소재수치해석

Assignment #9

신소재공학과 이학현

Problem

Problem) Consider injection of an alloying element B in a metallic matrix A. The initial composition of B in A is 0.01. Injection is carried out by maintaining the surface composition of B to be 0.05. The diffusion coefficient of B in A is 4.529×10^{-7} exp[-147723(J)/RT] (m^2/s). The injection temperature is between 1173K and 1473K. Injection distance is defined to be the distance from the surface of a point where the composition of B is half of the target value (0.03). Perform the followings:

(a) How does the injection distance depend on injection time?

(b) How does the injection distance depend on temperature?

(c) How can you determine the activation energy for the reaction, and what is it?

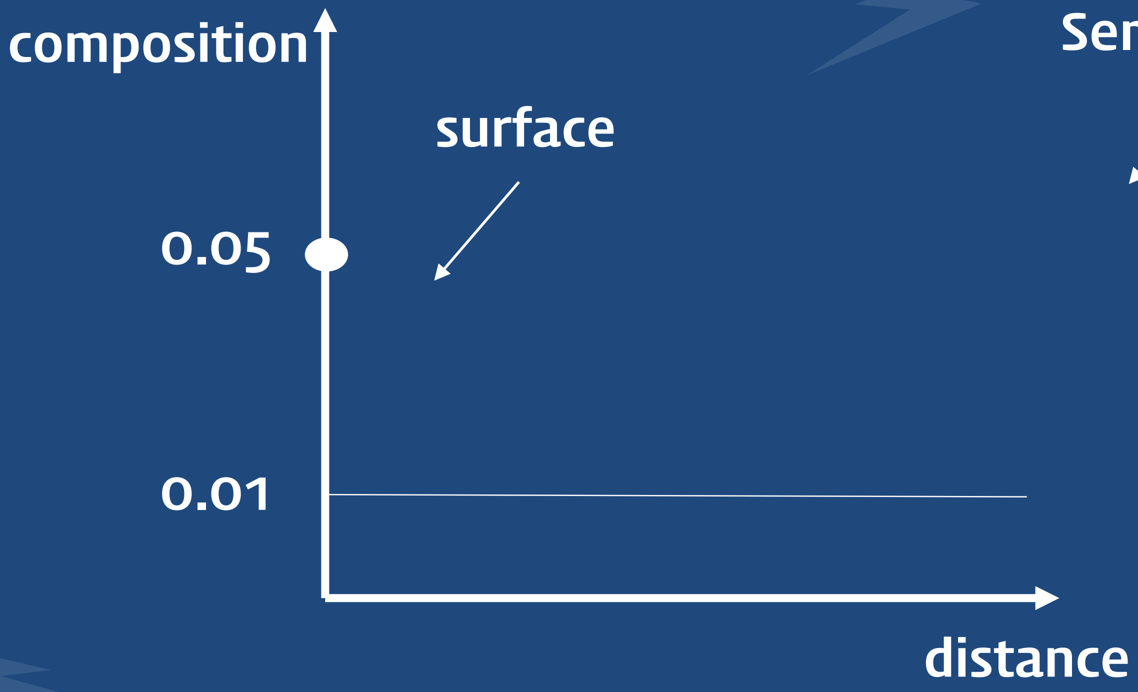
- **Initial composition of B in A** : 0.01
- **Surface composition of B** : 0.05
- **Diffusion coefficient of B in A** : 4.529×10^{-7} exp[-147723(J)/RT] (m^2/s)
- **Injection temperature** : 1173~1473K
- **Injection distance** : B농도가 0.03이 되는 지점

이론적 배경

Diffusion partial differential equation

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

U : composition
α : diffusion coefficient



이론적 배경

Explicit method

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

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

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

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

λ

$$\boxed{\frac{\alpha\Delta t}{\Delta x^2}\le\frac{1}{2}}$$

stability

$$A=\begin{bmatrix}(1-2\lambda) & \lambda & 0 & \cdots & 0\\ \lambda & (1-2\lambda) & \lambda & \ddots & \vdots\\ 0 & \ddots & \ddots & \ddots & 0\\ \vdots & & \ddots & \ddots & \lambda\\ 0 & \cdots & 0 & \lambda & (1-2\lambda)\end{bmatrix}$$

