Computational Microelectronics
Lecture 4 Diffusion

Sung-Min Hong (smhong@gist.ac.kr)
Semiconductor Device Simulation Laboratory
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology

Diffusion

Diffusion equation in 1D

- In general, $\frac{\partial C}{\partial t}$ does not vanish.
  - Therefore, we must solve the following equation:
$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2}$$
  - In the previous lecture, $\frac{\partial^2 C}{\partial x^2}$ was discretized.
$$\frac{\partial^2 C}{\partial x^2} \approx \frac{C(x_{i+1}) - 2C(x_i) + C(x_{i-1})}{(\Delta x)^2}$$
  - The remaining task is to discretize $\frac{\partial C}{\partial t}$.

Euler schemes

- Forward Euler scheme
  - In the forward Euler scheme, the time derivative is calculated by taking the difference between the ***future*** and ***present*** values:

$$\left.\frac{\partial C(x_i)}{\partial t}\right|_{t_k} \approx \frac{C(t_{k+1}) - C(t_k)}{t_{k+1} - t_k}$$

- Backward Euler scheme
  - In the backward Euler scheme, the time derivative is calculated by taking the difference between the ***present*** and ***past*** values:

$$\left.\frac{\partial C(x_i)}{\partial t}\right|_{t_k} \approx \frac{C(t_k) - C(t_{k-1})}{t_k - t_{k-1}}$$

Euler schemes

- Forward Euler scheme
  - In the forward Euler scheme, the time derivative is calculated by taking the difference between the ***future*** and ***present*** values:

$$\left.\frac{\partial C(x_i)}{\partial t}\right|_{t_k} \approx \frac{C(t_{k+1}) - C(t_k)}{t_{k+1} - t_k}$$

- Backward Euler scheme
  - In the backward Euler scheme, the time derivative is calculated by taking the difference between the ***present*** and ***past*** values:

$$\left.\frac{\partial C(x_i)}{\partial t}\right|_{t_k} \approx \frac{C(t_k) - C(t_{k-1})}{t_k - t_{k-1}}$$

In a vector form

- Solution vector, $[C(x_0)\ C(x_1)\ \ldots\ C(x_{N-1})]^T$
  - Then, in the backward Euler scheme, the time derivative can be written as

$$\left.\frac{\partial C}{\partial t}\right|_{t_k} \rightarrow \frac{1}{t_k - t_{k-1}}\begin{bmatrix} C(x_0, t_k) \\ C(x_1, t_k) \\ \vdots \\ C(x_{N-1}, t_k) \end{bmatrix} - \frac{1}{t_k - t_{k-1}}\begin{bmatrix} C(x_0, t_{k-1}) \\ C(x_1, t_{k-1}) \\ \vdots \\ C(x_{N-1}, t_{k-1}) \end{bmatrix}$$

Discretized diffusion equation

- Let us assume the fixed C values at two boundaries.
  - Then, in the backward Euler scheme,

$$\frac{1}{t_k - t_{k-1}}\begin{bmatrix} C(x_0, t_k) \\ C(x_1, t_k) \\ \vdots \\ C(x_{N-1}, t_k) \end{bmatrix} - \frac{1}{t_k - t_{k-1}}\begin{bmatrix} C(x_0, t_{k-1}) \\ C(x_1, t_{k-1}) \\ \vdots \\ C(x_{N-1}, t_{k-1}) \end{bmatrix}$$

$$= \frac{D}{(\Delta x)^2}\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} C(x_0, t_k) \\ C(x_1, t_k) \\ \vdots \\ C(x_{N-1}, t_k) \end{bmatrix}$$

Boundary condition

Boundary condition

Unknown and known variables

- $C(x_i, t_k)$ is unknown, while $C(x_i, t_{k-1})$ is known.
  - After simple manipulation,

$$\frac{1}{t_k - t_{k-1}}\begin{bmatrix} C(x_0, t_k) \\ C(x_1, t_k) \\ \vdots \\ C(x_{N-1}, t_k) \end{bmatrix} - \frac{D}{(\Delta x)^2}\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} C(x_0, t_k) \\ C(x_1, t_k) \\ \vdots \\ C(x_{N-1}, t_k) \end{bmatrix}$$

$$= \frac{1}{t_k - t_{k-1}}\begin{bmatrix} C(x_0, t_{k-1}) \\ C(x_1, t_{k-1}) \\ \vdots \\ C(x_{N-1}, t_{k-1}) \end{bmatrix}$$

It is $Ax = b$, again.

- In this form,

$$A = \frac{1}{t_k - t_{k-1}} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} - \frac{D}{(\Delta x)^2} \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$x = \begin{bmatrix} C(x_0, t_k) \\ C(x_1, t_k) \\ \vdots \\ C(x_{N-1}, t_k) \end{bmatrix} \text{ and } b = \frac{1}{t_k - t_{k-1}} \begin{bmatrix} C(x_0, t_{k-1}) \\ C(x_1, t_{k-1}) \\ \vdots \\ C(x_{N-1}, t_{k-1}) \end{bmatrix}$$

Implementation

- Start from the initial dopant profile.
  - Calculate a new dopint profile by solving the discretized diffusion equation.
  - Repeat this process until the time reaches at its target value.

