Numerical Analysis For Materials

Final project #1, #2

Gilwoon Lee

ID: 20120083
Dept: Material Science and Engineering

Final project #1

Darken's uphill diffusion

1. Theory

Problem situation

Fe	Fe
3.8wt% Si - 0.478wt% C	0.441 wt% C (0.001 wt% Si 가정)

$$T = 1323\ K,\ \mathrm{t} = 13\ days$$

1. Theory

1. 농도 변수 $u_k = x_k / \sum_{j \in S} x_j$ (농도와 부피의 곱으로 나타낸 새로운 변수)

2. 자유에너지, 각 파라미터

$$G_m = y_{Fe} y_{Va} {}^{o}G_{Fe:Va} + y_M y_{Va} {}^{o}G_{M:Va} + y_{Fe} y_C {}^{o}G_{Fe:C} + y_M y_C {}^{o}G_{M:C}$$
$$+ RT(y_{Fe} \ln y_{Fe} + y_M \ln y_M) + RT(y_{Va} \ln y_{Va} + y_C \ln y_C)$$
$$+ y_{Fe} y_M y_{Va} L_{Fe,M:Va} + y_{Fe} y_M y_C L_{Fe,M:C}$$
$$+ y_{Fe} y_C y_{Va} L_{Fe:C,Va} + y_M y_C y_{Va} L_{M:C,Va}$$

$y_{Fe} = 1 - y_{Si}$, $y_{Va} = 1 - y_C$

$${}^{o}G_{Fe:Va} = {}^{o}G_{Fe}^{fcc}$$
$${}^{o}G_{Si:Va} = {}^{o}G_{Si}^{Diamond} + 51000 - 21.8 \cdot T$$
$${}^{o}G_{Fe:C} = {}^{o}G_{Fe}^{fcc} + {}^{o}G_{C}^{graphite} + 77207 - 15.877 \cdot T$$
$${}^{o}G_{Si:C} = {}^{o}G_{Si}^{Diamond} + {}^{o}G_{C}^{graphite} - 20510 + 38.7 \cdot T$$

=0

$$L_{Fe,Si:Va} = -125248 + 41.116 \cdot T - 142708(y_{Fe} - y_{Si}) + 89907(y_{Fe} - y_{Si})^2$$
$$L_{Fe,Si:C} = +143219.9 + 39.31 \cdot T - 216320.5(y_{Fe} - y_{Si})$$
$$L_{Fe:C,Va} = -34671$$

1. Theory

3. 화학포텐셜

For substitutional M,

$$\mu_M = G_m + (1-y_M)\left(\frac{\partial G_m}{\partial y_M} - \frac{\partial G_m}{\partial y_{Fe}}\right) = G_m + (1-y_M)\frac{dG_m}{dy_M}$$

For interstitial C

$$\mu_C = \left(\frac{\partial G_m}{\partial y_C} - \frac{\partial G_m}{\partial y_{Va}}\right) = \frac{dG_m}{dy_C}$$

$y_{Fe} = 1 - y_{Si},\ y_{Va} = 1 - y_C$

$$\begin{aligned} G_m &= y_{Fe}y_{Va}{}^{o}G_{Fe:Va} + y_M y_{Va}{}^{o}G_{M:Va} + y_{Fe}y_C{}^{o}G_{Fe:C} + y_M y_C{}^{o}G_{M:C} \\ &+ RT(y_{Fe}\ln y_{Fe} + y_M \ln y_M) + RT(y_{Va}\ln y_{Va} + y_C \ln y_C) \\ &+ y_{Fe}y_M y_{Va}L_{Fe,M:Va} + y_{Fe}y_M y_C L_{Fe,M:C} \\ &+ y_{Fe}y_C y_{Va}L_{Fe:C,Va} + y_M y_C y_{Va}L_{M:C,Va} \end{aligned}$$

4. 확산 계수, FDM

$$\frac{\partial C_C}{\partial t} = \frac{\partial}{\partial x}\left[D_{CC}\frac{\partial C_C}{\partial x} + D_{CSi}\frac{\partial C_{Si}}{\partial x}\right]$$

$$\frac{\partial C_{Si}}{\partial t} = \frac{\partial}{\partial x}\left[D_{SiC}\frac{\partial C_C}{\partial x} + D_{SiSi}\frac{\partial C_{Si}}{\partial x}\right]$$

$$\frac{\partial C_i}{\partial t} = \frac{C_i^{j+1} - C_i^j}{\Delta t}$$

$$\frac{\partial}{\partial x}\left[D_i\frac{\partial C_i}{\partial x}\right] = \frac{1}{\Delta x}\left[\sqrt{D_{i+1}D_i}\frac{C_{i+1}^j - C_i^j}{\Delta x} - \sqrt{D_i D_{i-1}}\frac{C_i^j - C_{i-1}^j}{\Delta x}\right]$$

1. Theory

3. Finite Difference Methods for Parabolic Problems

- Heat or diffusion equation

$$\frac{\partial u}{\partial t}(x,t) = \alpha \frac{\partial^2 u}{\partial x^2}(x,t) \qquad \text{for} \quad 0 < x < l \quad \text{and} \quad t > 0$$

subject to the condition

$$u(0,t) = u(l,t) = 0 \qquad \text{for} \quad t > 0$$
$$u(x,0) = f(x) \qquad \text{for} \quad 0 \le x \le l$$

set $x_i = i\Delta x$ for i = 0, 1, 2, ⋯, m

$t_j = j\Delta t$ for j = 0, 1, 2, ⋯

$$\frac{\partial u}{\partial t}(x_i,t_j) = \frac{u(x_i,t_j+\Delta t)-u(x_i,t_j)}{\Delta t} - \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}(x_i,\mu_j)$$

$$\frac{\partial^2 u}{\partial x^2}(x_i,t_j) = \frac{u(x_i+\Delta x,t_j)-2u(x_i,t_j)+u(x_i-\Delta x,t_j)}{\Delta x^2} - \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i,t_j)$$

$$\frac{w_{i,j+1}-w_{i,j}}{\Delta t} = \alpha \frac{w_{i+1,j}-2w_{i,j}+w_{i-1,j}}{\Delta x^2}$$

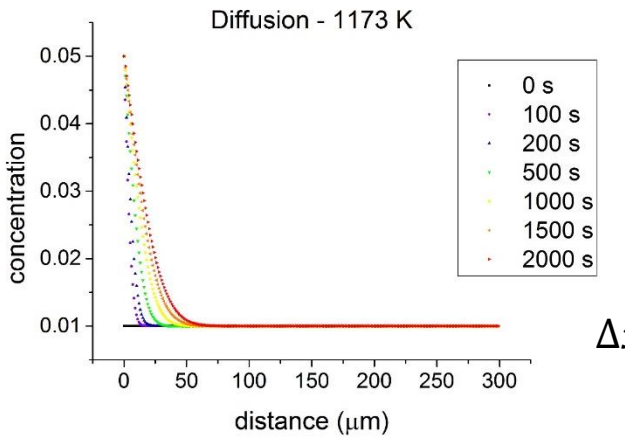
$$w_{i,j+1} = w_{i,j} + \frac{\alpha\Delta t}{\Delta x^2}(w_{i+1,j}-2w_{i,j}+w_{i-1,j}) = \left(1-2\frac{\alpha\ \Delta t}{\Delta x^2}\right) w_{i,j} + \frac{\alpha\ \Delta t}{\Delta x^2}(w_{i+1,j}+w_{i-1,j})$$

$$\frac{\partial u}{\partial t}(x,t) = D\frac{\partial^2 u(x,t)}{\partial x^2}$$

∴ Stability condition of explicit method: $\frac{\alpha\Delta t}{\Delta x^2} \le \frac{1}{2}$

$$\therefore \lambda = \frac{D\Delta t}{(\Delta x)^2}$$ If $\lambda > \frac{1}{2}$, reduce Δt until $\lambda \le \frac{1}{2}$ can be satisfied