$$\mathbf{w}^{(0)}=(f(x_1),f(x_2),\ldots,f(x_{m-1}))^t$$

$$\mathbf{w}^{(j)}=(w_{1j},w_{2j},\ldots,w_{m-1,j})^t,\quad \text{for each } j=1,2,\ldots,$$

$$\mathbf{w}^{(j)}=A\mathbf{w}^{(j-1)},\quad \text{for each } j=1,2,\ldots.$$

Key code

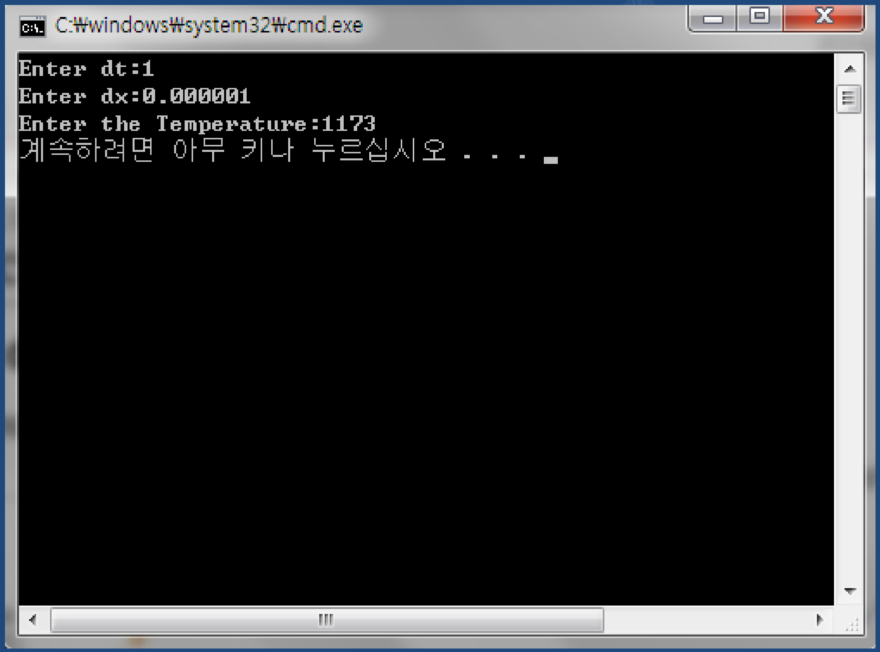
```c
result=fopen("result.txt","w");

printf("Enter dt:");
scanf("%lf",&dt);

printf("Enter dx:");
scanf("%lf",&dx);

printf("Enter the Temperature:");
scanf("%lf",&T);

n=1000/dt; // 데이터 계산 속도에 영향
D=4.529*pow(10.0,-7)*exp((-147723)/(R*T));
lamda=(D*dt)/(dx*dx);
x=0;

matrix=(double**)malloc(sizeof(double*)*n);
w=(double**)malloc(sizeof(double*)*n);
```

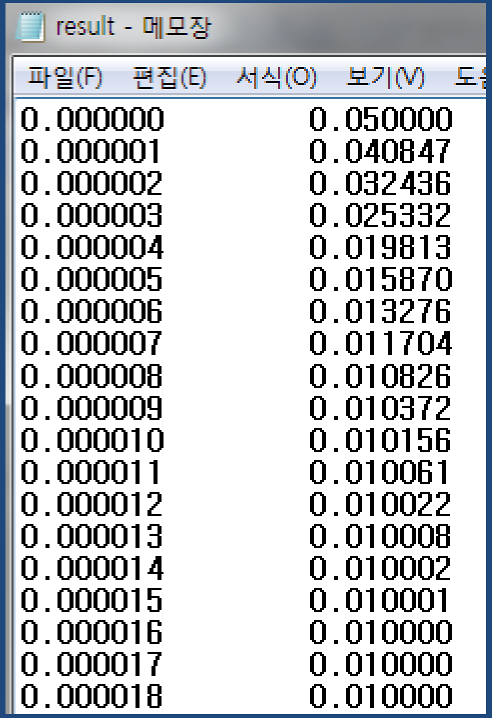
```c
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
        matrix[i][j]=0;
}

for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        if(i==j)
            matrix[i][j]=1-2*lamda;
    }
    for(j=0;j<n;j++)
    {
        if(j==i+1)
            matrix[i][j]=lamda;
    }
    for(j=0;j<n;j++)
    {
        if(j==i-1)
            matrix[i][j]=lamda;
    }
}
```

```c
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        if(i==0)
            w[i][j]=0.05;
        else
            w[i][j]=0.01;
    }
}
```

```c
for(i=0;i<n-1;i++)
{
    x=0;
    for(j=0;j<n-1;j++)
    {
        temp=0;
        for(k=0;k<n;k++)
        {
            temp=temp+matrix[j+1][k]*w[k][i];
        }
        w[j+1][i+1]=temp;
        if(i+1==50) // 데이터 뽑을때 여기 바꿔주기
        {
            for(l=0;l<n;l++)
            {

                if(fabs(w[l][i]-0.01)>0.00000001)
                    fprintf(result,"%lf %lf\n",x,w[l][i]);
                x=x+dx;
            }
        }
```

