NUMERICAL ANALYSIS

6

조 혜 성

1.주어진 데이터를 이용해 Phase Diagram 그리기

$$\mu_{Tk}{}^{L} = \Delta G_{Tk}{}^{Bcc\to L} + RTlnx_{Tk}^{L} + (1 - x_{Tk}^{L})^2 \{L[0] + L[1]T + \left(1 - 4x_{Tk}^{L}\right) * (L[2] + L[3]T)\}$$

$$\mu_{Tk}{}^{Fcc} = \Delta G_{Tk}{}^{Bcc\to Fcc} + RTlnx_{Tk}^{Fcc} + (1 - x_{Tk}^{Fcc})^2 \{FCC[0] + FCC[1]T + \left(1 - 4x_{Tk}^{Fcc}\right) * (FCC[2] + FCC[3]T)\}$$

$$\mu_{Tk}{}^{Bcc} = RTlnx_{Tk}^{Bcc} + (1 - x_{Tk}^{Bcc})^2 \{BCC[0] + BCC[1]T\}$$

$$\mu_{Ps}{}^{L} = \Delta G_{Ps}{}^{Fcc\to L} + RTln(1 - x_{Tk}^{L}) + \left(x_{Tk}^{L}\right)^2 \{L[0] + L[1]T + \left(3 - 4x_{Tk}^{L}\right) * (L[2] + L[3]T)\}$$

$$\mu_{Ps}{}^{Fcc} = RTln(1 - x_{Tk}^{Fcc}) + \left(x_{Tk}^{Fcc}\right)^2 \{FCC[0] + FCC[1]T + \left(3 - 4x_{Tk}^{Fcc}\right) * (FCC[2] + FCC[3]T)\}$$

$$\mu_{Ps}{}^{Bcc} = \Delta G_{Ps}{}^{Fcc\to Bcc} + RTln(1 - x_{Tk}^{Bcc}) + \left(x_{Tk}^{Bcc}\right)^2 \{BCC[0] + BCC[1]T\}$$

$$\begin{bmatrix} x_{Tk}^{L}{}_{\text{new}} \\ x_{Tk}^{FCC}{}_{\text{new}} \\ x_{Tk}^{BCC}{}_{\text{new}} \\ T_{\text{new}} \end{bmatrix} = \begin{bmatrix} x_{Tk}^{L} \\ x_{Tk}^{FCC} \\ x_{Tk}^{BCC} \\ T \end{bmatrix} - \begin{bmatrix} \frac{\partial f_{Tk}FCCtoLiq}{\partial x_{Tk}^{L}} & \frac{\partial f_{Tk}FCCtoLiq}{\partial x_{Tk}^{FCC}} & \frac{\partial f_{Tk}FCCtoLiq}{\partial x_{Tk}^{BCC}} & \frac{\partial f_{Tk}FCCtoLiq}{\partial T} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} f_{Tk}FCCtoLiq \\ f_{Ps}FCCtoLiq \\ f_{Tk}BCCtoLiq \\ f_{Ps}BCCtoLiq \end{bmatrix}$$

2.Algorithms

$$f_x AtoB = \mu_x^B - \mu_x^A$$

$$f_{Tk}FCCtoLiq, f_{Ps}FCCtoLiq, f_{Tk}BCCtoLiq, f_{Ps}BCCtoLiq, f_{Tk}BCCtoFCC, f_{Ps}BCCtoFCC$$