(Homework #9)



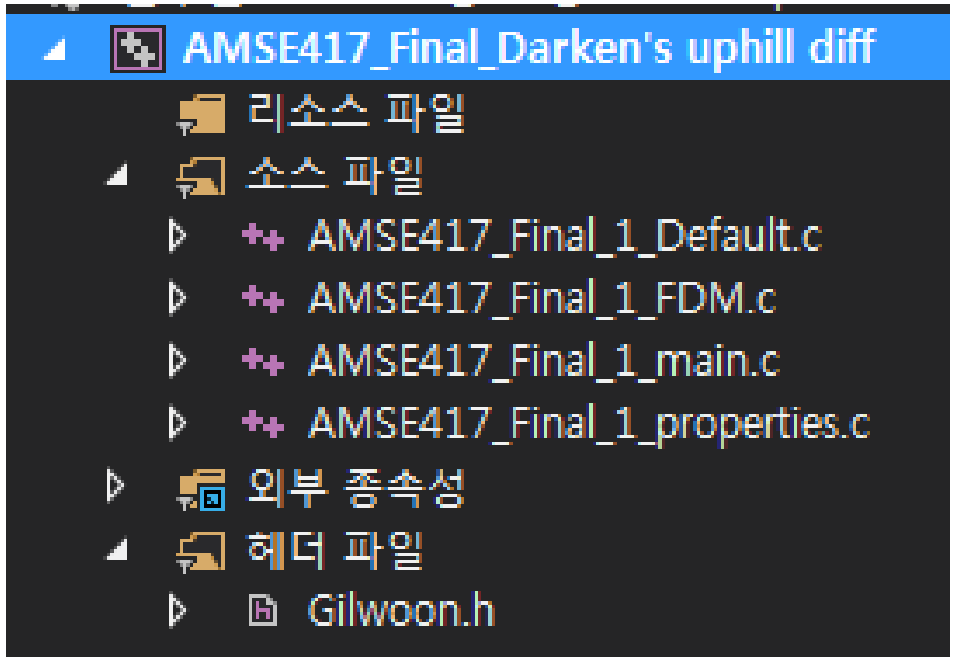
$\Delta x = 1\ \mu m,\ \Delta t = 1\ sec$

$\lambda = 0.11964802$

2. Programmed code

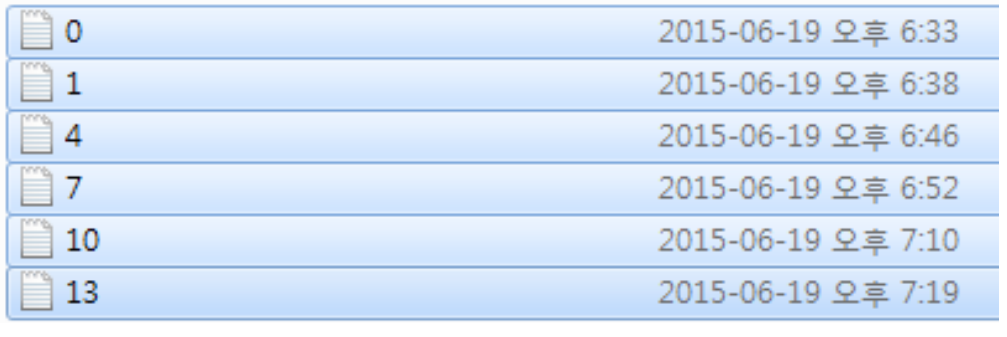
* Description of Program

1. Info. of Program



2. Characteristics of Program

- 헤더 및 각 기능 함수화
- 진행상황 볼 수 있음
- Increment 수정 용이
  (**setting** dt=1 sec, dx=0.001 m)
- 각 온도를 txt 파일명으로 하여 data 출력
- 출력 파일은 0, 1, 4, 7, 10, 13 day일 때 생성
- 컴퓨터가 느리면 오래 걸리는 특징



46분 소요!!

(실행 후 버거킹 다녀와도 끝나지 않는 프로그램!!)

2. Programmed code

* Description of Program

3. FDM()

```
void FDM()
{
FILE *file;
char filename[100];
double C[100];
double C_temp[100];
double Si[100];
double Si_temp[100];
double dx = 0.001;
double dt = 1;
int t = 0;
int i, j, k, l;
double m_Fe = 55.8450;
double m_Si = 28.0855;
double m_C = 12.0107;
double x_Fe, x_Si, x_C;
double wtC, wtSi;

//초기 농도 입력
x_Fe = (95.722 / m_Fe) / ((95.722 / m_Fe) + (3.8 / m_Si) + (0.478 / m_C));
x_Si = (3.8 / m_Si) / ((95.722 / m_Fe) + (3.8 / m_Si) + (0.478 / m_C));
x_C = (0.478 / m_C) / ((95.722 / m_Fe) + (3.8 / m_Si) + (0.478 / m_C));
for (k = 0; k < 50; k++) { C[k] = x_C/(x_Fe+x_Si); C_temp[k] = 0; Si[k] = x_Si/(x_Fe+x_Si); Si_temp[k] = 0; }
x_Fe = (99.558 / m_Fe) / ((99.558 / m_Fe) + (0.001 / m_Si) + (0.441 / m_C));
x_Si = (0.001 / m_Si) / ((99.558 / m_Fe) + (0.001 / m_Si) + (0.441 / m_C));
x_C = (0.441 / m_C) / ((99.558 / m_Fe) + (0.001 / m_Si) + (0.441 / m_C));
for (k = 50; k < 100; k++) { C[k] = x_C / (x_Fe + x_Si); C_temp[k] = 0; Si[k] = x_Si / (x_Fe + x_Si); Si_temp[k] = 0; }
```

2. Programmed code

* Description of Program

```
//W 값 할당
j = 0;
_itoa(t, filename, 10);
strcat(filename, ".txt");
file = fopen(filename, "w");
fprintf(file, "%d\n", t);
for (k = 0; k < 100; k++)
{
C_temp[k] = C[k] / (1 + C[k]);
Si_temp[k] = Si[k] * (1 - C_temp[k]);
wtC = ((C_temp[k] / m_Fe) / ((1 / m_C) - ((1 / m_C) - (1 / m_Fe))*C_temp[k])) * 100;
wtSi = m_Si*Si_temp[k] * ((wtC / m_C) + ((100 - wtC) / m_Fe));
fprintf(file, "%f %f\n", wtC, wtSi);
}
fclose(file);
t += dt;
while (t <= 13 * (24 * 60 * 60))
{
if (t % 3600 == 0)
{
printf("%d\n", t);
}
for (k = 1; k < 99; k++)
{
C_temp[k] = C[k] + (dt / pow(dx, 2))*(sqrt(D_CC(C[k + 1], Si[k + 1])*D_CC(C[k], Si[k]))*(C[k + 1] - C[k]) - sqrt(D_CC(C[k - 1], Si[k - 1])*D_CC(C[k], Si[k]))*(C[k] - C[k -
1]))
+ (dt / pow(dx, 2))*(sqrt(D_CSi(C[k + 1], Si[k + 1])*D_CSi(C[k], Si[k]))*(Si[k + 1] - Si[k]) - sqrt(D_CSi(C[k - 1], Si[k - 1])*D_CSi(C[k], Si[k]))*(Si[k] - Si[k - 1]));
Si_temp[k] = Si[k] + (dt / pow(dx, 2))*(sqrt(D_SiC(C[k + 1], Si[k + 1])*D_SiC(C[k], Si[k]))*(C[k + 1] - C[k]) - sqrt(D_SiC(C[k - 1], Si[k - 1])*D_SiC(C[k], Si[k]))*(C[k] -
C[k - 1]))
+ (dt / pow(dx, 2))*(sqrt(D_SiSi(C[k + 1], Si[k + 1])*D_SiSi(C[k], Si[k]))*(Si[k + 1] - Si[k]) - sqrt(D_SiSi(C[k - 1], Si[k - 1])*D_SiSi(C[k], Si[k]))*(Si[k] - Si[k - 1]));
}
```

