Non linear equation system

20120778
Subin Lee

Advanced Electron Microscopy Nanoscience Lab.

Department of Materials Science & Engineering,
Pohang University of Science & Technology (POSTECH)

Assignment

- Gauss-Jordan 방법으로 역행렬을 구하는 프로그램을 완성하시오. 몇 개의 선형 연립방정식 예제를 만들고 완성한 프로그램을 이용, 방정식의 해를 구하시오.

- 주어진 Gibbs energy 식을 이용하여 Ge-Si 2 원계 상태도를 계산으로 완성하시오.

$$ {}^{o}G_{Ge}^{dia \rightarrow liquid} = 36944.72 - 30.4975\ T $$

$$ {}^{o}G_{Si}^{dia \rightarrow liquid} = 50208.00 - 29.7617\ T $$

고상 (diamond 구조), 액상 모두에 대해 ideal solution 을 가정하시오.

Key codes

- For inverse matrix

$$A = \begin{bmatrix} -1 & 1 & 2 \\ 3 & -1 & 1 \\ -1 & 3 & 4 \end{bmatrix}$$

으로 주어진 행렬 A의 역행렬을 구하고자 하면,

$$[\,A \mid I\,] \rightarrow \left[\begin{array}{ccc|ccc} -1 & 1 & 2 & 1 & 0 & 0 \\ 3 & -1 & 1 & 0 & 1 & 0 \\ -1 & 3 & 4 & 0 & 0 & 1 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|ccc} -1 & 1 & 2 & 1 & 0 & 0 \\ 0 & 2 & 7 & 3 & 1 & 0 \\ 0 & 2 & 2 & -1 & 0 & 1 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|ccc} -1 & 1 & 2 & 1 & 0 & 0 \\ 0 & 2 & 7 & 3 & 1 & 0 \\ 0 & 0 & -5 & -4 & -1 & 1 \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} 1 & -1 & -2 & -1 & 0 & 0 \\ 0 & 1 & 3.5 & 1.5 & 0.5 & 0 \\ 0 & 0 & 1 & 0.8 & 0.2 & -0.2 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|ccc} 1 & -1 & 0 & 0.6 & 0.4 & -0.4 \\ 0 & 1 & 0 & -1.3 & -0.2 & 0.7 \\ 0 & 0 & 1 & 0.8 & 0.2 & -0.2 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -0.7 & 0.2 & 0.3 \\ 0 & 1 & 0 & -1.3 & -0.2 & 0.7 \\ 0 & 0 & 1 & 0.8 & 0.2 & -0.2 \end{array}\right]$$

$$A^{-1} = \begin{bmatrix} -0.7 & 0.2 & 0.3 \\ -1.3 & -0.2 & 0.7 \\ 0.8 & 0.2 & -0.2 \end{bmatrix}$$

Insert unit matrix

Elimination

Normalization

Backward elimination

```
//*** Inverse matrix
void inverse_matrix(double **array, int n)
{
    double **identity_matrix;
    identity_matrix = (double **) malloc( sizeof(double*)*SIZE );
    for( int i=0; i<(SIZE); i++) {
        identity_matrix[i] = (double *) malloc( sizeof(double)*(2*SIZE) );
    }

    for (int i=0; i<SIZE; i++){
        for(int j=0; j<SIZE; j++){
            identity_matrix[i][j]= array[i][j];
        }
    }
```

```
void elimination (double ** array, int n)
{
    double factor;
    for (int i=1; i<n;i++){
        for (int j=i; j<n; j++){
            factor = array[j][i-1]/array[i-1][i-1];
            for (int k=0; k<=(n+n); k++){
                array[j][k]=array[j][k]-factor*array[i-1][k];
            }
        }
    }
}

void normalization (double ** array, int n)
{
    double factor;
    for (int i=0; i<n; i++){
        factor=array[i][i];
        for (int j=i; j<(n+n); j++){
            array[i][j]=array[i][j]/factor;
        }
    }
}

void backward_elemination (double ** array, int n)
{
    double factor;
    for (int i=n-1; i!=0;i--){
        for (int j=i-1; j!=-1; j--){
            factor = array [j][i]/array[i][i];
            for (int k=i; k<=(n+n); k++){
                array[j][k]=array[j][k]-factor*array[i][k];
            }
        }
    }
}
```