Newton's Method

Matrix inversion

범위 Invariant equilibrium [T, T + 700], [T, T - 700]
If Iteration > 998: break

Plot

3. Code

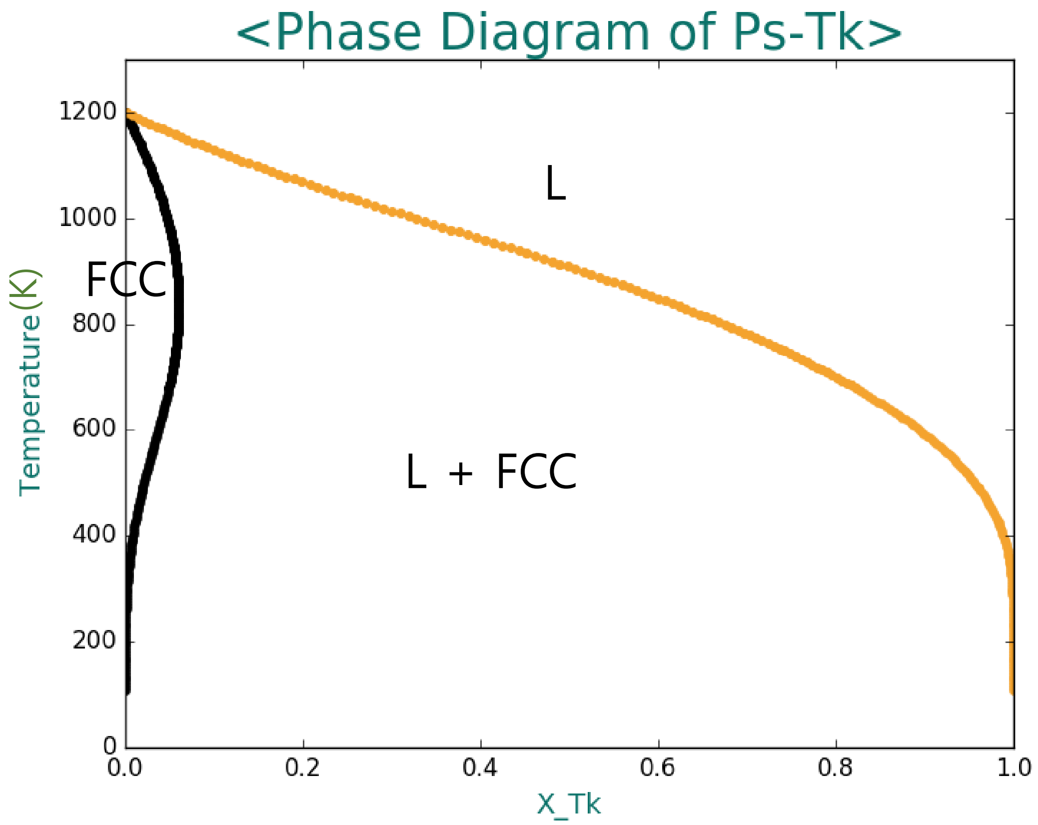
```
def fTkFCCtoLiq(L, FCC, T, xl, xfcc):
    G = 6000 - 10 * T
    fTk = G - GBF + R*T*(log(xl) - log(xfcc)) +(L[0]+L[1]*T +(1-4*x
    return fTk
def fPsFCCtoLiq(L, FCC, T, xl, xfcc):
def fTkBCCtoLiq(L, BCC, T, x1, x2):
def fPsBCCtoLiq(L, BCC, T, x1, x2):
def fTkBCCtoFCC(BCC, FCC, T, x2, x3):
def fPsBCCtoFCC(BCC, FCC, T, x2, x3):
def makeJinv(L, B, T, x1, x2, x3, x4, k):
def Newtons(T1, T2, a, s):
```

```
for i in range(0, 999, 1):
    if s == 0:
        Jinv = makeJinv(L, FCC, T,
        x1_new = x1 - (Jinv[0][0]
        x2_new = x2 - (Jinv[1][0]
    elif s == 1:
        Jinv = makeJinv(L, BCC, T,
        x1_new = x1 - (Jinv[0][0]
        x2_new = x2 - (Jinv[1][0]
    elif s == 2:
        Jinv = makeJinv(FCC, BCC,
        x1_new = x1 - (Jinv[0][0]
        x2_new = x2 - (Jinv[1][0]
    elif s == 3:
        Jinv = makeJinv(FCC, BCC,
        x1_new = x1 - (Jinv[0][0]
        x2_new = x2 - (Jinv[1][0]
        x3_new = x3 - (Jinv[2][0]
        x4_new = x4 - (Jinv[3][0]
```

