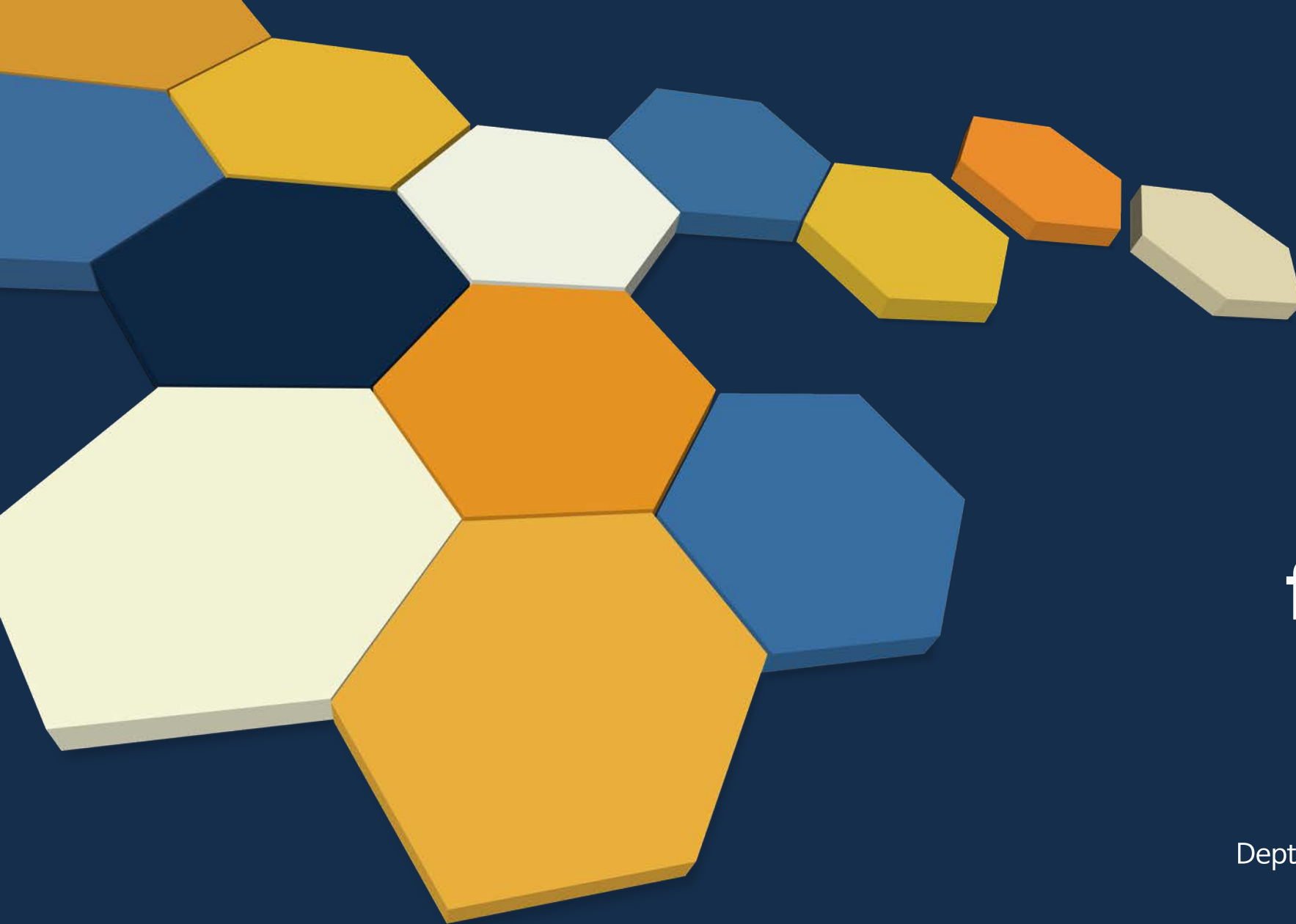




Numerical Analysis for Materials

Midterm

20130528
Dept. of Materials Science and Engineering
Cheolhee Han

Thermodynamics(L Parameter, Chemical Potential)



$$\Delta G_{ex} = X_{Tk}(1 - X_{Tk})\{L_0 + (1 - 2X_{Tk})L_1\} \quad \Delta G_{ex} = X_{Tk}(1 - X_{Tk})\{L_0 + (1 - 2X_{Tk})L_1\} = X_{Tk}(1 - X_{Tk})\Omega$$

