequality to represent their sensitivities:

$$\begin{aligned}\{\text{No.3: } N_{const}\} < \{\text{No.6: } \sigma_2\} < \{\text{No.2: } \sigma_1\} < \{\text{No.1: } N_{0poly}\} \\ < \{\text{No.4: } N_{0si}\} < \{\text{No.5: } x_0\} < \{\text{No.7: } t_{sio_2}\} < \{\text{No.8: } W_f\}\end{aligned} \tag{4.5}$$

In next section, we may find (4.5) benefits the multi-scale method very much.

Remark 1. Before we finish this section, let us introduce another MOS capacitor model, *deep depletion* model. It offers us much information in our research as well as the *low-frequency capacitance* model.

Its difference with the *low-frequency capacitance* model lies only the charge density $\rho(\phi(x), x)$ in Si layer. Here

$$\rho(\phi(x), x) = \mathrm{q}\left[\mathrm{n_i} \exp\left(\frac{-\mathrm{q}\phi(x)}{\mathrm{kT}}\right) - N_{si}(x)\right] \qquad \text{in } Si \text{ layer,} \tag{4.6}$$

where the second term on the right-hand side in (2.3) disappear.

Since the process of sensitivity analysis for *deep depletion* model is same, we directly give out the results. Figures 14-21 show outwardly the sensitivities of the parameters. Table 2 shows the values of sensitivities.

Table 2. The sensitivities of the eight parameters (*deep depletion* model).

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
0.1122	0.1001	0.0261	1.2353	2.9130	0.1605	8.2958	31.6144

and inequality (4.7) represents the relationship of their sensitivities,

$$\begin{aligned}\{\text{No.3: } N_{const}\} < \{\text{No.2: } \sigma_1\} < \{\text{No.1: } N_{0poly}\} < \{\text{No.6: } \sigma_2\} \\ < \{\text{No.4: } N_{0si}\} < \{\text{No.5: } x_0\} < \{\text{No.7: } t_{sio_2}\} < \{\text{No.8: } W_f\}.\end{aligned} \tag{4.7}$$

Remark 2. In fact, m can be set to other values such as 100, 1000, etc. After several computations, we may see that although these sensitivity values are different for different m used, the relationship of these sensitivities are the same. The inequalities (4.5) and (4.7) are still right and give us the indication for the multi-scale method.

5. The Inverse Problem As an Optimization Problem

The basic idea of this approach for solving the inverse problem of parameters extraction is by curve fitting the equations in the forward model to a set of measured C-V data using nonlinear least- squares optimization methods. Starting from the educated guess values for these parameters, a complete set of optimum parameters are extracted using numerical methods to minimize the error between the model and the measured data.

5.1. The Objective Function

Here we again simplify the forward model:

$$\mathbf{C} = \mathbf{C}(\mathbf{X}, \mathbf{V}), \tag{5.1}$$

where $\mathbf{X} \in \mathbb{R}^n$ is the vector of parameters, n is the total number of parameters and $\mathbf{C}, \mathbf{V} \in \mathbb{R}^m$, m is the number of data points that are to be fitted.

We define function $F(\mathbf{X})$ as the simulation error that is caused by the parameters $\mathbf{X}$. $F(\mathbf{X})$ is called *objective function*. It denotes the extent how the computed or simulated parameters' results approach the experimentally measured or desired data. It is assured that $F(\mathbf{X})$ is a real-valued function and is at least once continuously differentiable with respect to the parameters $\mathbf{X}$, although not necessarily in a strict mathematical sense. When $F(\mathbf{X}^*)$ is the minimum of

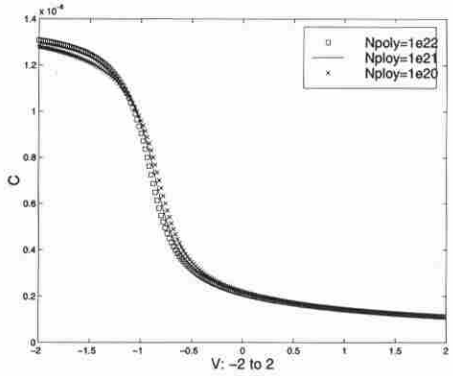

Figure 14: N_{0poly} = 1e20, 1e21, 5e21, 1e22 cm^{-3}.