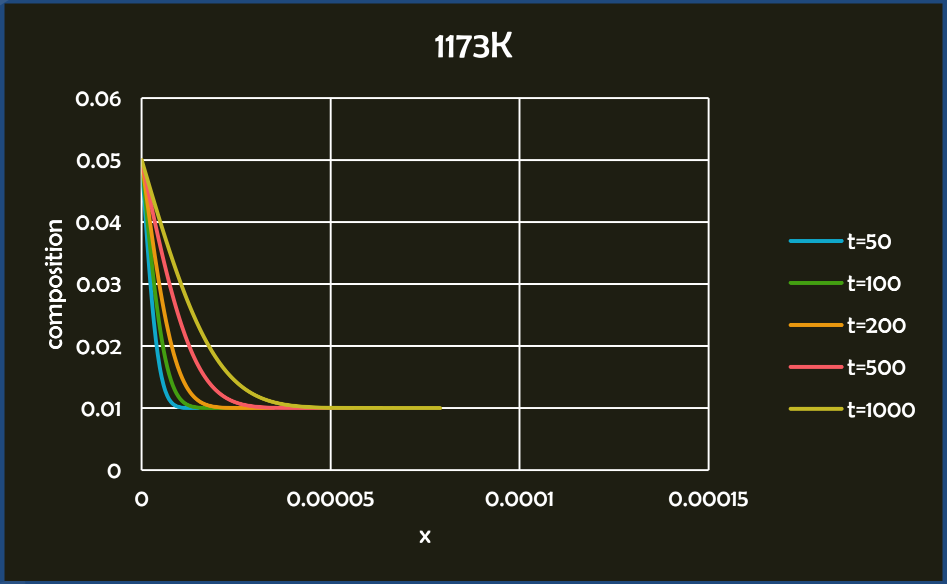
Key code



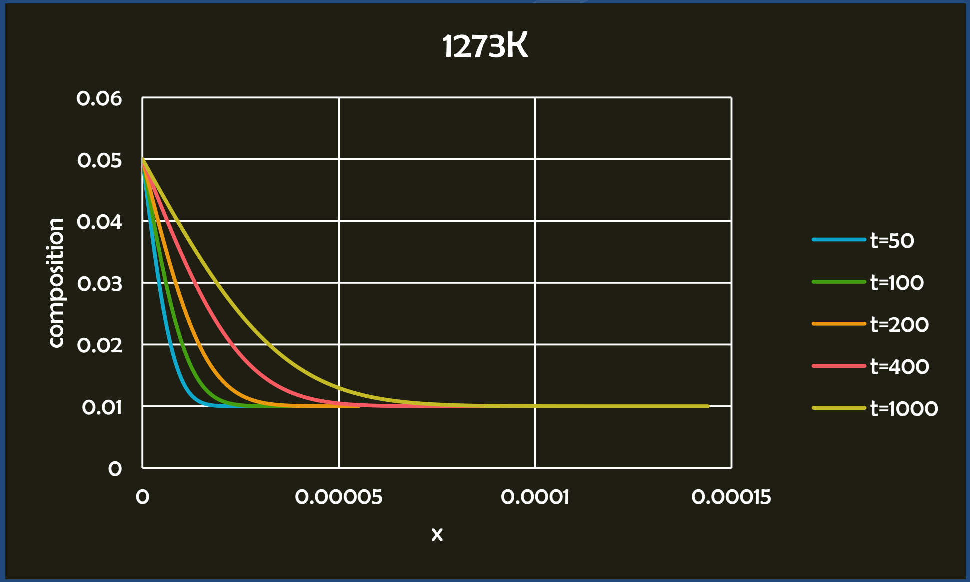


0.000000	0.050000
0.000001	0.040847
0.000002	0.032436
0.000003	0.025332
0.000004	0.019813
0.000005	0.015870
0.000006	0.013276
0.000007	0.011704
0.000008	0.010826
0.000009	0.010372
0.000010	0.010156
0.000011	0.010061
0.000012	0.010022
0.000013	0.010008