T-dependent L parameters

- Liquid(1500K): $\Omega = (12496.767886 - 7.996993T) + (1 - 2X_{Tk})(2397.462734 + 5.616841 \times 10^{-3}T)$
- FCC(600K): $\Omega = (8997.620500 - 4.994669T) + (1 - 2X_{Tk})(3599.988962 - 6.622663 \times 10^{-3}T)$
- BCC(600K): $\Omega = (7000.661574 - 4.057468T)$

$$\mu_{Tk}^L = 6000 - 10T + RT \ln x_{Tk} + (1 - x_{Tk})^2 \{L_0^L + (1 - 4x_{Tk})L_1^L\}$$

$$\mu_{Tk}^{Fcc} = 7500 + RT \ln x_{Tk} + (1 - x_{Tk})^2 \{L_0^{Fcc} + (1 - 4x_{Tk})L_1^{Fcc}\}$$

$$\mu_{Tk}^{Bcc} = RT \ln x_{Tk} + (1 - x_{Tk})^2 \{L_0^{Bcc} + (1 - 4x_{Tk})L_1^{Bcc}\}$$

$$\mu_{Ps}^L = 12000 - 10T + RT \ln(1 - x_{Tk}) + (x_{Tk})^2 \{L_0^L + (3 - 4x_{Tk})L_1^L\}$$

$$\mu_{Ps}^{Fcc} = RT \ln(1 - x_{Tk}) + (x_{Tk})^2 \{L_0^{Fcc} + (3 - 4x_{Tk})L_1^{Fcc}\}$$

$$\mu_{Ps}^{Bcc} = 4000 + RT \ln(1 - x_{Tk}) + (x_{Tk})^2 \{L_0^{Bcc} + (3 - 4x_{Tk}) * L_1^{Bcc}\}$$

