



# Numerical Analysis for Materials

Homework No.4

20130528  
Dept. of Materials Science and Engineering  
Cheolhee Han

# Inverse Matrix + Newton-Raphson



## Functions

- Multi-Component Phase Equilibria related functions (x: Si, 1-x: Ge)  
`double fGe = 36944.72 - 30.4975*T + R*T*log((1-xl) / (1-xs));`  
`double fSi = 50208.00 - 29.7617*T + R*T*log(xl / xs);`
- Inverse Matrix related functions
- Jacobian Matrix

## Algorithm

1. T\_min과 T\_max,  $x_{Si}^L$ 과  $x_{Si}^U$ 의 초기 값을 설정한다.
2. 예외 처리1: 역행렬 존재 여부를 확인하여 연산이 가능할 경우, Jacobian의 역행렬을 구한다.
3. 역행렬을 이용해 nonlinear equation system에 대한 Newton's Method를 시행한다.
4. 예외 처리2: x값이 0과 1 사이를 벗어날 경우, rand()를 이용하여 0과 1사이의 값을 임의로 지정해준다.
5. T를 T\_max까지 올라가면서 2~4의 과정을 시행한다.

# int main(void)

```
x1 = 0.4;
xs = 0.6;
for (T = T_min; T < T_max; T++)
{
    for(iter=0; iter<1000; iter++)
    {
        if (x1 == xs)
        {
            x1 += TOL;
            xs -= TOL;
        }
        fGe = Gibbs_Ge(T, x1, xs);
        fSi = Gibbs_Si(T, x1, xs);
        J[0][0] = R*T / x1;
        J[0][1] = -R*T / xs;
        J[1][0] = -R*T / (1 - x1);
        J[1][1] = R*T / (1 - xs);
        for (i = 0; i<dim; i++)
        {
            for (j = 0; j<dim; j++)
            {
                I[i][j] = (i == j) ? 1 : 0; //unit matrix의 경우 대각선에 1, 나머지는 0
            }
        }
        switch (Gauss_El(J, I, dim))
        //Gauss Elimination을 실행한다.
    }
}
```

## Main\_1

초기 조건은 임의로 0.4, 0.6으로 설정하였다.

x1과 xs가 같은 경우, 역행렬이 존재하지 않는다.  
이 때는 TOL만큼의 차이를 주어 값을 다르게 설정한다.

Gibbs\_X 함수는 diamond phase와 liquid phase의 chemical potential 차를 의미하며, 이 값이 0이 되는 점을 찾는 것이다.

Jacobian은 프로그램에서 작성하기 전에 x1과 xs에 대해 미분하여 구했으며, I는 Gauss-Jordan form으로 두어 역행렬로 만들 단위행렬이다.

# int main(void)



## Main\_2

Gauss Elimination을 거치면 I는 J의 역행렬이 된다.

Jacobian Matrix를 이용하여 nonlinear equation system을 풀 경우,  $x_{new}$ 는  $x$ 와 역행렬-chemical potential 차의 연산에 의해 결정된다.

Chemical potential 차가 Tolerance보다 작아져 0과 구분할 수 없을 경우, 답을 파일 형태로 출력하고 iteration만 복구한다.

값이 0과 1 사이를 탈출할 경우, random하게 0과 1 사이의 값을 지정해주어 예외 처리한다.

```
x1_new = x1 - (I[0][0] * fSi + I[0][1] * fGe);
xs_new = xs - (I[1][0] * fSi + I[1][1] * fGe);
if (fabs(fSi) < TOL && fabs(fGe) < TOL)
    break;
if ((x1_new < 0) || (x1_new > 1))
{
    x1_new = (double)(rand() % 10000) / 10000;
}
if ((xs_new < 0) || (xs_new > 1))
{
    xs_new = (double)(rand() % 10000) / 10000;
}
x1 = x1_new;
xs = xs_new;
}
printf("%lf %lf %lf %d\n", T, x1, xs, iter);
fprintf(result, "%lf %lf %lf %d\n", T, x1, xs, iter);
iter = 0;
}
```



# Investigation & Conclusion



## Answers

1. Chemical Potential에 관한 식을 이용하여 Ge-Si binary phase diagram을 plot할 수 있다.
2. Newton-Raphson Method를 nonlinear equation system에 적용하여 풀 수 있다.
3. 주어진 정답과 거의 일치하게 plot할 수 있다.

## 초기 조건에 따른 연산 횟수 차이

1. 초기 조건에 무관하게 3회 내에 연산을 마친다.
2. 첫 연산과 마지막 연산의 경우 비정상적으로 iteration이 튀는 것을 확인할 수 있다.
3.  $x_i$ 과  $x_s$ 를 변화시켰을 경우 극초반의 연산 횟수만 달라지고 이후 결과값과 연산 횟수가 동일하다.