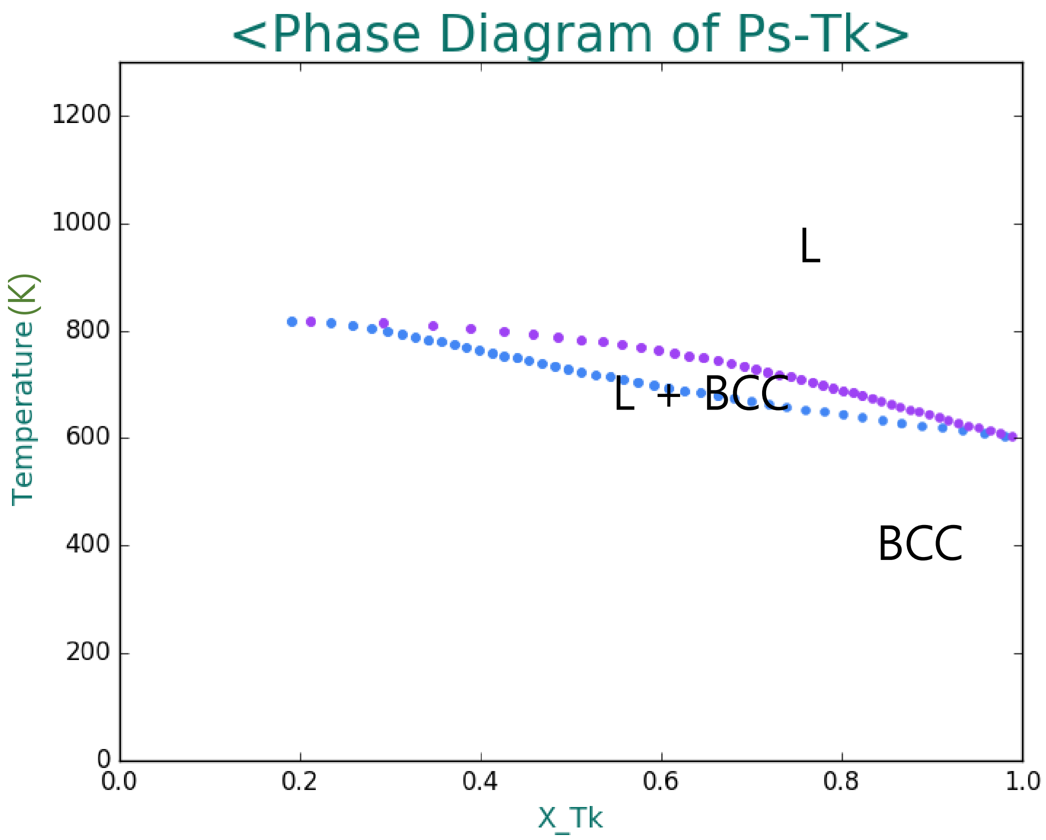
```
XTk_Liq: 0.8224  XTk_FCC: 0.0490      XTk_BCC: 0.6610      Temperature: 679.3177   i: 12
```