0.000014	0.010002
0.000015	0.010001
0.000016	0.010000
0.000017	0.010000
0.000018	0.010000

How does the injection distance depend on injection time?



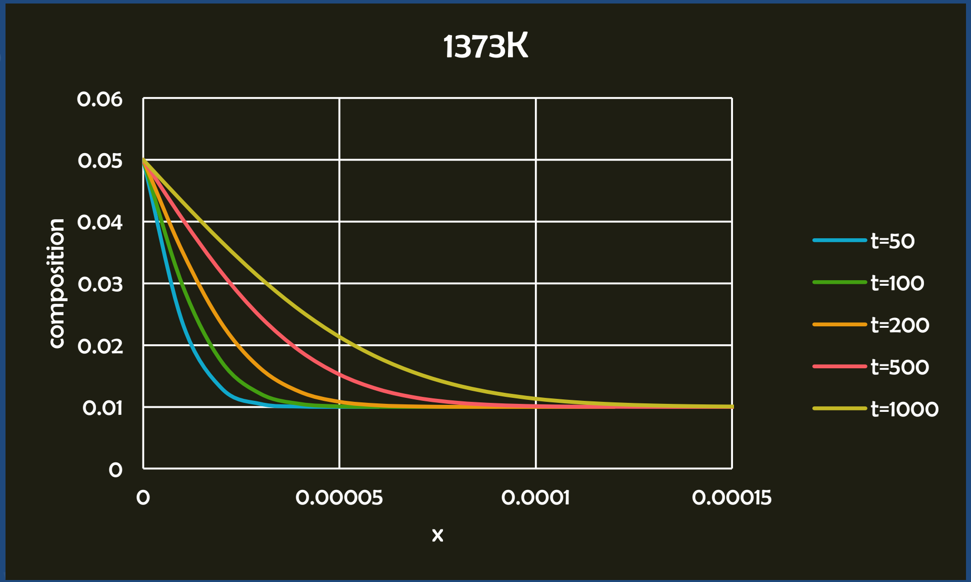
dt=1, dx=1um
λ= 0.1196479



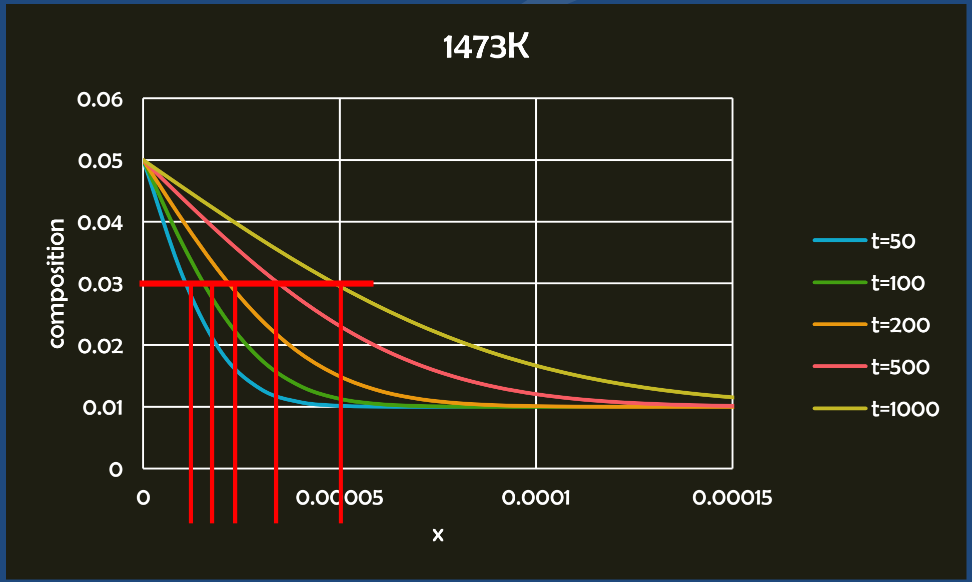
dt=1, dx=1um
λ =0.3932285

How does the injection distance depend on injection time?