Example) Boron diffusion

- Initially, a Dirac-delta-like profile with a dose of 2 X 10^{13} cm^{-2}
  - Thermal diffusion for 60 min at 1100 °C
  - Diffusivity follows

$$D = D^0 \exp\left(-\frac{E_A}{k_B T}\right)$$

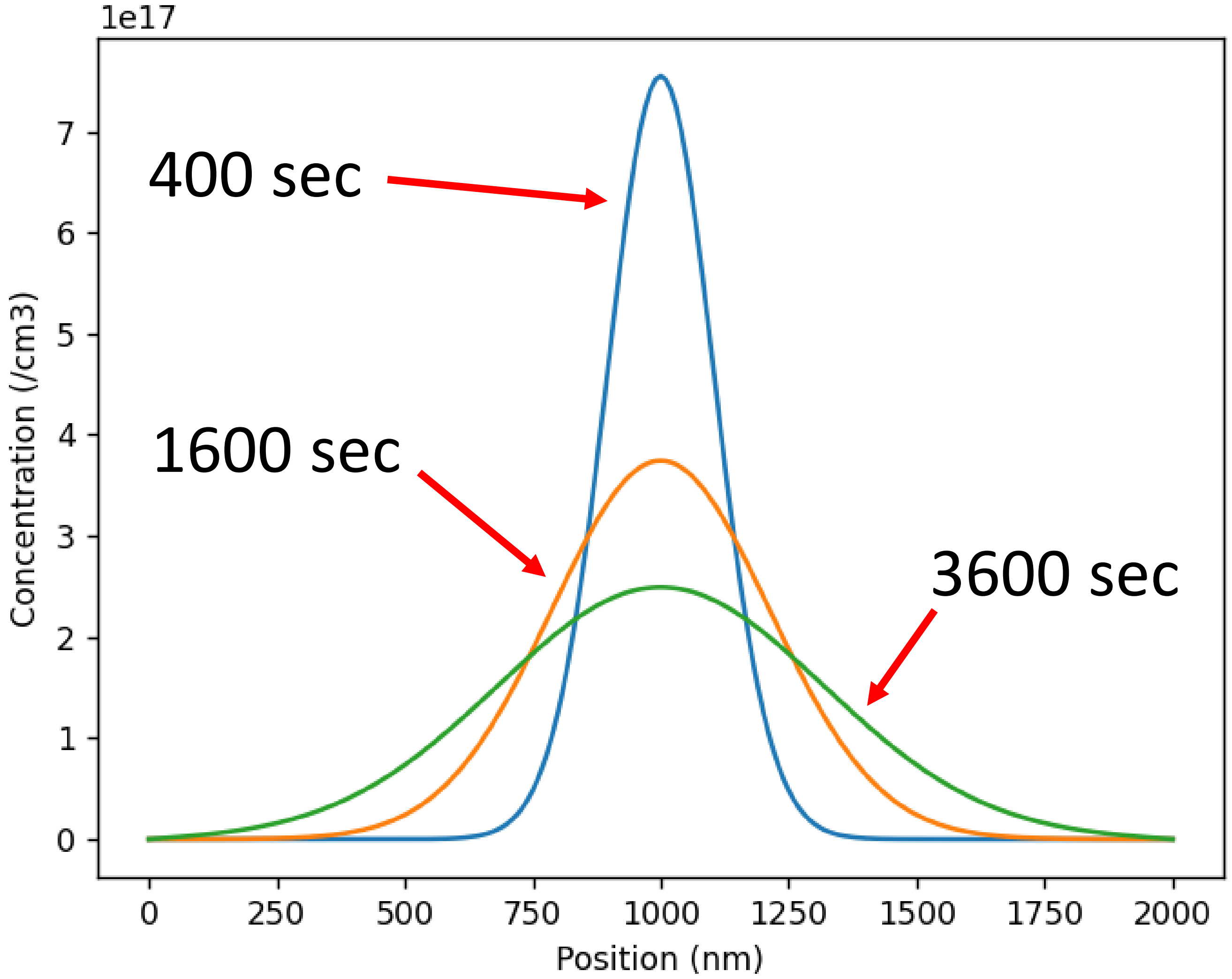
  - For borons, D^0 is 1.0 cm^2 sec^{-1} and E_A is 3.5 eV.
  - Then, at 1100 °C, D is about 1.42 X 10^{-13} cm^2 sec^{-1}.

Simulation parameters

- Real space
  - 1001 mesh points with 10 nm spacing
  - Initially, at 500th point, 2 X 10^{19} cm^{-3}. (Match the dose.) At all other points, the concentration vanishes.
- Time
  - Time evolves from 0 to 3600 sec.

Simulation results

- Dopant profiles at some time instances (400 sec, 1600 sec, and 3600 sec)
  - Gaussian profile
  - Peak ~ $\frac{1}{\sqrt{Dt}}$



HW#4

- Due: AM08:00, September 11
- Problem#1
  - Implement the diffusion solver. Verify your simulation results against the graph shown in this lecture. Also, test the same problem at 1000 °C. (Since the temperature is lower, the diffusion becomes quite weak. Modify your simulation parameters for this lower temperature.)

Thank you!