4. Result

$f_{Tk}FCCtoLiq,\ f_{Ps}FCCtoLiq$

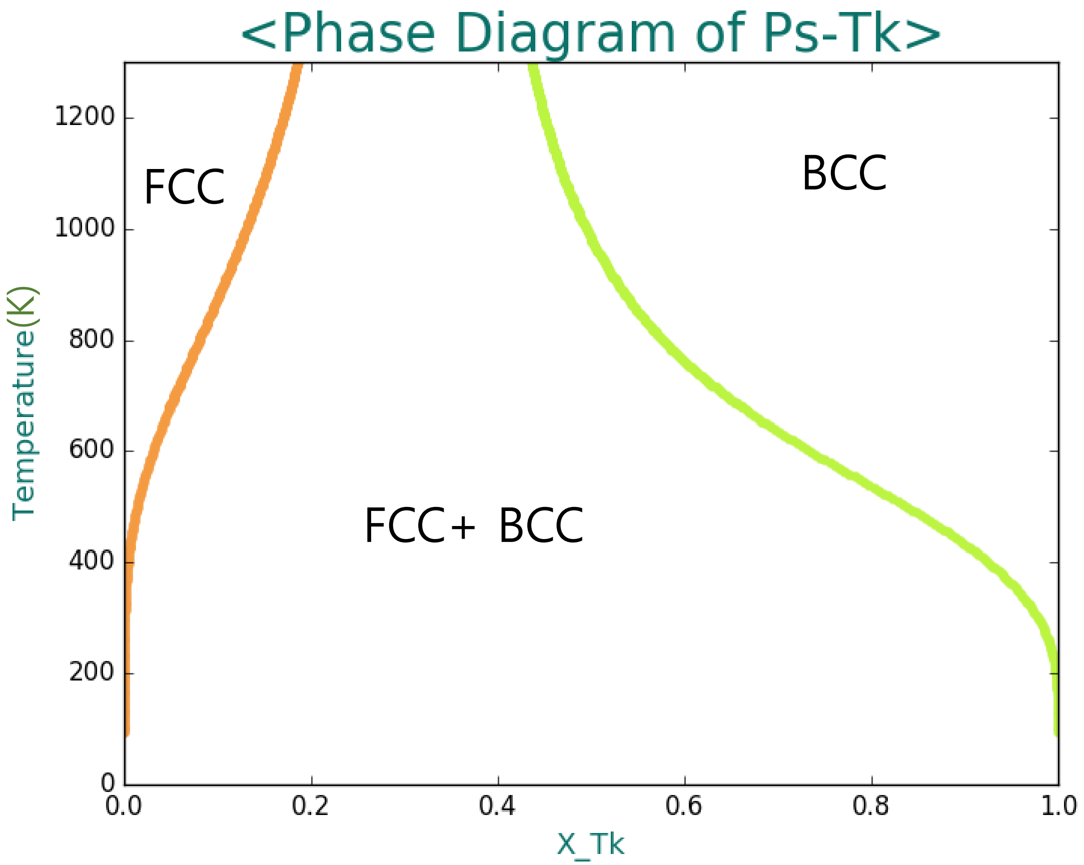


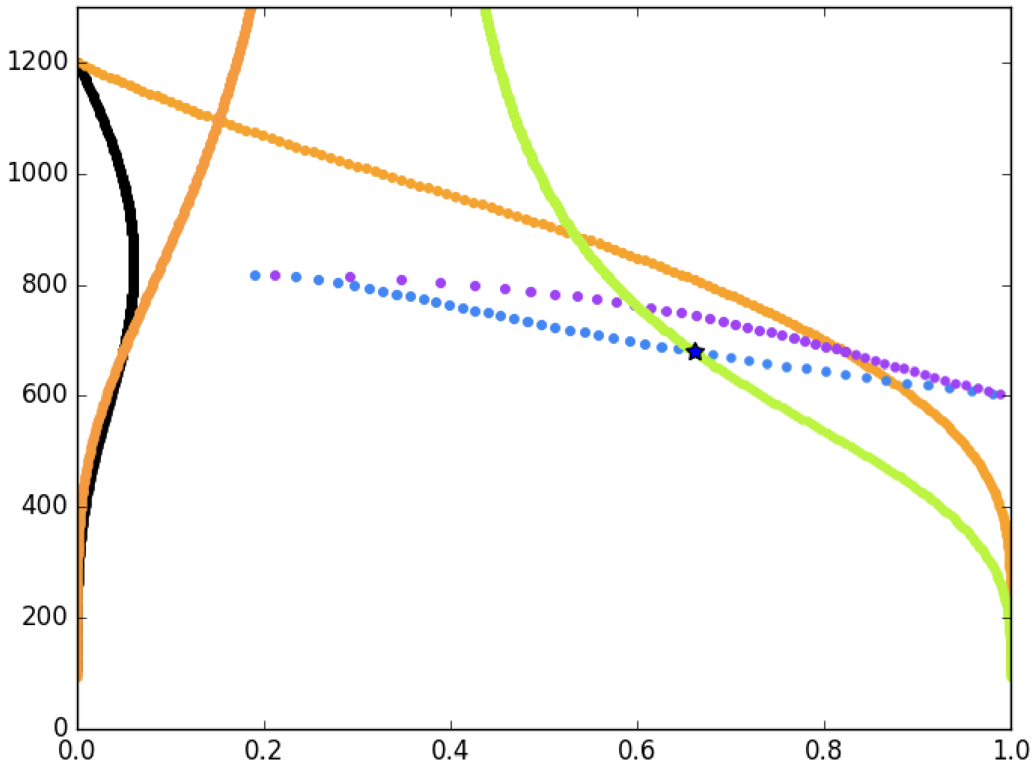
$f_{Tk}BCCtoLiq,\ f_{Ps}BCCtoLiq$



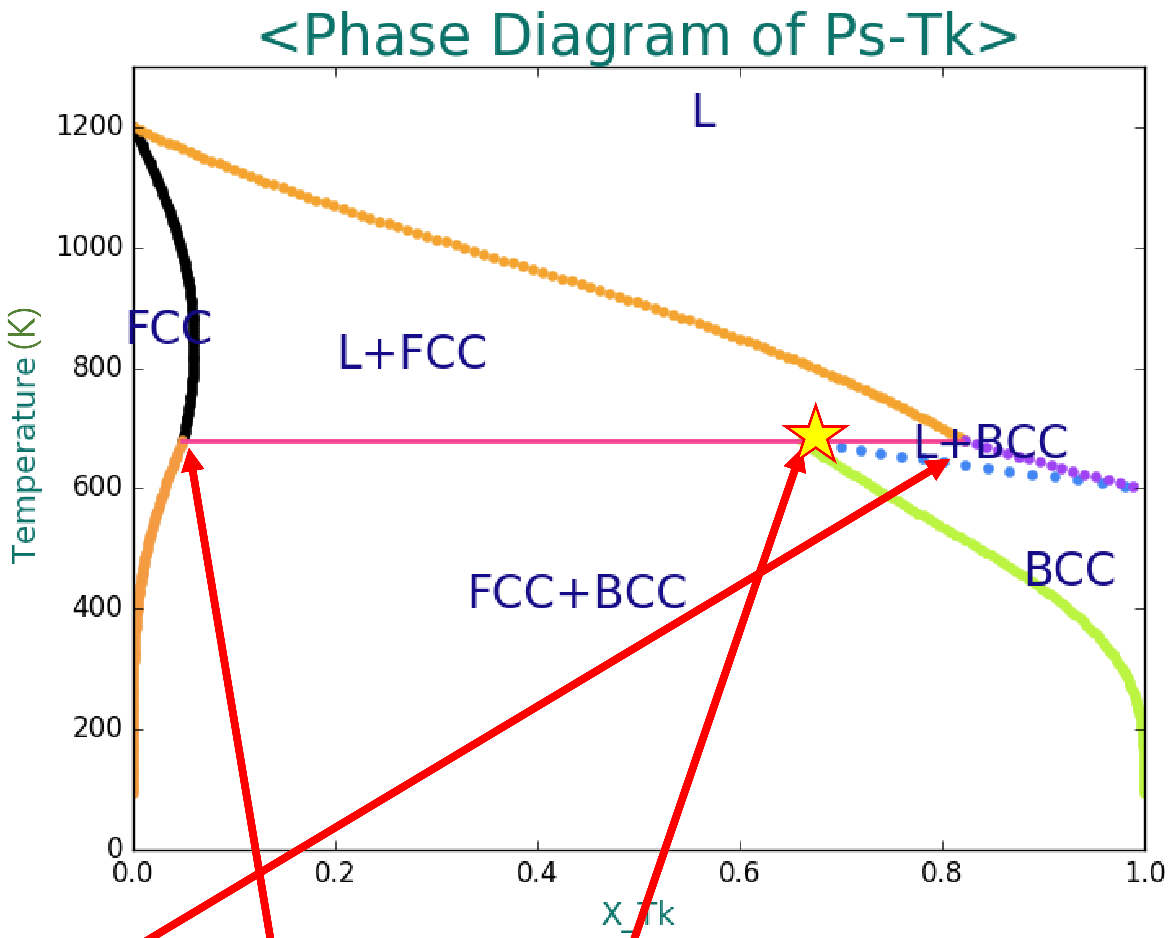
4. Result

$$f_{Tk}BCCtoFCC, f_{Ps}BCCtoFCC$$





4. Result