2. Programmed code

* Description of Program

```
for (k = 1; k < 99; k++) { C[k] = C_temp[k]; Si[k] = Si_temp[k]; }
if (t == (24 * 60 * 60) || t == 4 * (24 * 60 * 60) || t == 7 * (24 * 60 * 60) || t == 10 * (24 * 60 * 60) || t == 13 * (24 * 60 * 60))
{
_itoa((int)(t / (24 * 60 * 60)), filename, 10);
strcat(filename, ".txt");
file = fopen(filename, "w");
fprintf(file, "%d\n", t);
for (k = 0; k < 100; k++)
{
C_temp[k] = C[k] / (1 + C[k]);
Si_temp[k] = Si[k] * (1 - C_temp[k]);
wtC = ((C_temp[k] / m_Fe) / ((1 / m_C) - ((1 / m_C) - (1 / m_Fe))*C_temp[k])) * 100;
wtSi = m_Si*Si_temp[k] * ((wtC / m_C) + ((100 - wtC) / m_Fe));
fprintf(file, "%f %f\n", wtC, wtSi);
}
fclose(file);
}
t += dt;
}
}
```

2. Programmed code

* Description of Program

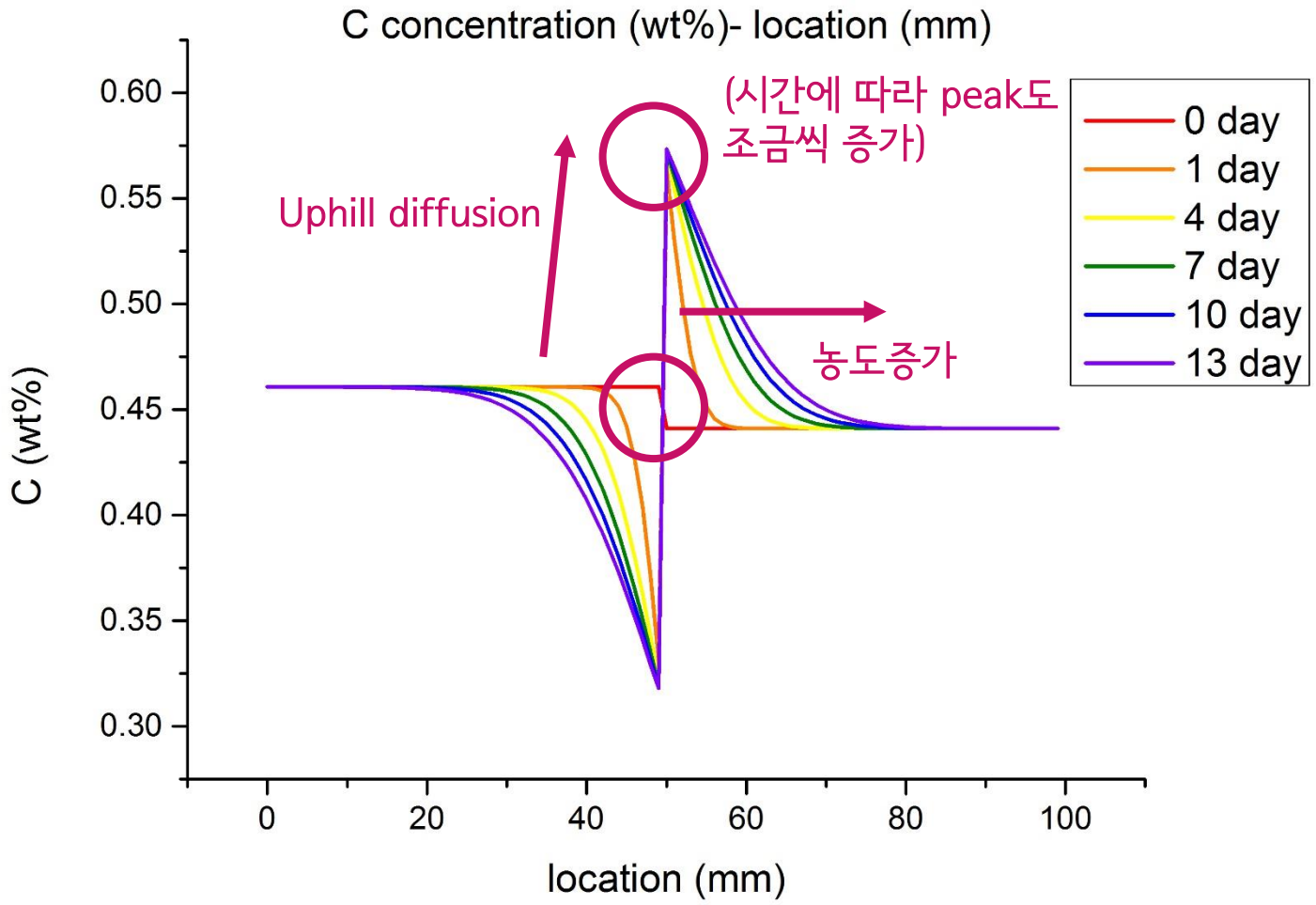
4. Gilwoon.h

```
#ifndef __GILWOON_H__//헤더가 정의 되지 않았을 때만 아래를 실행
#define __GILWOON_H__//헤더 정의

//defalut information 출력 관련 함수 모음
void Program_explanation();
void Program_info();

//실행 관련 함수
void FDM();

//Material properties 관련 함수 모음
double G_FeVa();
double G_SiVa();
double G_FeC();
double G_SiC();

double L_FeSiVa(double y_Si);
double L_FeCVa(double y_Si);

double O_Fe();
double O_Si();
double O_C(double y_C);
```

```
double U_C(double y_C, double y_Si);
double dG_SiSi(double y_C, double y_Si);

double dU_CC(double y_C, double y_Si);
double dU_CSi(double y_C, double y_Si);
double dU_SiC(double y_C, double y_Si);
double dU_SiSi(double y_C, double y_Si);
double dU_FeC(double y_C, double y_Si);
double dU_FeSi(double y_C, double y_Si);

double D_CC(double y_C, double y_Si);
double D_CSi(double y_C, double y_Si);
double D_SiC(double y_C, double y_Si);
double D_SiSi(double y_C, double y_Si);

#endif
```

3. Result & Conclusion

1. Diffusion simulation result



Chemical potential에 의존!!