dt=1, dx=10um



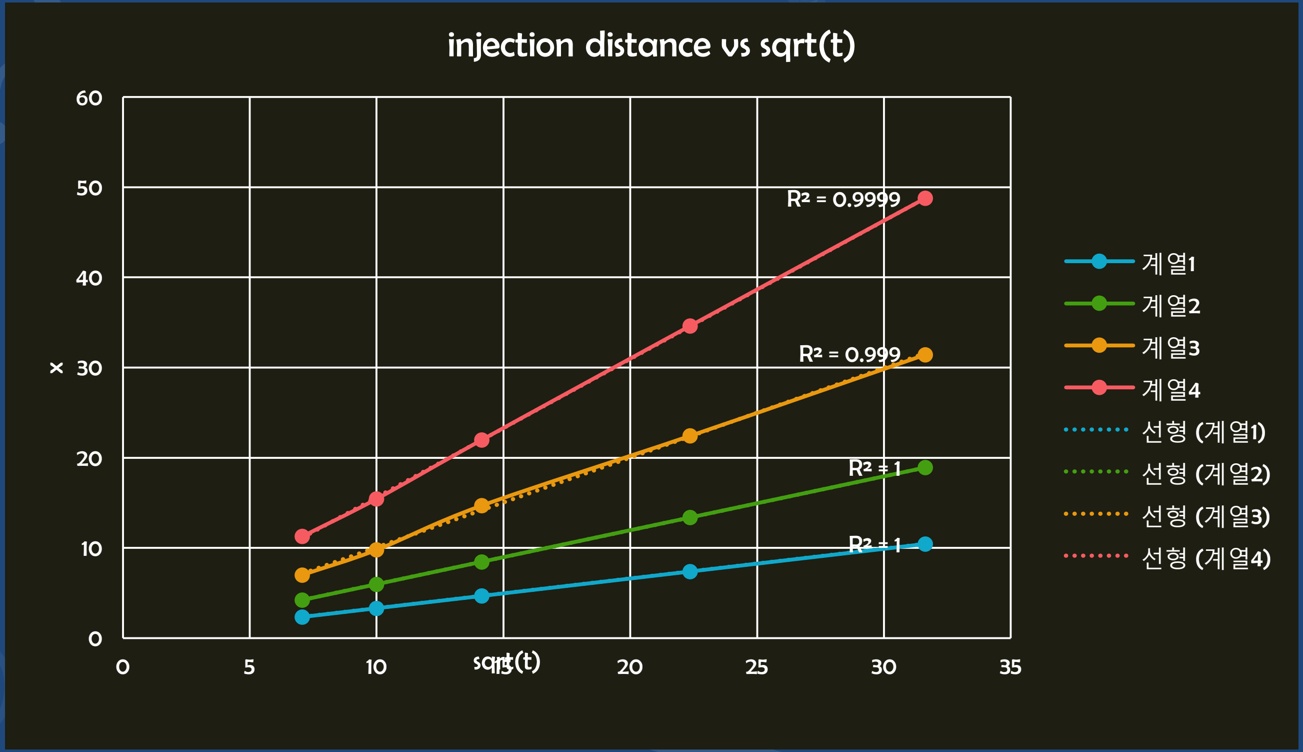
dt=1, dx=10um



Injection time 이 증가할 수록, injection distance 증가

How does the injection distance depend on injection time?

