A Discretization of the semiconductor equations

Typically, the finite volume method (or the box method) (Laux & Grossman, 1985) is applied to convert the differential operator into the algebraic form. Then, the equation is converted into a collection of algebraic equations, which are evaluated at many sampling points (called vertex nodes). Since the original set is nonlinear with respect to n, p, and ϕ, the converted algebraic equations are also nonlinear. The Newton-Raphson method is applied to the resultant set of equations. In this section, the discretization of the semiconductor equations, (1) and (3), is sketched.

It is observed that both equations have a common form of

$$\nabla \cdot \mathbf{F} - s = 0, \tag{8}$$

where $\mathbf{F}$ is the flux term and s is the source term. In order to get a discretized form of the above equation at a certain grid point, we integrate the equation over a finite volume (or a box) surrounding the point. For example, in a grid shown in Figure 8, the discretized equation for the center point can be obtained by the integration over the red pentagon. With help of the divergence theorem (Jackson, 1999), we have

$$\oint_{Surface} \mathbf{F} \cdot d\mathbf{a} - \int_{Box} s d^3x = 0. \tag{9}$$

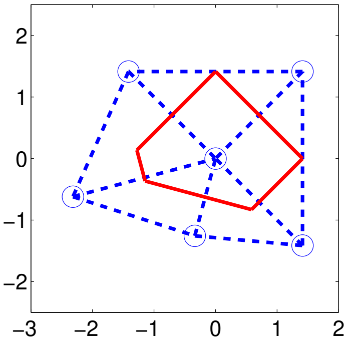


Figure 8: Grid points (Blue circles) and a finite volume surrounding the center point (A red pentagon).

Every node except for the terminal nodes contributes the residual vector following (9). We must also consider the boundary conditions at the terminal nodes. For the electron density (n) and the hole density (p), we have

$$n - n_{eq} = 0, \tag{10}$$

$$p - p_{eq} = 0, \tag{11}$$

where n_{eq} and p_{eq} are the carrier densities at the equilibrium condition. These densities do not depend on the applied terminal voltages. However, the boundary condition for the electrostatic potential (ϕ) reads

$$\phi - \phi_{eq} - V_{terminal} = 0, \tag{12}$$

where ϕ_{eq} is the equilibrium potential and $V_{terminal}$ is the applied terminal voltage. When the applied terminal voltage changes, s_i's in (5) at the terminal nodes change accordingly.

B Bias ramping

The set of discretized equations in (5) is highly nonlinear and the iterative Newton-Raphson method is used to solve the system. Although a nonlinear relaxation scheme to avoid the full Newton-Raphson method (Meinerzhagen et al., 1991) has been proposed, the full Newton-Raphson method is still preferred due to its robustness.

Let us consider a typical case, where a user want to know the drain current at $V_G = V_D = V_{DD}$. V_{DD} is the maximum voltage which can be applied to the MOSFET. For example, V_{DD} is 1.1 V in our example. The drain current at this bias condition is called as the ON current and the ON current represents the current-driving performance of the MOSFET. Since the ON current is of interest, it would be most desirable to calculate the ON current immediately. However, an appropriate initial guess to start the Newton-Raphson method at $V_G = V_D = V_{DD}$ is not available.

When $V_G = V_D = 0$ V, the device is in its equilibrium condition. In this special case, an effective way to generate the initial guess is well known (Jungemann & Meinerzhagen, 2003). It is based upon the charge neutrality condition, where $p - n + N_{dop}^{+} = 0$ is imposed locally. Unfortunately, this method cannot be applied to non-zero terminal voltages. No established method for directly solving the equation at high terminal voltages exists. In order to overcome such a difficulty, the bias ramping technique is typically adopted. Starting from the equilibrium operating condition with all zero terminal voltages, many intermediate steps toward the target terminal voltages are introduced. Based upon the converged solution at the previous condition as an initial guess, the next step is solved with the Newton-Raphson method.

In this conventional solution method, the overall simulation time, τ_{conv}, can be written as

$$\tau_{conv} = \sum_{i=1}^{N_{step}} \tau_{conv}^{i} = \sum_{i=1}^{N_{step}} N_{newton}^{i} \tau_{single} = N_{step} \times N_{newton} \times \tau_{single}, \tag{13}$$

where N_{step} is the number of the entire bias conditions, τ_{conv}^{i} is the simulation time for the i-th bias condition, N_{newton}^{i} is the number of the Newton iterations for the i-th bias condition, τ_{single} is the time spent for a single Newton iteration, and N_{newton} is the average number of the Newton iterations. The bias ramping heavily increases the simulation time, because it introduces a large N_{step}.