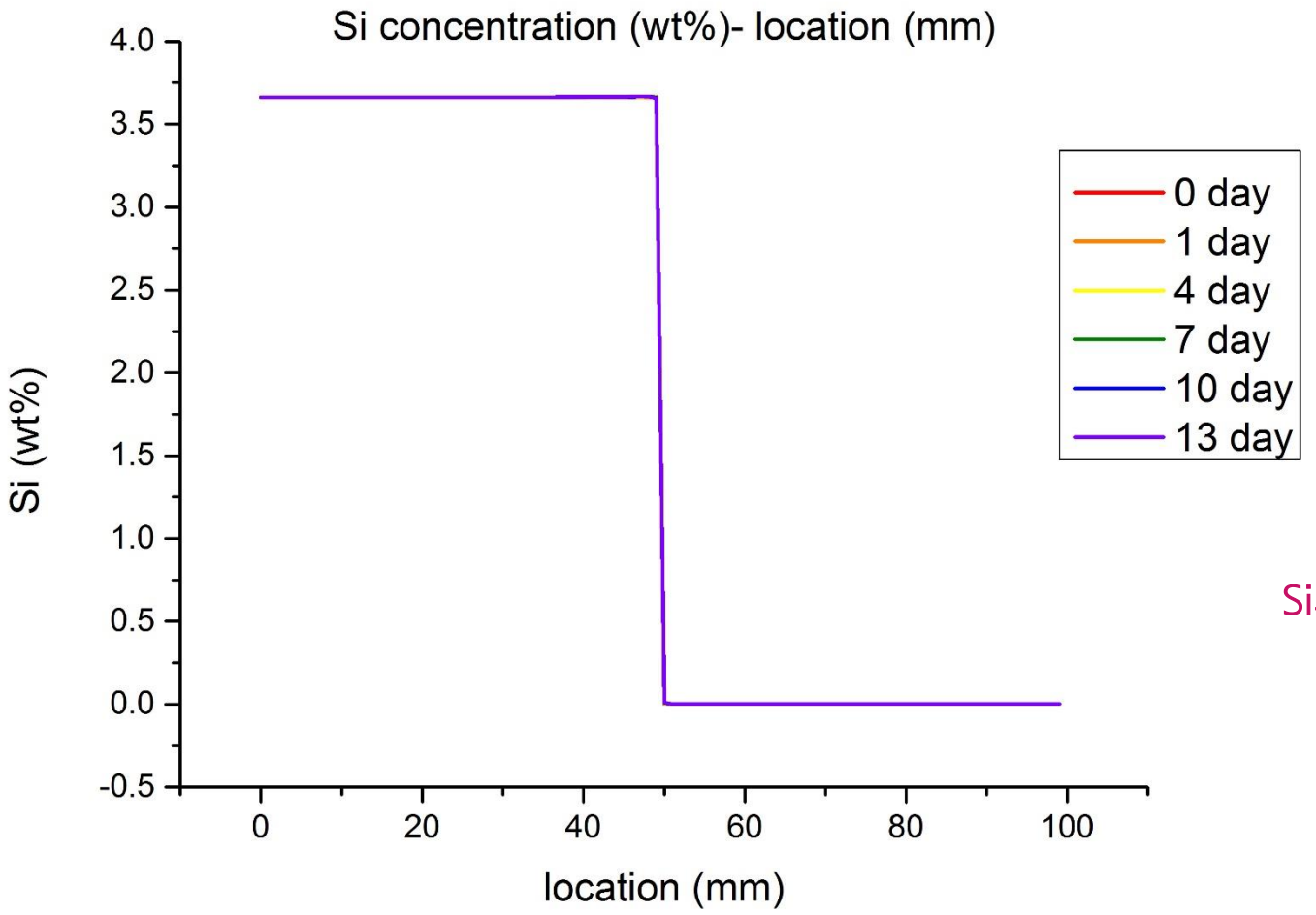
$$\Delta x = 0.001\,m,\ \Delta t = 1\,sec$$

$$\lambda < \frac{1}{2}$$

For each diffusion coeff.

3. Result & Conclusion

1. Diffusion simulation result



Si의 diffusion 모델에서는 농도 차이가 크게 없음을 확인

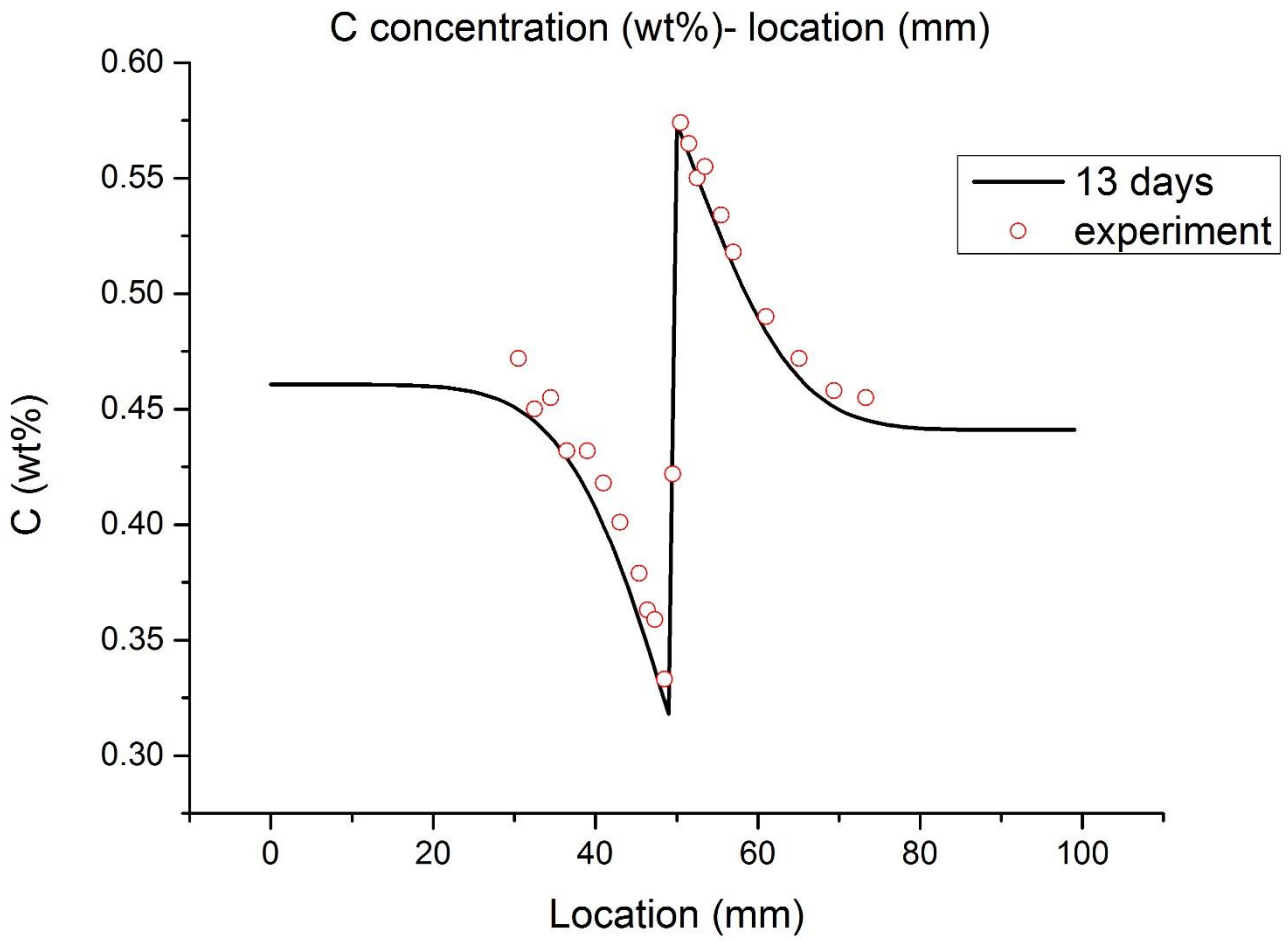
$$\Delta x = 0.001\,m,\ \Delta t = 1\,sec$$

$$\lambda < \frac{1}{2}$$

For each diffusion coeff.