Verification for inverse matrix

```
Insert the size of the materix
5
Insert the input file name
test.dat
*** Argumented matrix ***
 -1.000000000000000 1.000000000000000 2.000000000000000 82.000000000000000 47.000000000000000
 9.000000000000000 18.000000000000000 21.000000000000000 4.000000000000000 39.000000000000000
 17.000000000000000 5.300000000000000 17.000000000000000 3.000000000000000 -1.000000000000000
 1.000000000000000 5.000000000000000 91.000000000000000 27.000000000000000 10.000000000000000
 2.000000000000000 5.700000000000000 94.000000000000000 28.000000000000000 87.000000000000000

*** Inverse matrix ***
 0.001206561184799 -0.018123995271159 0.068399668381481 -0.019464248244151 0.010496202192099
 0.001090237273387 0.069236141879050 -0.034027879581523 0.026584217641617 -0.035072652196894
 -0.003721280806946 -0.003328258290076 0.000714818521133 0.009167522964904 0.002456802989787
 0.012310501160627 -0.000838905682578 0.001591334885603 0.006908391801689 -0.007050216276554
 -0.000040472545662 -0.000253475629758 -0.000627479780824 -0.013422800841576 0.013165382618869
```

Calculated by calculator from web

```
 0.001206561184799 -0.018123995271159  0.068399668381481 -0.019464248244152  0.010496202192099
 0.001090237273387  0.069236141879050 -0.034027879581523  0.026584217641617 -0.035072652196895
-0.003721280806946 -0.003328258290076  0.000714818521133  0.009167522964904  0.002456802989787
 0.012310501160627 -0.000838905682578  0.001591334885603  0.006908391801689 -0.007050216276554
-0.000040472545662 -0.000253475629758 -0.000627479780824 -0.013422800841576  0.013165382618869
```

Calculation of phase diagram

Equilibrium condition

$$\mu_{Si}^{L} = \mu_{Si}^{S} \qquad \mu_{Ge}^{L} = \mu_{Ge}^{S}$$

$$\mu_{Si}^{L} - \mu_{Si}^{S} = {}^{o}G_{Si}^{dia\rightarrow liquid} + RT\ln x_{Si}^{L} - RT\ln x_{Si}^{S}$$

$$\mu_{Ge}^{L} - \mu_{Ge}^{S} = {}^{o}G_{Ge}^{dia\rightarrow liquid} + RT\ln x_{Ge}^{L} - RT\ln x_{Ge}^{S}$$

- 주어진 Gibbs energy 식을 이용하여 Ge-Si 2 원계 상태도를 계산으로 완성하시오.

$${}^{o}G_{Ge}^{dia\rightarrow liquid} = 36944.72 - 30.4975\ T$$

$${}^{o}G_{Si}^{dia\rightarrow liquid} = 50208.00 - 29.7617\ T$$

고상 (diamond 구조), 액상 모두에 대해 ideal solution 을 가정하시오.

For Si $\quad 50208.00 - 29.7617T + RT\ln x_{Si}^{L} - RT\ln x_{Si}^{S} = 0$

For Ge $\quad 36944.72 - 30.4975T + RT\ln(1 - x_{Si}^{L}) - RT\ln(1 - x_{Si}^{S}) = 0$

2 variables, 2 equations

Solve by Newton's method

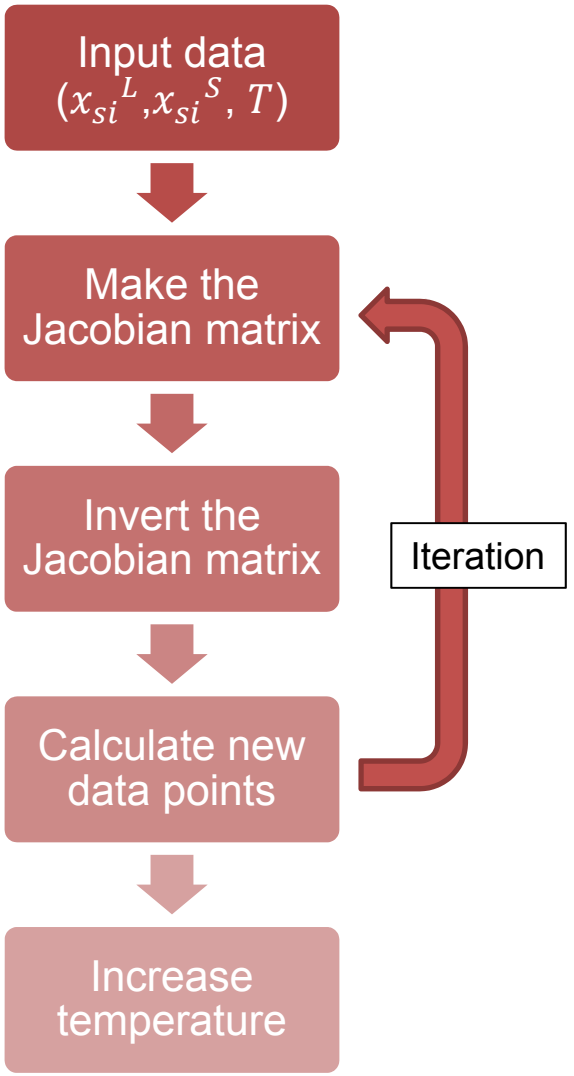
$$P_{(k)} = p_{(k-1)} - [J(P_{(k-1)})]^{-1} F(P_{(k-1)})$$