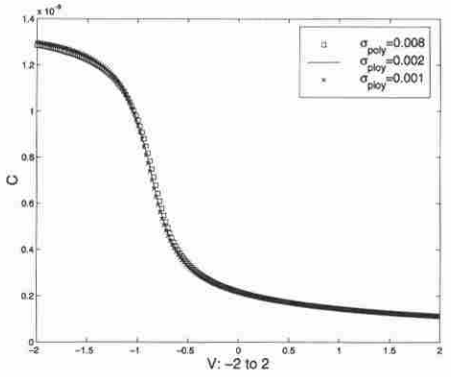

Figure 15: σ_1 = 0.001, 0.002, 0.004, 0.008 μm.

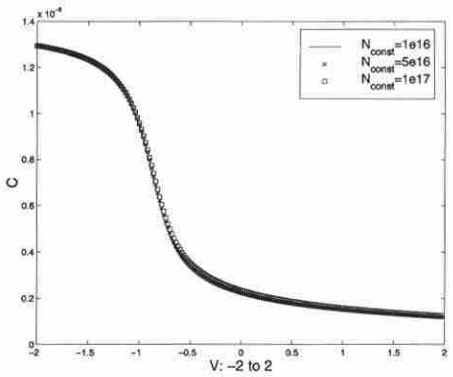

Figure 16: N_{const} = 1e16, 5e16, 1e17 cm^{-3}.

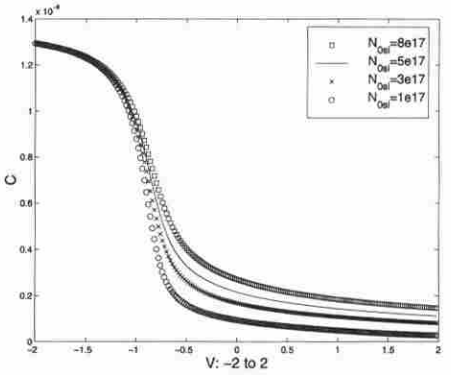

Figure 17: N_{0si} = 1e17, 3e17, 5e17, 8e17 cm^{-3}.

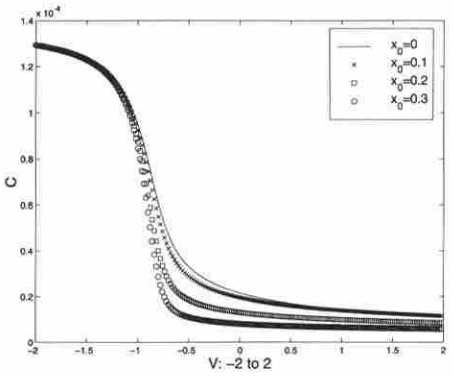

Figure 18: x_0 = 0.0, 0.1, 0.2, 0.3 μm.

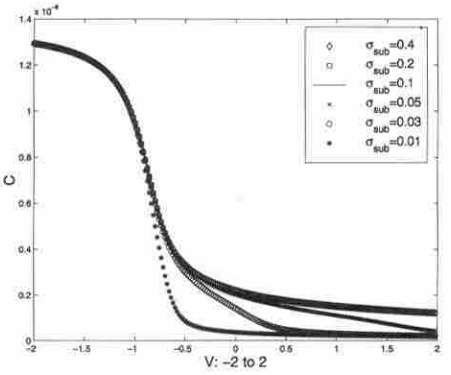

Figure 19: σ_2 = 0.01, 0.03, 0.05, 0.1, 0.2, 0.4 μm.