3. Result & Conclusion

1. Diffusion simulation result



Final project #2

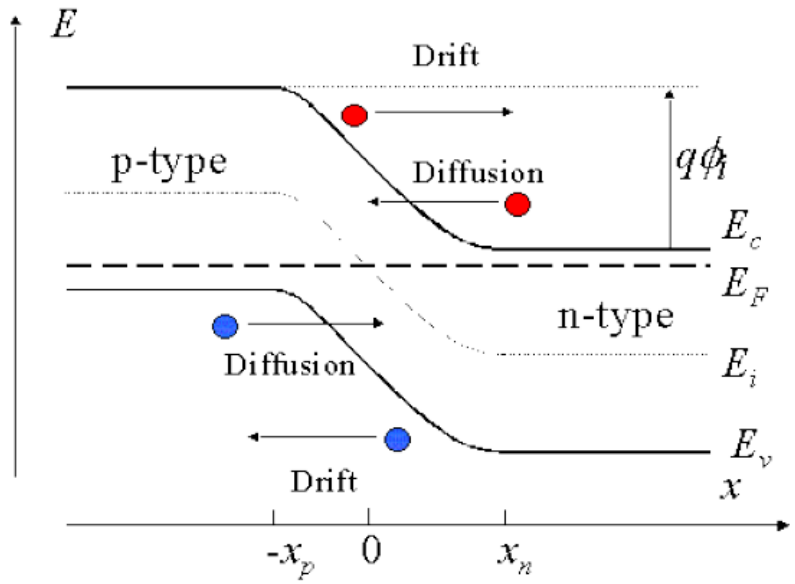
STM modeling of Si with ion implantation

1. Theory

Aimed simulation

2. pn junction: structure

- Energy band in pn junction



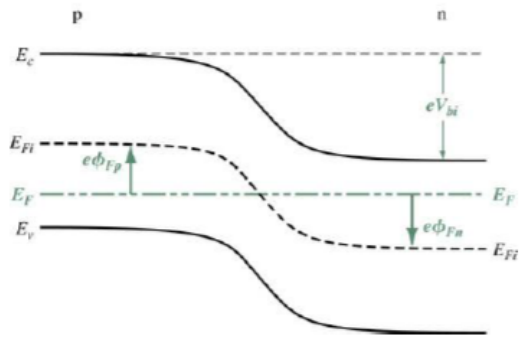
Energy band in thermal Eq. state

- In Thermal Eq. state: flat Fermi state

$$E_i - E_F = kT\ln\left(\frac{n_0}{n_i}\right)(n-type)$$

$$E_F - E_i = kT\ln\left(\frac{p_0}{n_i}\right)(p-type)$$

$$kT\ln\left(\frac{p_0}{n_i}\right)+kT\ln\left(\frac{n_0}{n_i}\right)=kT\ln\left(\frac{n_0p_0}{n_i^2}\right)=kT\ln\left(\frac{N_dN_a}{n_i^2}\right)[eV]$$



POSTECH

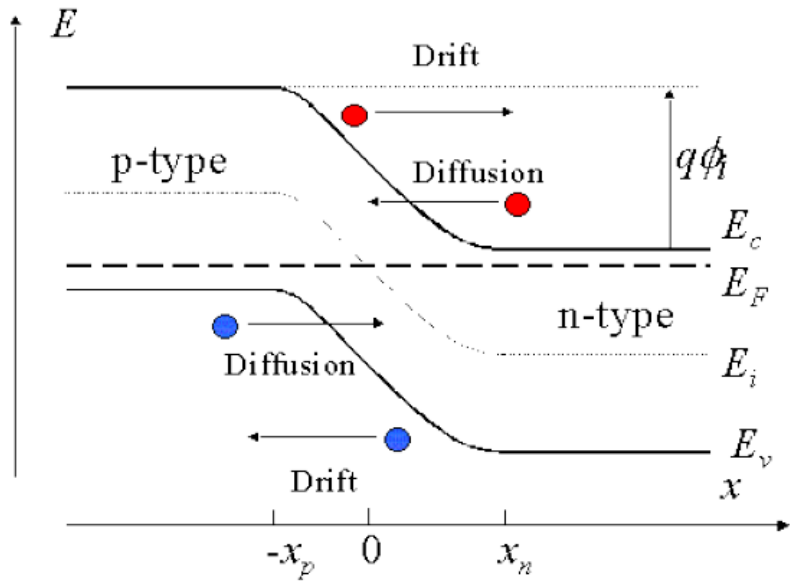
1. Theory

Aimed simulation

AMSE451 Nano Electronics — pn junction: principle and its application

2. pn junction: structure

- Energy band in pn junction



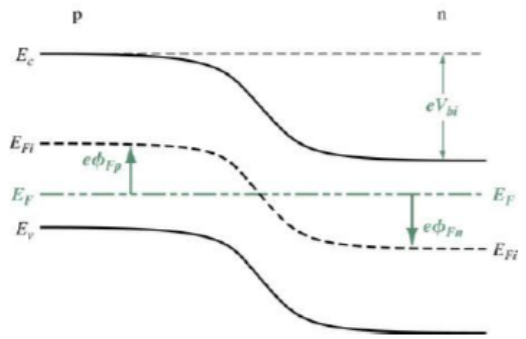
Energy band in thermal Eq. state

- In Thermal Eq. state: flat Fermi state

$$E_i - E_F = kT\ln\left(\frac{n_0}{n_i}\right)\ (n-type)$$

$$E_F - E_i = kT\ln\left(\frac{p_0}{n_i}\right)\ (p-type)$$

$$kT\ln\left(\frac{p_0}{n_i}\right)+kT\ln\left(\frac{n_0}{n_i}\right)=kT\ln\left(\frac{n_0p_0}{n_i^2}\right)=kT\ln\left(\frac{N_dN_a}{n_i^2}\right)[eV]$$