In this work, we propose to solve the equation set at the target terminal voltages directly without the bias ramping. The overall simulation time with the proposed method based upon the neural network, τ_{nn}, can be written as

$$\tau_{nn} = N_{newton}^{direct} \times \tau_{single}, \tag{14}$$

where N_{newton}^{direct} is the number of the Newton iterations at the target bias condition. Two numbers, N_{newton} and N_{newton}^{direct}, may be comparable. It is noted that N_{step} in (13) does not appear any more in the above equation. The reduction factor of the simulation time, which is defined as τ_{conv}/τ_{nn}, can be approximated with N_{step}, as shown in (6).

C Device simulator

In this work, the device simulator in Han & Hong (2019) has been used to obtain the simulation results. As it is free from the license issue, several simulation runs can be performed simultaneously. The Poisson equation, (3), the electron continuity equation, (1), and the hole continuity equation are coupled and solved self-consistently. A rectangular grid with varying spacing is adopted. Therefore, every finite volume for a grid point has four faces. The device template is implemented to accept the device parameters (such as L_G, t_{ox}, N_{sd}, and N_{sub}) as input parameters. When the Netwon-Raphson method is used, we must solve the following matrix equation:

$$\mathbf{J}\mathbf{u} = -\mathbf{r}, \tag{15}$$

where $\mathbf{J}$ is the Jacobian matrix, $\mathbf{u}$ is the update vector, and $\mathbf{r}$ is the residual vector. It is noted that the Jacobian matrix is very sparse in the semiconductor device simulation. In order to solve the above equation, a sparse matrix solver, the UMFPACK library (Davis, 2004), is used.

D More information on dataset

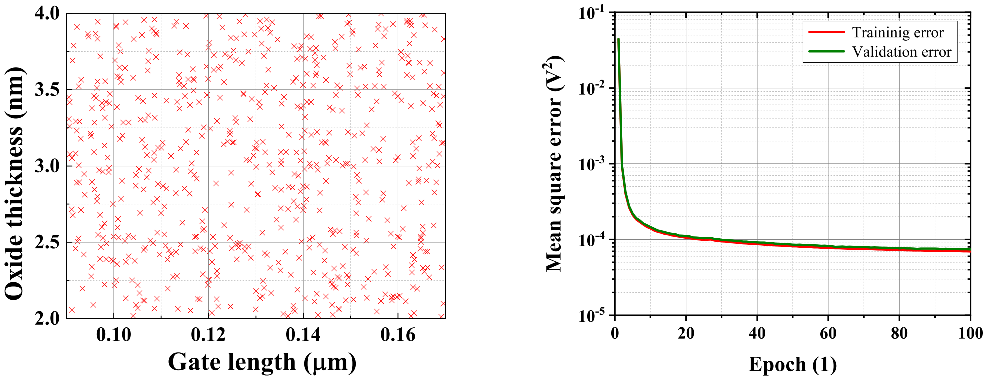
More information on the dataset is provided.

Each data point is specified with parameters of L_G, t_{ox}, N_{sd}, N_{sub}, V_G, and V_D and their ranges are summarized in Table 2.

Table 2: Ranges of input parameters

Parameter	Range
Gate length, L_G	90 nm $\sim$ 170 nm
Oxide thickness, t_{ox}	2 nm $\sim$ 4 nm
Source/drain doping density, N_{sd}	5×10^{19} cm^{-3} $\sim$ 5×10^{20} cm^{-3}
Substrate doping density, N_{sub}	5×10^{17} cm^{-3} $\sim$ 5×10^{18} cm^{-3}
Gate voltage, V_G	0 V $\sim$ 1.1 V
Drain voltage, V_D	0 V $\sim$ 1.1 V

We curate the dataset with 10,112 instances that are randomly selected. as shown in Figure 9a. The training and validation errors are measured as a function of the learning epoch as shown in Figure 9b.



(a)

(b)

Figure 9: (a) Distribution of the selected devices in the L_g-t_{ox} plane. (b) Training and validation errors as functions of the learning epoch.