Composition이 0.03인 지점의 injection distance를 알기 위해 cubic spline 적용



Variable : T, t, x

T : fixed

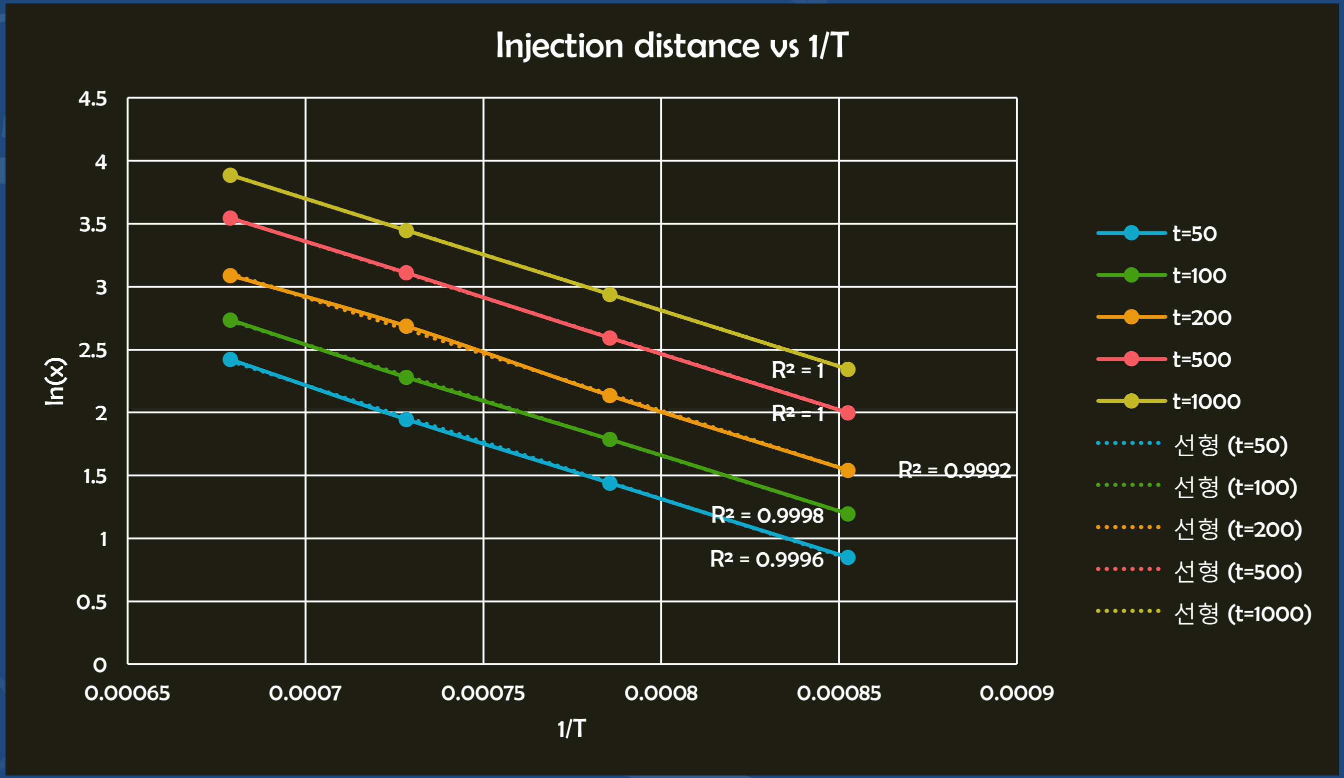
$$u(x,t) = u_0 \mathrm{erfc}\left(\frac{x}{2\sqrt{Dt}}\right)$$

각 온도에서 u, u0, D Fix

$$x \propto \sqrt{Dt}$$

$$x \propto \sqrt{t}$$

How does the injection distance depend on Temperature?



Variable : T, t, x
t : fixed

$$x \propto \sqrt{Dt}$$

$$x \propto \sqrt{D} \propto \sqrt{\exp(-\frac{1}{T})}$$

$$\ln(x^2) \propto -\frac{1}{T}$$

$$\boxed{\ln x \propto \frac{1}{T}}$$

How can you determine the activation energy for the reaction, and what is it?

ex) for a diffusion controlled reaction, $t_f \propto 1 / D$

$$D = D_o \cdot exp\,(-Q^{diff}/RT)$$

$$Q^{reac} = R \cdot \frac{\partial(\ln 1/D)}{\partial(1/T)} = -R\frac{\partial(-Q^{diff}/RT)}{\partial(1/T)} = Q^{diff}$$