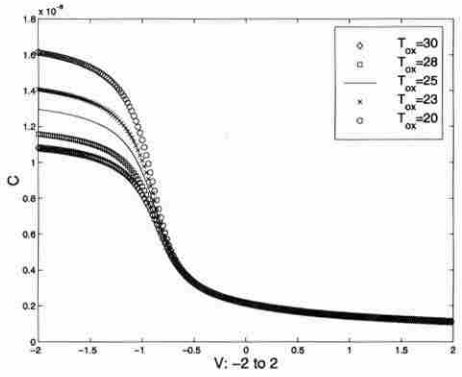

Figure 20: $t_{sio_2} = 20,\ 23,\ 25,\ 28,\ 30$ Å.

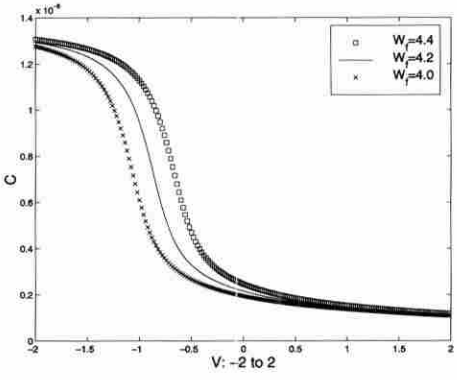

Figure 21: $W_f = 4.0,\ 4.2,\ 4.4$ V.

$F(\mathbf{X})$, $\mathbf{X}^*$ is the optimal parameter values. Then the inverse problem of parameters extraction is reduced to finding $\mathbf{X}$ such that $F(\mathbf{X})$ is minimized.

So choosing an appropriate objective function is very pivotal step. In most practical problems, *the least-square function* is a good one:

$$F(\mathbf{X}) = \sum_{i=1}^{m} \left[\frac{C_i - C(\mathbf{X}; V_i)}{\omega_i} \right]^2, \tag{5.2}$$

in which C_i is experimentally measured capacitance value , $C(\mathbf{X}; V_i)$ is the simulated value and ω_i is *weight factor* which denotes more weight to the specific data points in a certain region of the device characteristics than to others, so that the model is forced to fit adequately the data in those regions. Here, we take the simplest case $\omega_i = 1$, then each data point is equally weighted. In general,

$$m(\text{number of data points } (C_i, V_i)) > n(\text{number of model parameters})$$

In our problem, we use an expression with the *relative error* sense

$$F(\mathbf{X}) = \sum_{i=1}^{m} \left[\frac{C_i - C(\mathbf{X}; V_i)}{C_i} \right]^2 \tag{5.3}$$

instead of (5.2). In most softwares for parameters extraction [7, 12, 13, 14, 15], the equation (5.3) is used. Once we gain the minimum of the objective function, the error of the modeling can be obtained by the following expression:

$$Error = \sqrt{\frac{F(\mathbf{X})}{m}}, \tag{5.4}$$

which would be a good criterion for quantitatively evaluating agreement between the model equations and measured characteristics.

5.2. The Optimization Methods and the Multi-Scale Problem

For the nonlinear least-squares problem, there are many researchers who have studied and developed various algorithms such as the steepest descent method, the Guass-Newton method and the Levenberg-Marquardt method. Here we do not discuss the details about these methods. The readers can find them in any books or papers on optimization methods [2, 8, 9, 11].

In this paper we mainly show some optimization strategies, not only optimization methods. After some computation, we find it difficult to extract these parameters simultaneously using the Levenberg-Marquardt method solely because the physical scales of these parameters are

various, which make their sensitivities widely divergent. Some parameters' sensitivities are very weak, even on the whole device operating range of voltage. Other sensitivities are strong on part of the range, while weak on other part. So it is more practical to extract the parameters by coupling the optimization technique with the analysis of the parameters' sensitivities. In the next subsection, we will give the detailed strategy based on the Levenberg-Marquardt method.