Strategy



1. Raw Phase Diagram

$$\text{Liquid - FCC } \mu_{Ps}^L = \mu_{Ps}^F \quad \mu_{TK}^L = \mu_{TK}^F$$

$$\text{Liquid - BCC } \mu_{Ps}^L = \mu_{Ps}^B \quad \mu_{TK}^L = \mu_{TK}^B$$

$$\text{FCC - BCC } \mu_{Ps}^F = \mu_{Ps}^B \quad \mu_{TK}^F = \mu_{TK}^B$$

Jacobian matrix $J(X)$

$$J(X) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

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

2. Peritectic Condition

$$\mu_{Ps}^{Liq} = \mu_{Ps}^{FCC} = \mu_{Ps}^{BCC}$$

$$\mu_{Ks}^{Liq} = \mu_{Ks}^{FCC} = \mu_{Ks}^{BCC}$$

$$f_1 = \mu_{Ps}^{Liq} - \mu_{Ps}^{FCC} = 0$$

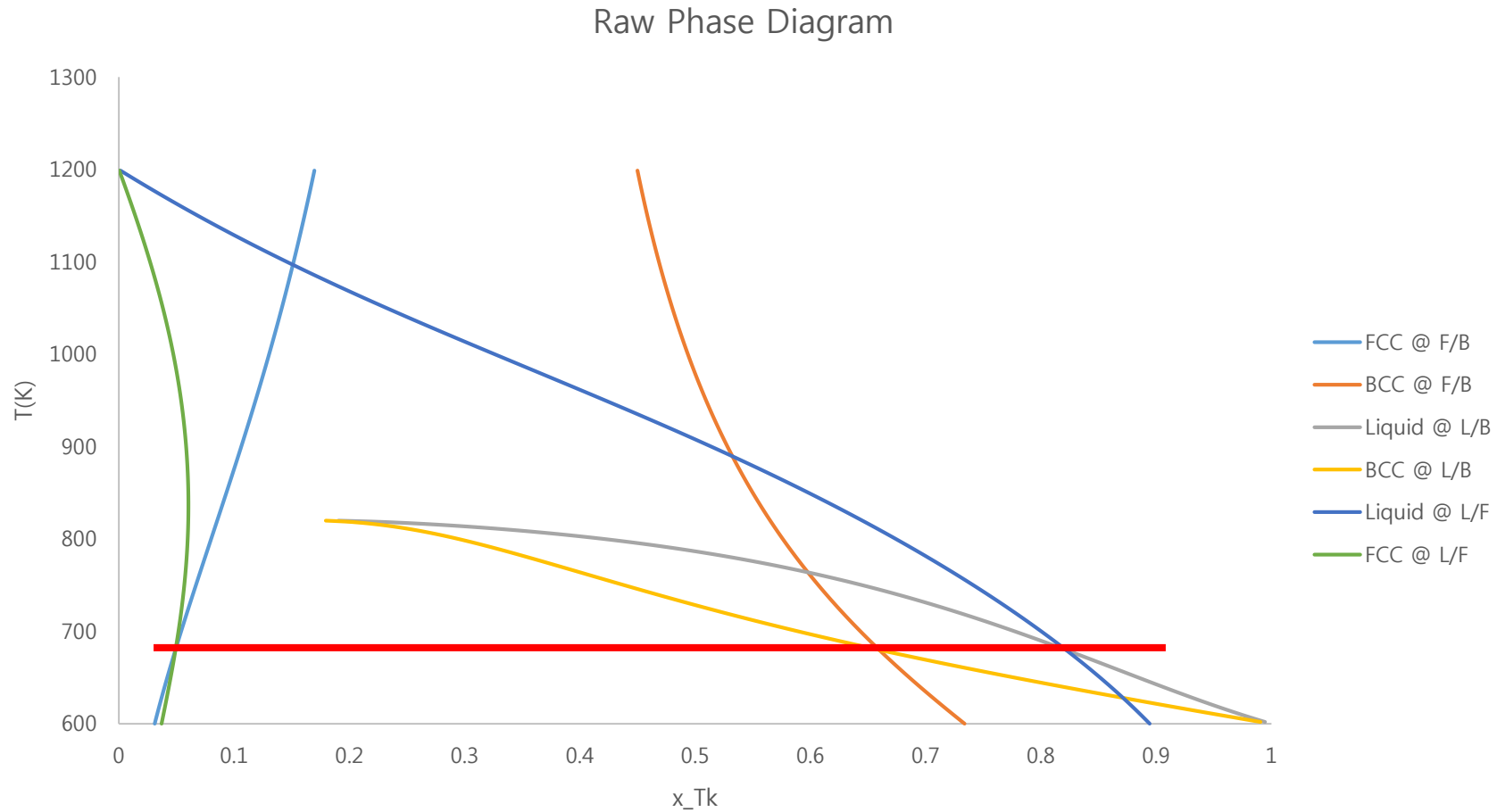
$$f_2 = \mu_{Ps}^{Liq} - \mu_{Ps}^{BCC} = 0$$

$$f_3 = \mu_{TK}^{Liq} - \mu_{TK}^{FCC} = 0$$

$$f_4 = \mu_{TK}^{Liq} - \mu_{TK}^{BCC} = 0$$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial T} & \frac{\partial f_1}{\partial X_{Ks}^{Liq}} & \frac{\partial f_1}{\partial X_{Ks}^{FCC}} & \frac{\partial f_1}{\partial X_{Ks}^{BCC}} \\ \frac{\partial f_2}{\partial T} & \frac{\partial f_2}{\partial X_{Ks}^{Liq}} & \frac{\partial f_2}{\partial X_{Ks}^{FCC}} & \frac{\partial f_2}{\partial X_{Ks}^{BCC}} \\ \frac{\partial f_3}{\partial T} & \frac{\partial f_3}{\partial X_{Ks}^{Liq}} & \frac{\partial f_3}{\partial X_{Ks}^{FCC}} & \frac{\partial f_3}{\partial X_{Ks}^{BCC}} \\ \frac{\partial f_4}{\partial T} & \frac{\partial f_4}{\partial X_{Ks}^{Liq}} & \frac{\partial f_4}{\partial X_{Ks}^{FCC}} & \frac{\partial f_4}{\partial X_{Ks}^{BCC}} \end{bmatrix}$$