Failed

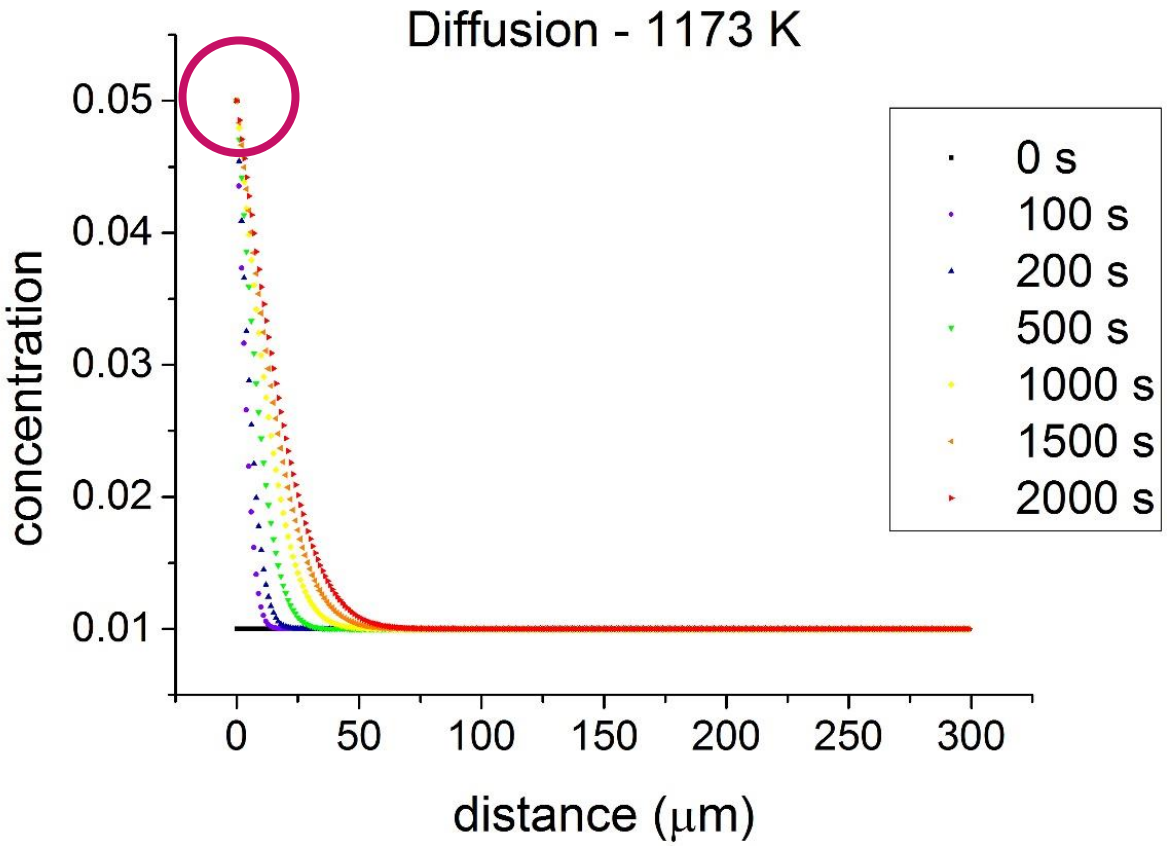
FDM시 drift에서 오류발생

POSTECH

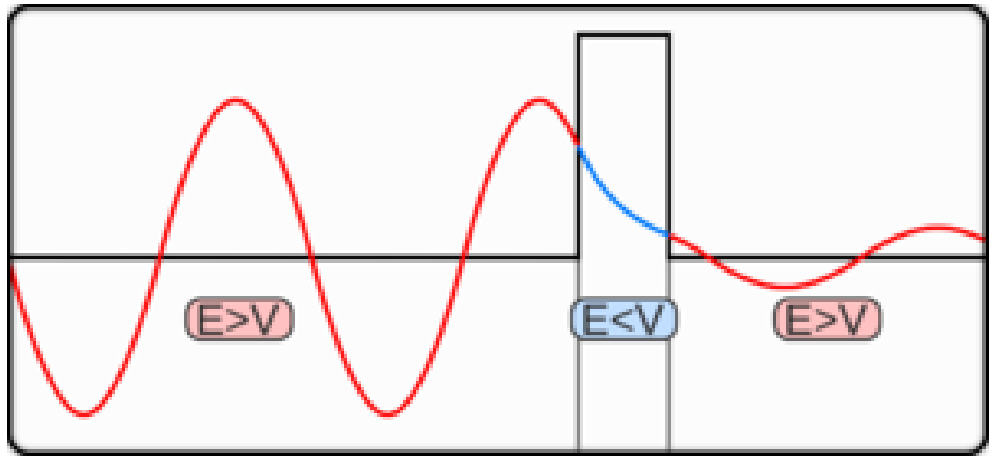
1. Theory

(Homework #9)

고정된 농도 -> 달라진다면?



(Tunneling)

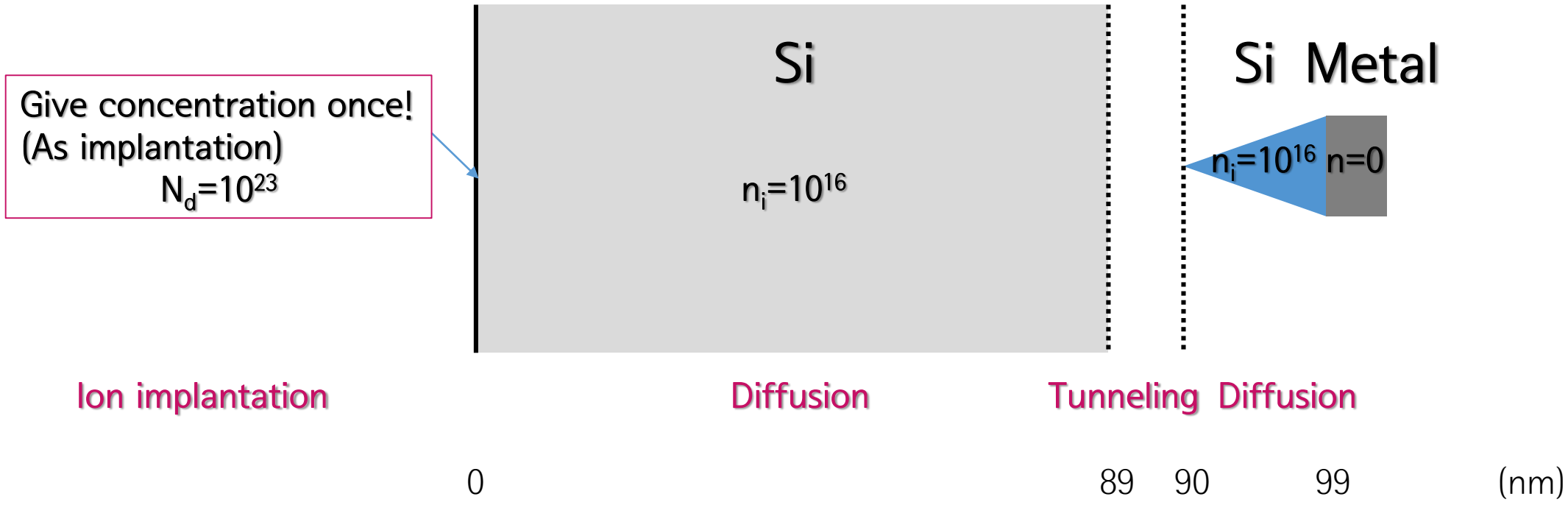


Ion implantation
+ Diffusion (equal quantity in solid)
+ Tunneling (vacuum region in STM)

1. Theory

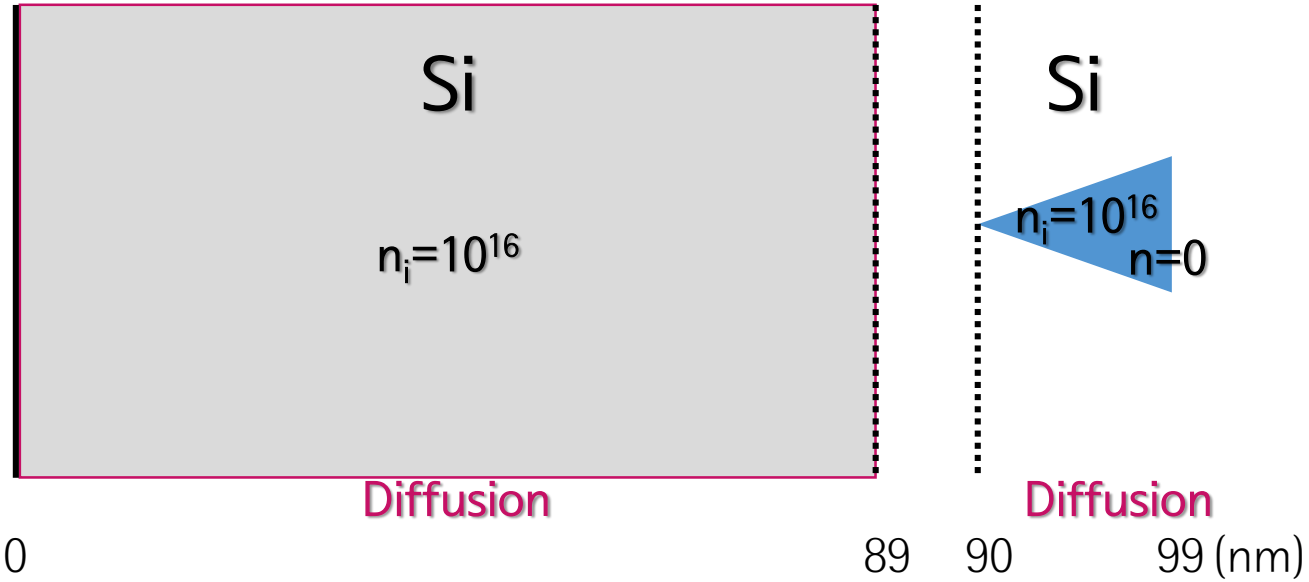
Problem situation

(표시되지 않은 모든 단위는 SI 기본 단위계, SI 기본 단위계가 아닌 경우 변환하여 계산하였음)