5.3. The Multi-Scale Method for the Inverse Problem

We first consider the case in *low-frequency capacitance* model. The case in *deep depletion* model is similar and shown later. According to the parameters' sensitivities in Table 1 and (4.5), we divide the eight parameters into four groups on different levels. The first group is No.7 and No.8 parameters that have the largest two sensitivities among the eight ones. Then the second group is No.4 and No.5 parameters, the sensitivities of which are of order 10^0. The following group is No.1 and No.2 parameters with sensitivities' of order 10^{-1}. The last group is No.6 and No.3 parameters, the sensitivities of which are the smallest with their order 10^{-2}. We use the levels to arrange the sequences of the parameters' optimization.

After these preparation, we proceed the optimization strategy in following steps:

Step 1: Invert No.7 and No.8 parameters together.

Because No.7 and No.8 parameters have the largest two sensitivities, we invert them firstly. And in Figures 12 and 13, C change on the majority of interval $[-2, 2]$. We use all data points on the whole interval $[-2, 2]$.

Step 2: Invert No.4, 5, 7, 8 parameters together.

In this step, we begin to invert No.4 and No.5 parameters that belong to the second group. In Figures 9 and 10, the shifts of C arise on the majority of interval $[-1, 1]$. We use all data points on the interval $[-1, 1]$. But please note the values of No.7 and No.8 parameters are not frozen, we still invert them with No.4 and No.5 together at the same time.

Step 3: Invert No.1, 2, 4, 5, 7, 8 parameters together.

In this step, we add No.1 and No.2 parameters with smaller sensitivities to the group No.4, No.5, No.7 and No.8. We invert them on the interval $[0, 2]$ because for No.1 and No.2 parameters, C change mainly on the interval $[0, 2]$ as Figure 6 and 7 show.

Step 4: Invert No.1, 2, 4, 5, 6, 7, 8 parameters together.

Still according to our analysis, we begin to invert No.6 parameter with the parameters in Step 3. We invert them on the interval $[-1, 1]$ because for No.6 parameter, the shift of C arises on the interval $[-1, 1]$ (see Figure 11).

Step 5: Invert all eight parameters together.

After Step 4, we begin to invert No.3 parameter, whose sensitivity is the smallest. We invert all the eight parameters on the whole interval $[-2, 2]$.

Steps Continue: Return back to Step 1.

After a cycle of these five steps, we return back to Step 1 and continue the iterations. We hope that the big cycles can lead the parameters' values to the standard values at last.

A flow diagram for the strategy is shown in Figure 22.

Remark 3. In the procedure, you may note that we control the stop of every step by a *fix number* besides the relative error. The number comes from the Levenberg-Marquardt method. In fact, there are several interior cycles of iterations in every step. In Figure 22, for example in step 2, 40×2 means there are two cycles, each of which contains 40 iterations. Usually in well-posed problem, the iteration can converge soon. On the other hand, at the beginning of iteration, the effect of convergence is very obvious. But sometimes in ill-posed problem or in the later of the procedure, the convergence is very slow and much more iterations are needed. So the number can be used to restrict the procedures constrainedly. In addition, we use *cycle number* to limit the total times of the big cycle including the five steps. In our problem, we find these chosen numbers are very effective after many numerical experiments.