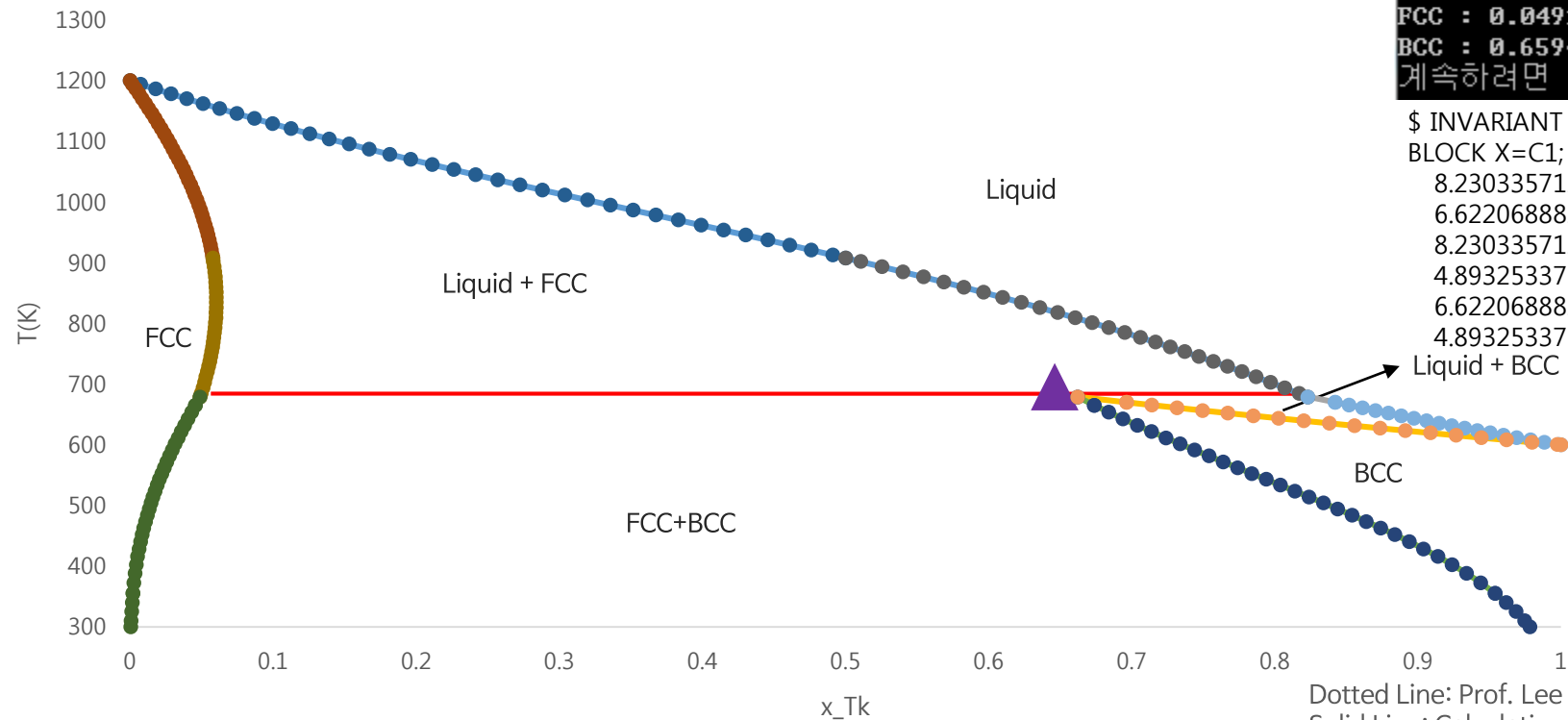
Raw Phase Diagram



Peritectic Condition



Ps-Tk Phase Diagram



Peritectic Condition

T : 680.083304

Liquid : 0.821584

FCC : 0.049136

BCC : 0.659490

계속하려면 아무 키나 누르십시오 . . .

\$ INVARIANT EQUILIBRIUM

BLOCK X=C1; Y=C2; GOC=C3,WAD;