1. Theory

Diffusion

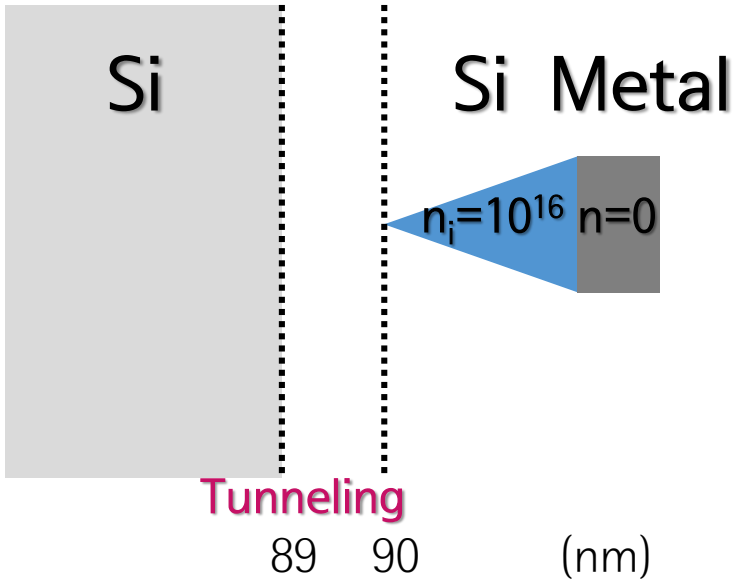


$T = 273\ K$ $\mu_e = 1000\ \mathrm{cm^2V^{-1}s^{-1}}$ for $\mathrm{N_d} = 10^{23}$ in Si

$D_e = kT\mu_e$ (Einstein relation)

$$n_i^{j+1} = \left(1 - \frac{2D_e\Delta t}{\Delta x}\right) n_i^j + \frac{D_e\Delta t}{\Delta x}\left(n_{i-1}^j + n_{i+1}^j\right)$$ (FDM)

Tunneling



$E_K \sim 3kT = 0.0777\ \mathrm{eV}$ $V = 4\ \mathrm{eV}$

$E_{C(Si)} - E_{D(As)} = 0.032\ \mathrm{eV} = E_{b(Si)}$ $E = 0.045\ \mathrm{eV}$

(S.O.Kasap, *Principles of electronic Materials and Devices*)

$$T = \left\{1 + \frac{(e^{\kappa L} - e^{-\kappa L})^2}{16\varepsilon(1-\varepsilon)}\right\}^{-1}$$

(P. Atkins, *Physical Chemistry*)

2. Programmed code

* Important codes

1. Diffusion, Tunneling

```
n_temp[0] = (1 - D_e*dt / dx)*n[0] + (D_e*dt / dx)*n[1];
for (k = 1; k < 89; k++)
{
n_temp[k] = (1 - 2 * D_e*dt / dx)*n[k] + (D_e*dt / dx)*(n[k - 1] + n[k + 1]);
}
n_temp[89] = (1 - D_e*dt / dx)*n[89] + (D_e*dt / dx)*(n[88]) - n[89] * T;
n_temp[90] = n[90] + n[89] * T - n[90]* D_e*dt / dx + (D_e*dt / dx)* n[91];
for (k = 91; k < 99; k++)
{
n_temp[k] = (1 - 2 * D_e*dt / dx)*n[k] + (D_e*dt / dx)*(n[k - 1] + n[k + 1]);
}
n_temp[99] = 0;
for (k = 0; k < 100; k++) { n[k] = n_temp[k]; }
```

2. Tunneling barrier modeling

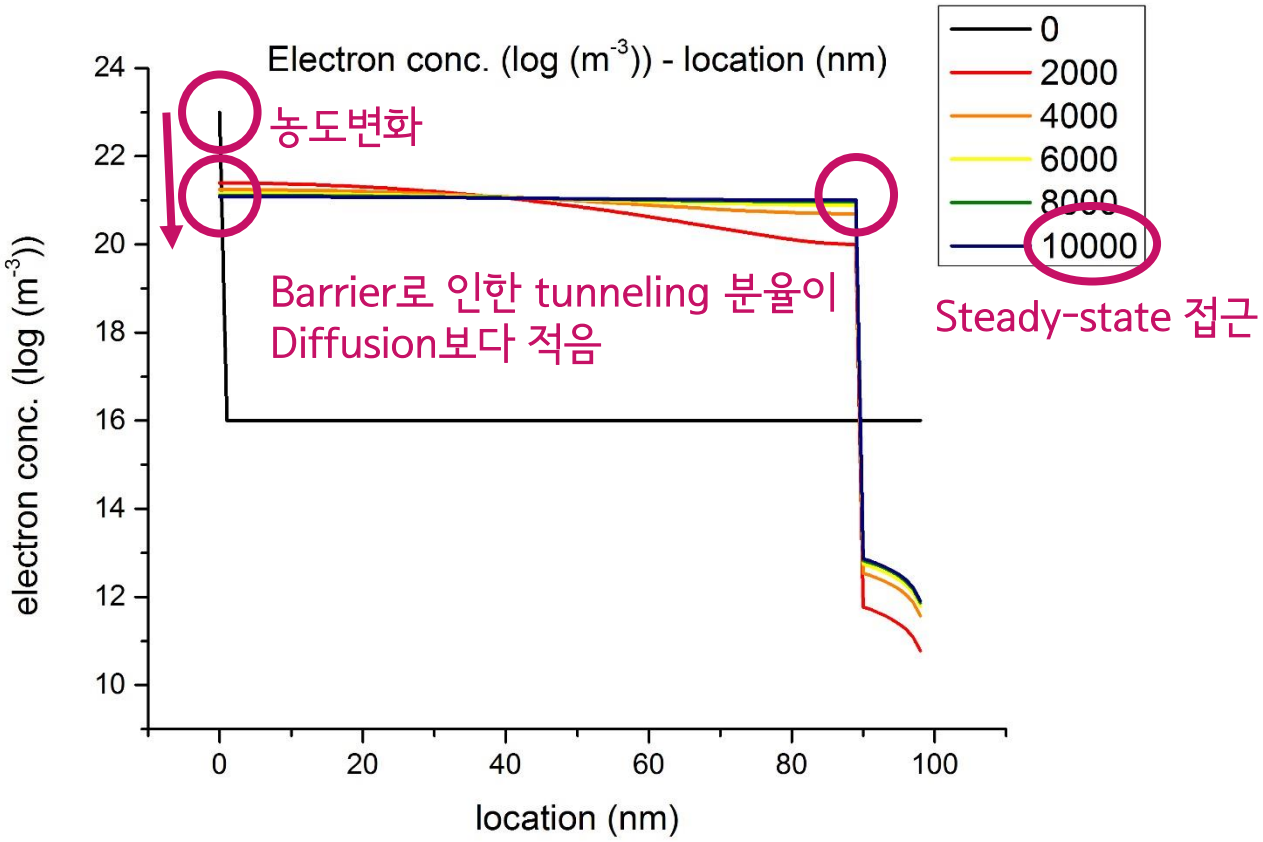
```
//Tunneling constant
E = 3*KT-0.032;
V = E + 4.0;
epsilon = E / V;
kappa = sqrt(2 * m*(V - E)*1.6*pow(10, -19) )/ hbar;
T = 1 / (1 + pow((exp(kappa*dx) - exp(-kappa*dx)), 2) / (16 * epsilon*(1 - epsilon)));
```