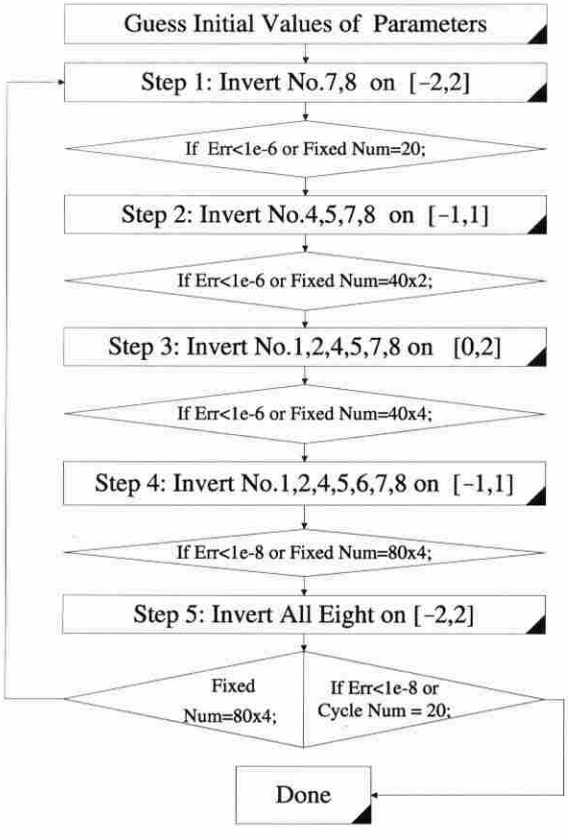

Figure 22: Optimization Flow Diagram.

For the *deep depletion* model, the method is similar except for several details. One is the warranty to classify the parameters, which is different from that for the *low-frequency capacitance* model. From Figure 14-21 we note that the C-V curve of the *deep depletion* model is very distinct. When the gate voltage V_g is increased above threshold, the C-V curve does not rise anymore. About the physical background of the model, please see related books on semiconductor devices (Chapter 4 in [1], Chapter 4 in [3]). We classify the parameters mainly according to both different intervals where C of these parameters change and their sensitivities.

We still choose No.7 and No.8 parameters that have the largest two sensitivities as the the first group with the interval $[-2,-1]$ on which the main shifts of their C-V curves arise (see Figure 20 and 21). Then the second group is No.4, No.5 and No.6 parameters with the interval $[-1,2]$ (see Figures 17, 18 and 19). The following group is No.1 and No.2 parameters with the interval $[-2,-1]$ (see Figures 14 and 15). The last group is No.3 parameter, the sensitivity of which is the smallest (Figures 16).

We briefly proceed the strategy for *deep depletion* model in following steps:

Step 1: Invert No.7 and No.8 parameters together.
We use all data points on the interval $[-2,-1]$.

Step 2: Invert No.4, 5, 6, 7, 8 parameters together.
We use all data points on the interval $[-1,2]$.

Step 3: Invert No.1, 4, 5, 6, 7, 8 parameters together.

In this step, we invert them on the whole interval $[-2, 2]$ because for No.1, No.7 and No.8 parameters, the interval is $[-2, -1]$, while the interval is $[-1, 2]$ for No.4, No.5 and No.6.

Step 4: Invert No.1, 2, 7, 8 parameters together.
We invert them on the interval $[-2, -1]$.

Step 5: Invert all eight parameters together.
We invert them on the whole interval $[-2, 2]$.

Steps Continue: Return back to Step 1.

6. Results

6.1. The Results of *Low-Frequency Capacitance* Model

We set the number of data points $m = 40$. To show the inversion results clearly, we list all data in the form of the ratios of the parameters' initial guess values or their inversion values to their standard values so that the data are centered at 1.0 (except for the No.5 parameter whose standard value is 0.0, we let its initial guess value or its inversion value divided by the thickness of Si layer t_{si}). Table 3 shows the results with several kinds of initial deviations:

Table 3. The results of *low-frequency capacitance* model.

	Parameter	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8
	Standard value	1.0	0.002	0.1	5.0	0.0	0.1	0.0025	4.2
Ex. 1 +10% deviation	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
	Initial ratio	1.1	1.1	1.1	1.1	0.1	1.1	1.1	1.1
	Inversion ratio	1.0006	1.0007	0.9755	1.0005	0.0000	1.0002	1.0000	1.0000
Ex. 2 ±10% deviation	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
	Initial ratio	0.9	1.1	0.9	1.1	0.1	1.1	0.9	0.9
	Inversion ratio	1.0005	1.0006	0.6772	1.0064	0.0000	1.0034	1.0000	1.0000
Ex. 3 +20% deviation	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
	Initial ratio	1.2	1.2	1.2	1.2	0.2	1.2	1.2	1.2
	Inversion ratio	1.0022	1.0025	0.4959	1.0010	0.0000	1.0052	1.0000	1.0000
Ex. 4 +30% deviation	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
	Initial ratio	1.3	1.3	1.3	1.3	0.2	1.3	1.2	1.2
	Inversion ratio	0.9998	0.9998	0.2624	1.0147	0.0000	1.0077	1.0000	1.0000