8.2303357124E-01 6.7871173096E+02 M

6.6220688820E-01 6.7871173096E+02

8.2303357124E-01 6.7871173096E+02 M

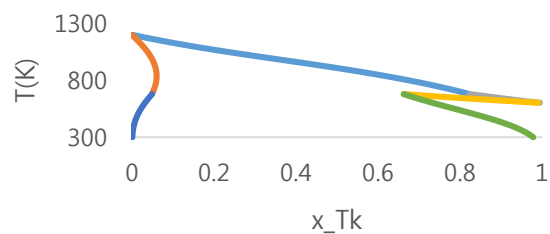
4.8932533711E-02 6.7871173096E+02

6.6220688820E-01 6.7871173096E+02 M

4.8932533711E-02 6.7871173096E+02

Liquid + BCC

Dotted Line: Prof. Lee
 Solid Line: Calculation
 Triangle: Peritectic Point



Investigation & Conclusion

Answers

1. Regression으로 구한 L parameter를 이용해 phase 별로 적합한 model을 적용, chemical potential을 계산할 수 있다.
2. Newton's Method를 이용하여 chemical potential로부터 raw phase diagram을 구성하는 data를 얻고 plot할 수 있다.
3. Newton's Method를 이용하여 Peritectic point(T, x_Tk)를 찾고 phase diagram을 완성할 수 있다.
4.
$$\text{Relative Error}(T) = \frac{680.083304K - 678.711731K}{678.711731K} \times 100(\%) = 0.202085\%$$
5. Peritectic point를 찾을 때에는 초기 분율과 온도 설정이 매우 중요하다.

```
//Functions for Thermodynamics
double L(char phase, int order, double T){ ... }
double Gibbs(char element, char phase, double T){ ... }
double chemical_potential(char element, char phase, double x, double T){ ... }
double derivative(char element, char phase, double x, double T){ ... }
double potential_difference(char element, char phase1, char phase2, double x1, double x2, double T){ ... }
double derivative_difference(char element, char phase1, char phase2, double x1, double x2, double T){ ... }

//Functions for Inverse Matrix(Jacobian)
void swap(double* a, double* b){ ... }
int max_row(double** A, int dim, int row){ ... }
int indet_chk(double** A, int dim, int pivot){ ... }
int pivoting(double** A, double** I, int dim, int pivot){ ... }
int Gauss_El(double** A, double** I, int dim){ ... }
int main()
```

```
printf("Phase Selection:\nChoose the Phase.\n");
printf("Liquid: l, FCC: f, BCC: b.\n");
printf("Example: l, f.\n");
scanf("%c", &c);
fflush(stdin);
x1 = 0.5;
x2 = 0.5;
T = 700;
for (T = T_min; T < T_max; T++)
{
    for (iter = 0; iter<1000; iter++)
    {
        if (x1 == x2)
        {
            x1 += TOL;
            x2 -= TOL;
        }
        u1 = potential_difference('K', phase1, phase2, x1, x2, T);
        u2 = potential_difference('P', phase1, phase2, x1, x2, T);
        u3 = potential_difference('K', phase1, phase2, x1, x2, T);
        u4 = potential_difference('P', phase1, phase2, x1, x2, T);
        //A_11q derivative
        A[0][0] = T_derivative('P', phase1, x1, T) - T_derivative('P', phase2, x2, T);
        A[1][0] = T_derivative('P', phase1, x1, T) - T_derivative('P', phase2, x3, T);
        A[2][0] = T_derivative('K', phase1, x1, T) - T_derivative('K', phase2, x2, T);
        A[3][0] = T_derivative('K', phase1, x1, T) - T_derivative('K', phase2, x3, T);
        //A_11q derivative
        A[0][1] = derivative('P', phase1, x1, T);
        A[1][1] = derivative('P', phase1, x1, T);
        A[2][1] = derivative('K', phase1, x1, T);
        A[3][1] = derivative('K', phase1, x1, T);
        //A_11q derivative
        A[0][2] = -derivative('P', phase2, x2, T);
        A[1][2] = 0;
        A[2][2] = -derivative('K', phase2, x2, T);
        A[3][2] = 0;
        //A_BCC derivative
        A[0][3] = 0;
        A[1][3] = -derivative('P', phase3, x3, T);
        A[2][3] = 0;
        A[3][3] = -derivative('K', phase3, x3, T);
        for (i = 0; i<dim; i++)
        {
            printf("No inverse\n");
        }
    }
}
```

```
phase1 = 'l';
phase2 = 'f';
phase3 = 'b';
x1 = 0.66;
x2 = 0.66;
x3 = 0.8;
T = 700;
for (iter = 0; iter<100; iter++)
{
    u1 = potential_difference('K', phase1, phase2, x1, x2, T);
    u2 = potential_difference('K', phase1, phase2, x1, x2, T);
    u3 = potential_difference('P', phase1, phase2, x1, x2, T);
    u4 = potential_difference('P', phase1, phase2, x1, x2, T);
    //derivative
    A[0][0] = T_derivative('P', phase1, x1, T) - T_derivative('P', phase2, x2, T);
    A[1][0] = T_derivative('P', phase1, x1, T) - T_derivative('P', phase2, x3, T);
    A[2][0] = T_derivative('K', phase1, x1, T) - T_derivative('K', phase2, x2, T);
    A[3][0] = T_derivative('K', phase1, x1, T) - T_derivative('K', phase2, x3, T);
    //A_11q derivative
    A[0][1] = derivative('P', phase1, x1, T);
    A[1][1] = derivative('P', phase1, x1, T);
    A[2][1] = derivative('K', phase1, x1, T);
    A[3][1] = derivative('K', phase1, x1, T);
    //A_11q derivative
    A[0][2] = -derivative('P', phase2, x2, T);
    A[1][2] = 0;
    A[2][2] = -derivative('K', phase2, x2, T);
    A[3][2] = 0;
    //A_BCC derivative
    A[0][3] = 0;
    A[1][3] = -derivative('P', phase3, x3, T);
    A[2][3] = 0;
    A[3][3] = -derivative('K', phase3, x3, T);
    for (i = 0; i<dim; i++)
    {
        printf("No inverse\n");
    }
}
```

Phase Selection:
Choose the Phase.
Liquid: l, FCC: f, BCC: b
Example: l, f
l, f