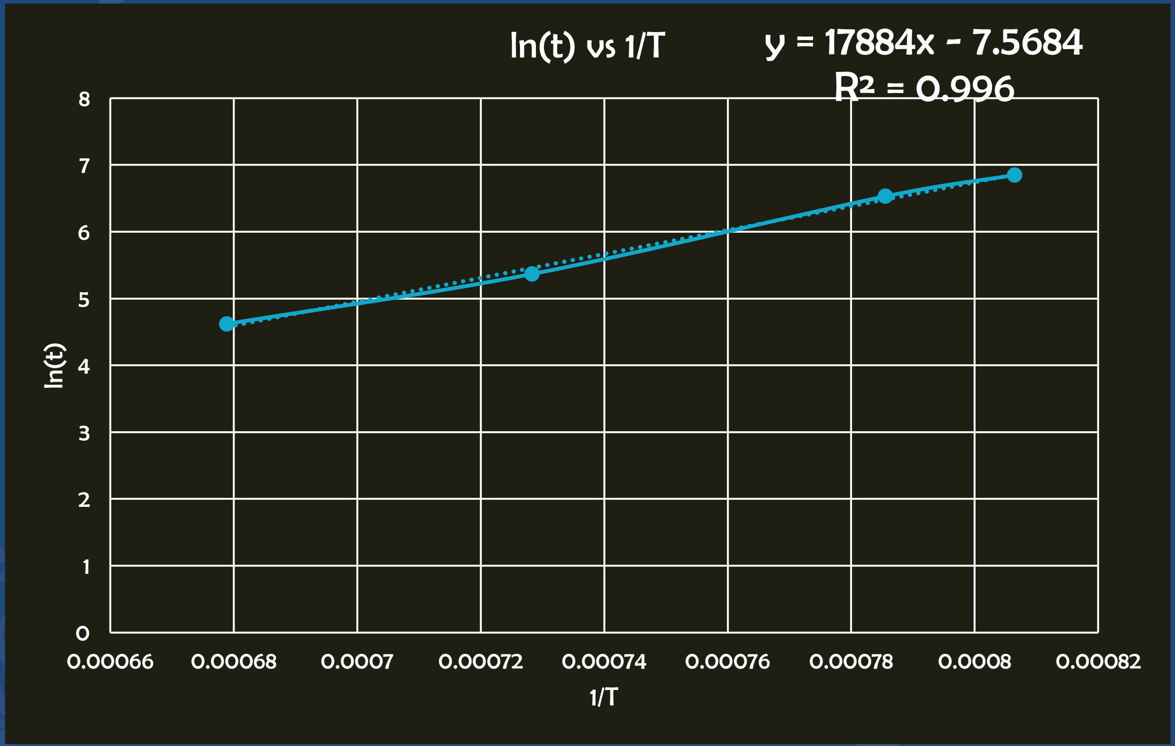
$$Q^{reac} = R \cdot \frac{\partial(\ln t_f)}{\partial(1/T)}$$

Injection distance 를 고정 시킨 후, 각 온도의 시간에 따른 composition의 data를 통해서, cubic spline으로 composition이 0.03 일 때의 해당 온도에 따른 시간을 data를 추출 후 plotting

위의 relationship에 따라 ln(t)와 1/T 사이의 기울기에 R을 곱한 값은 Reaction activation energ와 같고, 이는 diffusion activation energy 와 같다.

How can you determine the activation energy for the reaction, and what is it?

Variable : T, t, x
X : fixed (15um)



Q= 17884*8.3144621
= 148695.8402 J

Qdiff=147723 J

상대 오차 = 0.65%

Conclusion

- Explicit method를 이용하여 diffusion differential equation을 풀 수 있다.

- Stability 조건을 만족 시키기 위해서 dx, dt 값을 적절히 고려해주어야 한다.

Discussion

0.000069	0.010000
0.000070	0.010000
0.000071	0.010000
0.000072	0.010000
0.000073	0.010000
0.000074	0.010000
0.000075	0.010000
0.000076	0.010000
0.000077	0.010000
0.000078	0.010000
0.000079	0.010000
0.000925	0.010000
0.000926	0.010000
0.000927	0.010000
0.000928	0.010000
0.000929	0.010000
0.000930	0.010000
0.000931	0.010000
0.000932	0.010000
0.000933	0.010000
0.000934	0.010000
0.000935	0.010000
0.000936	0.010000
0.000937	0.010000
0.000938	0.009999
0.000939	0.009999
0.000940	0.009999
0.000941	0.009999
0.000942	0.009998
0.000943	0.009998
0.000944	0.009997
0.000945	0.009996