Example 1. We let all initial deviations be +10%. The inversion result matches the standard values very well with the last deviations smaller than 10^{-3}, except for No.3 parameter.

Example 2. We take some initial deviations as +10% and some as −10%. The result still matches the standard values well with some last deviations smaller than 10^{-3} (No.1, 2, 5, 7 and 8) some smaller than 10^{-2} (No.4 and 6) , except for No.3 parameter. In fact, we have many other results with various combinations of upper and lower deviations. They all match the standard values well except for No.3 parameter. Limited to space, we do not list them here. These results show the generality of the optimization strategy.

Example 3. Here we increase all initial deviations to +20%. The result is still very satisfying. The last deviations are smaller than 10^{-2} except for No.3 parameter.

Example 4. Here we increase some deviations to 30% (No.1, 2, 3, 4 and 6). The result keeps good except for No.3 parameter.

6.2. The Results of *Deep Depletion* Model

Table 4 shows the results with several kinds of initial deviations for *deep depletion* model:

Table 4. The results of *deep depletion* model.

	Parameter	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8
	Standard value	1.0	0.002	0.1	5.0	0.0	0.1	0.0025	4.2
Ex. 5	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
+10%	Initial ratio	1.1	1.1	1.1	1.1	0.1	1.1	1.1	1.1
deviation	Inversion ratio	1.0134	1.0139	1.0997	0.9980	0.0000	0.9989	1.0000	1.0000
Ex. 6	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
±10%	Initial ratio	0.9	1.1	0.9	1.1	0.1	1.1	0.9	0.9
deviation	Inversion ratio	0.9876	0.9860	1.0996	0.9965	0.0000	0.9980	1.0000	1.0000
Ex. 7	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
±10%	Initial ratio	1.1	0.9	1.1	0.9	0.1	1.1	1.1	0.9
deviation	Inversion ratio	0.9879	0.9865	1.0994	0.9980	0.0000	0.9988	1.0000	1.0000
Ex. 8	Standard ratio	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
±20%	Initial ratio	1.2	1.2	1.1	1.2	0.2	1.2	0.8	1.2
deviation	Inversion ratio	1.0120	1.0135	1.0999	0.9980	0.0000	0.9989	1.0000	1.0000

Example 5. We let all initial deviations be +10%. The result is good. The last deviations are smaller than 10^{-2} except for No.3 parameter.

Example 6 and 7. Here we give two results with various combinations of upper and lower deviations. The two results are still satisfying, which again show the generality of the optimization strategy except for No.3 parameter.

Example 8. Here the initial deviations reach 20% except for No.3 parameter. The result keeps good except for this parameter.

Remark 4. From Table 3 and 4, we may note that for No.1 and No.2 parameters, the results of *deep depletion* model are not good as those of *low-frequency capacitance* model. The last deviations of *deep depletion* model are larger than 10^{-2} and those of *low-frequency capacitance* model are smaller than 10^{-2}. We think the sensitivity difference is the cause. In Figure 14 and 15 the C-V curves of *deep depletion* change slightly on the whole interval $[-2, 2]$, even like it changes in Figure 16 of No.3 parameter. But in Figure 6 and 7, the C-V curves of *low-frequency capacitance* model change at least on $[0, 2]$. Numerically, according to Table 1 and 2, the sensitivities for No.1 and No.2 parameters of *low-frequency capacitance* model are 0.3304 and 0.2915, while those of *deep depletion* are 0.1122 and 0.1001, which produce the difference in last deviations. This phenomena again indicates the importance of sensitivity analysis.