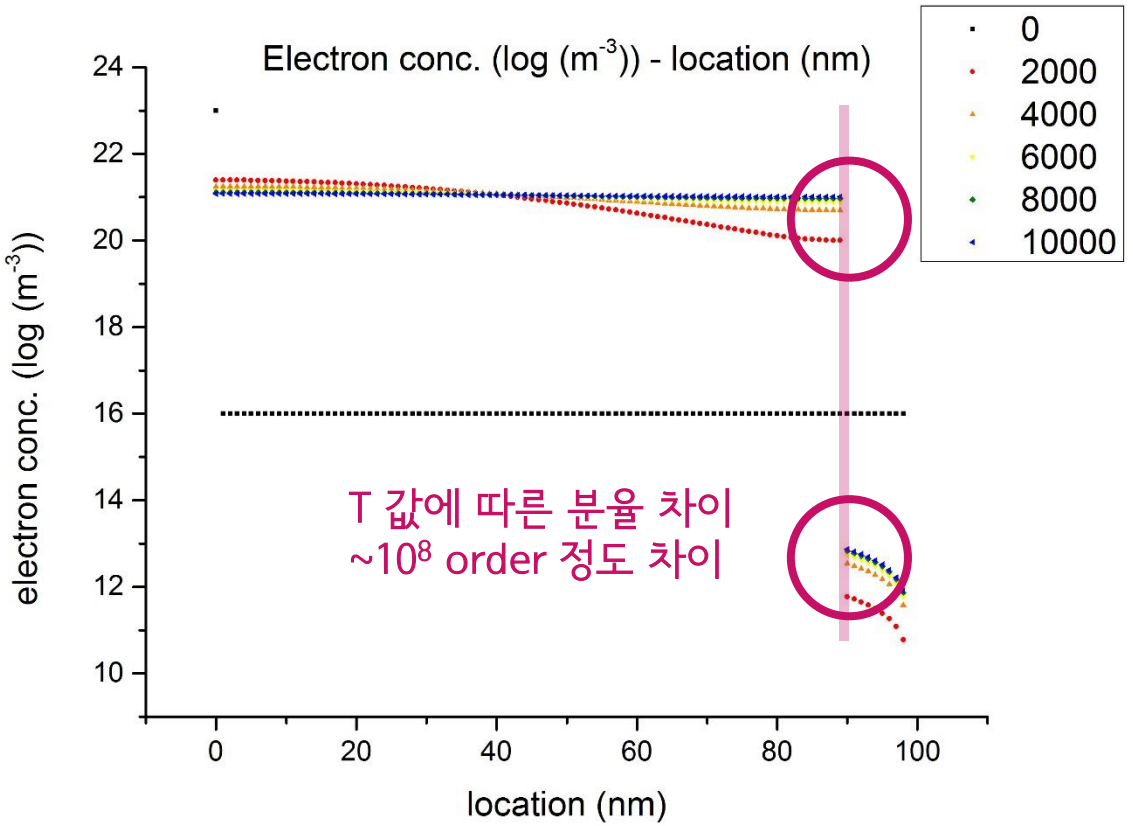
3. Ion implantation, Initial value

```
//Ion Implantation + Initial value
n[0] = pow(10, 23);
for (k = 1; k <= 89; k++) { n[k] = pow(10, 16); }
for (k = 90; k < 99; k++) { n[k] = pow(10, 16); }
n[99] = 0;
```

3. Result & Conclusion

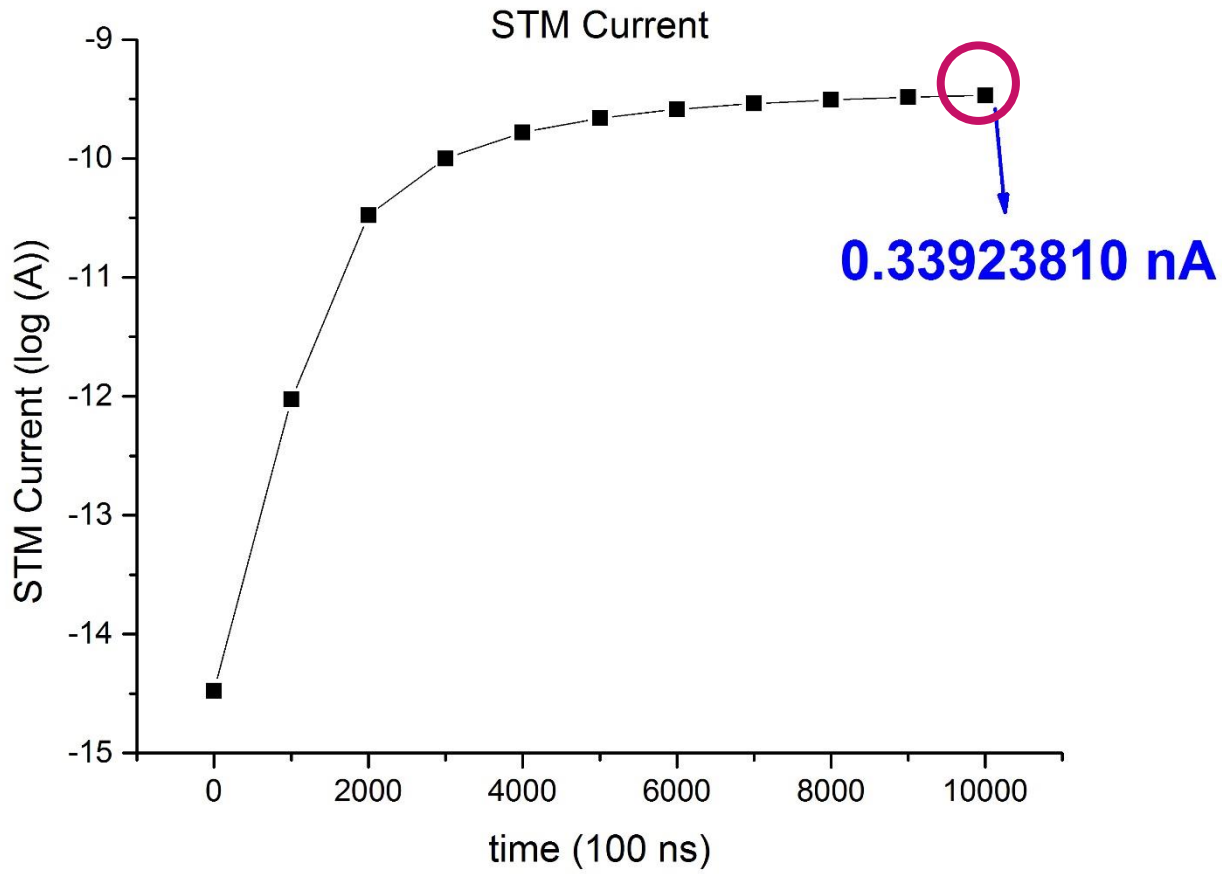
1. Simulation result



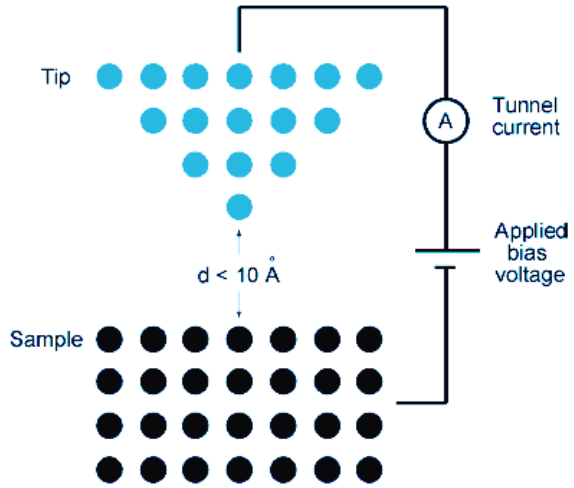


3. Result & Conclusion

1. Simulation result



In the technique of STM a sharp metal tip is brought very close (< 10 Å) to a conducting surface (*Fig.1*). When a bias voltage is placed across the tip – sample junction, electrons quantum mechanically tunnel across the gap and produce a measurable tunneling current (typically from 10pA to 10nA). This current has an exponential dependence on the tip – sample separation, resulting in atomic resolution of surface features.



http://www.diss.fu-berlin.de/diss/servlets/MCRFileNodeServlet/FUDISS_derivate_000000001253/01_chapter1.pdf?hosts=

4. Conclusion

1) 각각의 재료 현상을 시뮬레이션 해보았고, 실제 기기에서 쓰이는 원리를 시뮬레이션으로 확인하였다.
2) 시뮬레이션의 결과 값(전류 값)이 실제 구동 범위와 맞아 떨어짐을 확인하였다.
3) 조금은 힘들었지만 매우 의미 있었던 수업!! ☺ (특히 열역학을 제대로 배웠습니다!!)