Calculation of phase diagram

$$P_{(k)} = p_{(k-1)} - [J(P_{(k-1)})]^{-1} F(P_{(k-1)})$$

```
for (int i=0; i<1000; i++){
    //Jacobian matrix 만들기
    jacob(jacobian, si_l, si_s, temp);
    inverse_matrix(jacobian, SIZE);
    f_si=func_si(si_l, si_s, temp);
    f_ge=func_ge(1-si_l, 1-si_s, temp);
    si_l_new= si_l - (jacobian[0][0]*f_si+jacobian[0][1]*f_ge);
    si_s_new= si_s - (jacobian[1][0]*f_si+jacobian[1][1]*f_ge);
    if (fabs((si_l_new - si_l))<0.000001){
        printf("%.0lf %.15lf %.15lf loop %d\n", temp, si_l, si_s,i);
        fp=fopen("result.txt", "a");
        fprintf(fp, "%.0lf %.15lf %.15lf %d\n", temp, si_l, si_s, i);
        fclose(fp);
        temp=temp+5;
        break;
    }else{

        si_l=si_l_new;
        si_s=si_s_new;
        if (si_l<0.0) {
            while(si_l<0.00){
                si_l=si_l+0.01;
            }
        }
        else{
            while(si_l>1.0){
                si_l=si_l-0.01;
            }
        }
        if (si_s<0.0) {
            while(si_s<0.00){
                si_s=si_s+0.01;
            }
        }
        else{
            while(si_s>1.0){
                si_s=si_s-0.01;
            }
        }
```



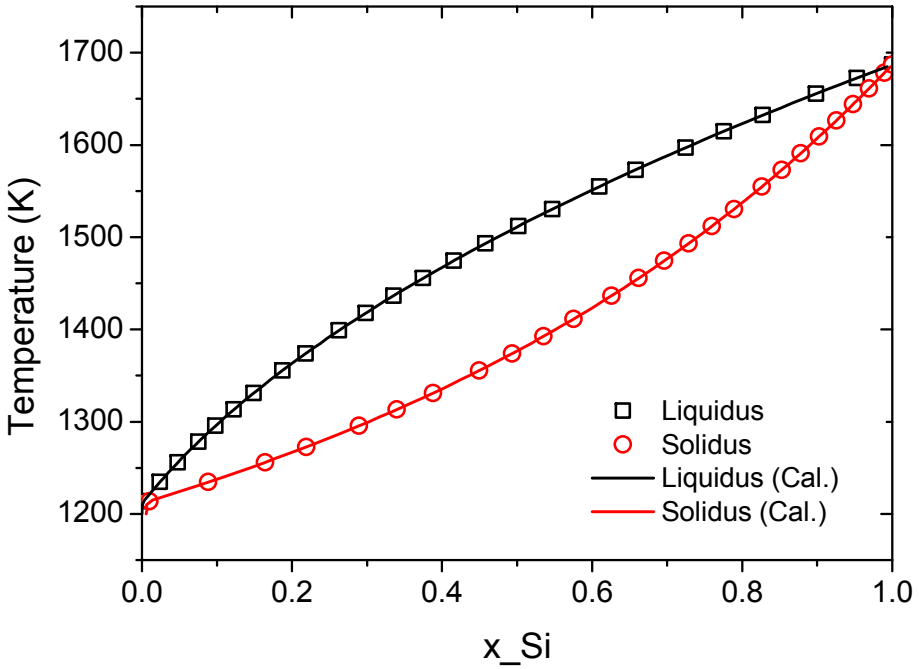
Results and discussion

```
Insert the initial points
0.2 0.9 1200

Temp.    x_si_liquid       x_si_solid    # of iteration
-------------------------------------------------------
1200  0.000017670089731  0.009053439969917   196
1205  0.000003460905163  0.003961093222607   189
1210  0.000285534615084  0.006079322483548   6
1215  0.003568262562659  0.014334269035804   5
1220  0.008616682034077  0.033917119303882   2
1225  0.013771254575317  0.053121513270307   2
1230  0.019032089382746  0.071957861518625   2
1235  0.024399793356786  0.090436690871799   2
1240  0.029874937888161  0.108568138709554   2
```

```
Insert the initial points
0.2 0.9 1300

Temp.    x_si_liquid       x_si_solid    # of iteration
-------------------------------------------------------
1300  0.104149638942692  0.302300436886694   5
1305  0.111068303331835  0.316695499165834   2
1310  0.118101643456483  0.330854778491840   2
1315  0.125249802931887  0.344783575417029   2
1320  0.132513207570516  0.358487815542776   2
1325  0.139892275208970  0.371973238501044   2
1330  0.147387415750456  0.385245404772006   2
1335  0.154999031206359  0.398309702214752   2
1340  0.162727515738598  0.411171352327023   2
```



- Large iteration number if x is close to 0
- After several increase in temperature, results were obtained within 2 iteration process