Remark 5. Although No.3 parameter is still a difficulty in our problem because of its extraordinarily weak sensitivity, we must emphasize the effectiveness of the optimization strategy. Before applying the strategy, we can extract only **five** parameters (No.1, 2, 4, 5 and 7) simultaneously using the Levenberg-Marquardt method solely for *low-frequency capacitance* model. Now we can extract **seven** parameters for both two models.

7. Conclusions and Future Work

In this paper the multi-scale methods for the inverse modeling in 1-D MOS capacitor are described. Numerical experiments show the efficiency of the methods for both *low-frequency capacitance* and *deep depletion* models. In fact, these methods are some practical optimization strategies based on our sensitivity analysis of the parameters and those popular optimization algorithms. However, the multi-scale physical difference of parameters widely exists in all kinds of practical problems and is an unavoidable obstacle in solving forward and inverse problems.

It has become a key point in parameters extraction not only from semiconductor industry, but also from others, e.g. petroleum exploitation industry, seismic tomography in geology, large-scale weather forecasting, etc. So we hope our study can give some illumination or help to the researchers or engineers in these aspects.

Current and future work with these methods includes application into a more complicated MOS model with quantum transport, the extension to 2-D and 3-D modeling problems and improving the initial guess with which to begin the iterative process. In the model with quantum transport, the effect of multi-scale physical problem will be more serious because not only the physical difference of parameters is great, but also the equations for the forward problem lie on very different physical levels i.e. Possion's equation on classical macrostructure and Schrödinger equation on quantum microstructure. In addition, in practical problem the values of the parameters are unknown and to be extracted, we can not make initial guess of them by some deviations from standard values like in this paper, so getting a good initial guess is necessary and helpful.

References

[1] N. Arora, *MOSFET Models for VLSI Circuit Simulation, Theory and Practice* (Springer–Verlag, Wien 1993).

[2] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ 1983).

[3] B. El-Kareh and R. J. Bombard, *Introduction to VLSI Silicon Devoices, Physics, Technology and Characterization* (Kluwer Academic Publishers, Boston 1986).

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations* Third edition, (The Johns Hopkins University Press, Baltimore and London 1998).

[5] R. Kress, *Numerical Analysis* (Springer–Verlag, New York 1998).

[6] R. Li, T. Tang and P. W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, *J. Comput. Phys.* **170**(2001), 562-588.

[7] W. Maes, K. M. De Meyer, and L. H. Dupas, SIMPAR: A veratile technology independent parameter extraction program using new optimized fit strategy, *IEEE Trans. Computer-Aided Design*, CAD-5, 320-325 (1986).

[8] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.* **11**(1963), 431-441.

[9] J. More, The Levenberg-Marquardt Algorithm: Implementation and Theory, in: *Numerical Analysis, Lecture Notes in Mathematics*, G. A. Watson (ed.) vol. 630, 105-116, (Springer–Verlag, Berlin 1977).

[10] E. H. Nicollian and J. R. Brews, *MOS (Metal Oxide Semiconductor) Physics and Technology* (Wiley, New York 1982).

[11] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge University Press, Cambridge 1990).

[12] M. Sharma and N. Arora, OPTIMA: A nonlinear model parameter extraction program with statistical confidence region algorithms, *IEEE Trans. Computer-Aided Design*, CAD-12, (May 1993).

[13] S. J. Wang, J. Y. Lee, and C. Y. Chang, An efficient and reliable approach for semiconductor device parameter extraction, *IEEE Trans. Computer-Aided Design*, CAD-6, 170-178 (1986).

[14] D. E. Ward and K. Doganis, Optimized extraction of MOS model parameters, *IEEE Trans. Computer-Aided Design*, CAD-1, 163-168 (1982).

[15] P. Yang and P. K. Chatterjee, An optimal parameter extraction program for MOSFET models, *IEEE Trans. Electron Devices*, ED-30, 1214-1219 (1983).