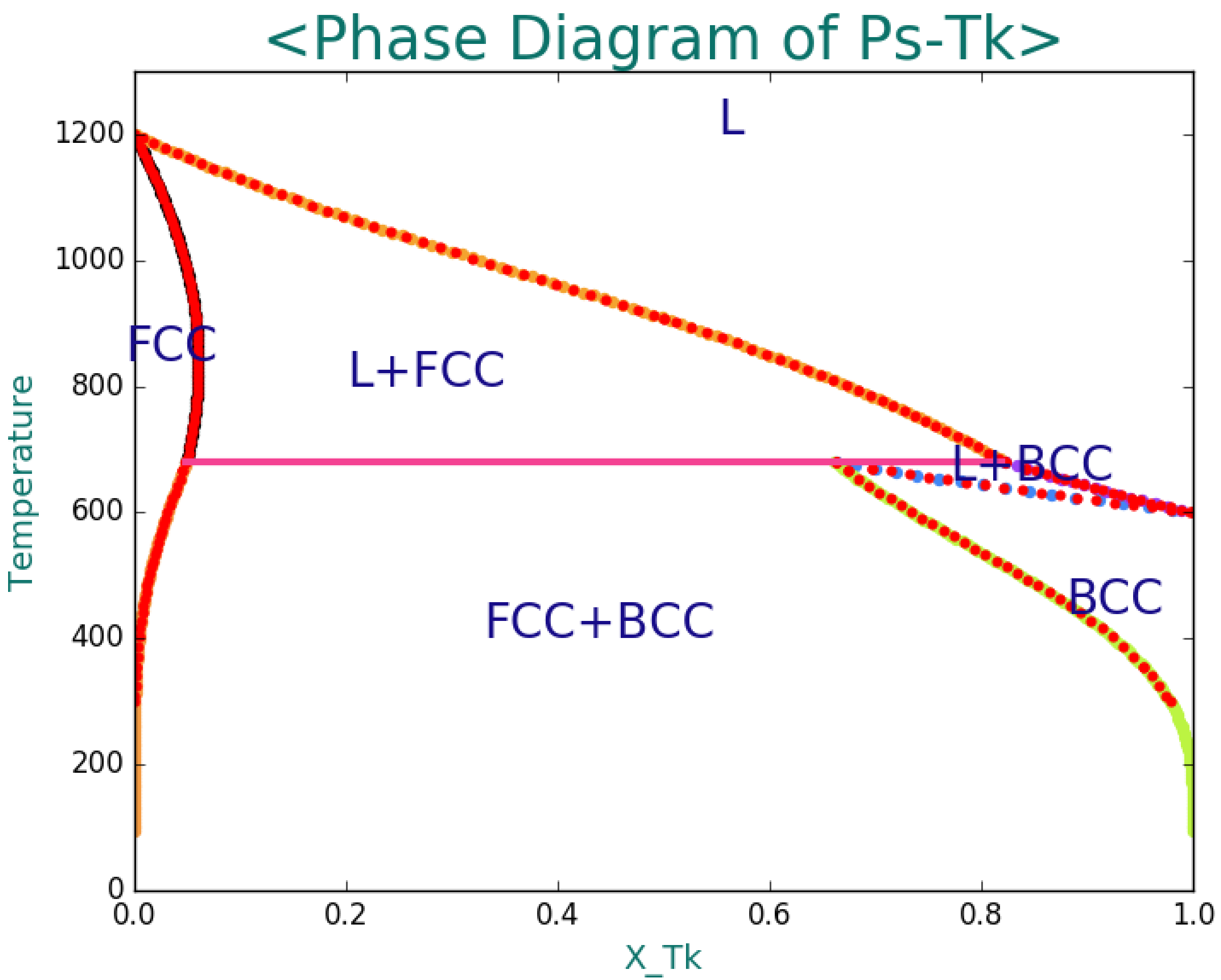


Peritectic point at
XTk_BCC= 0.66
T=679.3177 K

```
XTk_Liq: 0.8224  XTk_FCC: 0.0490     XTk_BCC: 0.6610     Temperature: 679.3177   i: 12
```

4. Result

Red dots =
주어진 data



5. 마지막 장입니다.

지금까지의 과제 중에 가장 프로그램을 돌리는데 오래 걸려서 기다리는 시간이 많았다

Binary Phase diagram에서는 각 성분의 녹는점을 기준으로 알고리즘을 돌렸는데,

이번 과제는 모든 상의 녹는점을 알 수 없었음

→ 모든 상이 존재하는 Peritectic Point를 기준, 위 아래로 데이터를 구함으로 해결

Plot하기 전에 Eutectic, Peritectic을 알 수는 없을까?

→XTk_FCC < XTk_Liq < XTk_BCC 이면 Eutectic

→XTk_FCC < XTk_BCC < XTk_Liq 이면 Peritectic으로 판단

이제 다음 과제 하러 가볼까요?ㅎ.ㅎ