

**PMT2512 - Laboratório de Termodinâmica Computacional - Exercício 1 -
29/02/2008**

Cálculo do Diagrama de Fases do Sistema Zr-Nb

1 Introdução

O programa de cálculos termodinâmicos que será utilizado no presente curso chama-se THERMO-CALC. Ele corresponde a um dos primeiros programas desenvolvidos para esta finalidade e foi criado no Departamento de Metalurgia Física do Instituto Real de Tecnologia de Estocolmo, Suécia. O programa tem uma estrutura modular, o que significa dizer que ele consiste de blocos com diferentes funções. Existe uma versão gráfica do programa, que entretanto não será utilizada no presente curso. Ao contrário, trabalharemos com a versão em linha de comando.

Ao se iniciarmos o programa iremos obter uma janela contendo os seguintes dizeres:

```
Thermo-calc version P on WinNT  
Copyright (1993,2002), Foundation for Computational Thermodynamics  
Stockholm, Sweden  
Double precision version linked ---
```

SYS:

O prompt do programa aceita agora instruções na forma de comandos em linha. Uma lista de comandos disponíveis no módulo pode ser obtida digitando-se:

```
SYS:? <Enter>
```

O módulo inicial é o módulo de controle, que tem uma funcionalidade limitada para os objetivos do presente exercício. Ele serve, basicamente, para permitir a comunicação entre todos os módulos. Desejamos prosseguir com o cálculo e para tanto devemos mudar de módulo. O comando que apresenta esta funcionalidade é **GOTO_MODULE** que tem a sintaxe

GOTO_MODULE *nome do módulo*

A linha de comando do Thermo-calc, entretanto, é sensível ao contexto: basta digitarmos o número de letras do nome do comando suficiente para permitir ao programam distinguir entre os diferentes comandos disponíveis, sendo assim, os dois comandos:

```
SYS:GOTO_MODULE <Enter>
```

e

```
SYS:go <Enter>
```

tem o mesmo efeito.

O comando também aceita a versão completa, ou seja, contendo todos os argumentos necessários ao comando, ou incompleta, onde os argumentos são dados sequencialmente pelo usuário, de forma interativa.

Por exemplo, digite o seguinte:

```
SYS:go <Enter>
```

O programa, logo em seguida, apresentará um prompt contendo:

```
SYS:go  
MODULE_NAME:
```

Se você, em seguida, digitar:

```
SYS:go  
MODULE_NAME: <Enter>
```

obterá uma lista de todos os módulos disponíveis (teste!).

Desejamos ir para o módulo **DATABASE_RETRIEVAL**, para tanto:

```
SYS:go  
MODULE_NAME: data <Enter>
```

ou simplesmente

```
SYS:go data <Enter>
```

O prompt em seguida irá mudar para:

```
TDB_SSOL:
```

Indicando que mudamos de módulo.

2 Carregando os dados do sistema

No módulo DATABASE_RETRIEVAL iremos definir o sistema em que estamos interessados e carregar os dados presentes na base de dados. Os comandos disponíveis neste módulo pode ser obtidos por:

```
TDB_SSOL:? <Enter>
```

Ao entrarmos no módulo definimos automaticamente a base de dados SGTE Solid Solution (SSOL) como padrão para os cálculos. Esta não é a única base de dados disponível no programa, porém iremos tratar deste assunto em outra aula. No momento iremos continuar trabalhando com a base SSOL.

Devemos agora definir o nosso sistema termodinâmico. Para tanto usamos o comando **DEFINE_SYSTEM**:

```
TDB_SSOL: def-sys Zr Nb <Enter>
```

Como resultado iremos obter:

```
TDB_SSOL: def-sys Zr Nb
ZR                      NB DEFINED
TDB_SSOL:
```

Iremos inicialmente escolher as fases que compõem o sistema. Primeiramente limpamos a lista de fases com o comando **REJECT**:

```
TDB_SSOL: rej ph * <Enter>
```

e em seguida restauramos as fases desejadas com o comando **RESTORE**:

```
TDB_SSOL:res pha liq bcc fcc hcp <Enter>
```

e carregamos os dados termodinâmicos para esta fase com o comando **GET_DATA**:

```
TDB_SSOL:get <Enter>
```

O procedimento de eliminação de fases desnecessárias, feito acima, a rigor não é necessário no presente caso. Entretanto, sistemas mais complexos (ternários, quaternários) podem conter uma quantidade muito grande de fases, que eventualmente apenas atrapalham no cálculo. Desta forma é sempre uma boa política introduzir no cálculo apenas as fases que apresentam alguma chance de aparecer no cálculo (ou que são suficientemente simples para não atrapalhar o cálculo, como por exemplo a fase fcc, que foi incluída acima).

Você irá notar que o programa apresenta uma lista de referências, que são aquelas que forma usadas para montar a base de dados para o sistema em questão. Note que o artigo estudado em aula está listado.

3 Calculando o diagrama de fases

Com os dados do sistema carregados, podemos agora iniciar os cálculos, para tanto iremos para o módulo POLY-3 que realiza os cálculos do equilíbrio termodinâmico.

```
TDB_SSOL:go poly <Enter>
```

Como resultado teremos:

```
TDB_SSOL:go poly
```

```
POLY Version 3.22  
POLY_3:
```

Inicialmente iremos definir o estado termodinâmico do sistema. Para tanto precisamos definir a temperatura T , a pressão P , o número de moles de que o sistema é composto n e uma variável de composição, por exemplo a fração em massa de nióbio no sistema, $w(Nb)$. O comando que permite fixar as variáveis de estado do sistema é **SET_CONDITION**:

```
POLY_3:s-con t=2200 p=101325 n=1 w(nb)=0.25
```

Com isto fixamos as variáveis:

$$\begin{aligned} T &= && 2200 \text{ K} \\ P &= && 101325 \text{ MPa (1 atm)} \\ n &= && 1 \text{ mol} \\ w(Nb) &= && 0.20 \end{aligned}$$

Verifique o estado termodinâmico do sistema com o comando **LIST_CONDITION**:

```
POLY_3:li-con <Enter>
```

O resultado indica que todas as variáveis de estado disponíveis foram fixadas (número de graus de liberdade = 0). Podemos agora calcular o equilíbrio correspondente com o comando **COMPUTE_EQUILIBRIUM**:

```
POLY_3:c-eq <Enter>
```

Repita o procedimento acima até que o número de iterações do cálculo se estabilize. A seguir liste o equilíbrio obtido com o comando **LIST_EQUILIBRIUM**:

```
POLY_3:l-eq <Enter>  
Output file: /TERMINAL/: <Enter>  
Options: /VWCS/:<Enter>
```

Observe o resultado.

Para iniciar o cálculo do diagrama de fases necessitamos de uma condição inicial que contenha duas fases em equilíbrio (isto não é estritamente necessário, mas simplifica consideravelmente o cálculo). O resultado do cálculo anterior indicava, entretanto, apenas a fase "Liquid" como fase estável. Sabemos que o campo correspondente ao equilíbrio entre uma fase sólida (BCC) e a fase líquida se encontra em uma temperatura menor que 2200K, vamos, portanto, reduzir a temperatura:

```
POLY_3:s-con t=2000 <enter>
```

Utilize agora o comando **LIST_CONDITION** para verificar os valores das demais variáveis de estado:

```
POLY_3:l-con <enter>
T=2000, P=101325, n=1, w(Nb)=0.20
DEGREES OF FREEDOM 0
```

Portanto o novo comando **SET_CONDITION** manteve os demais valores inalterados. Calcule o novo equilíbrio:

```
POLY_3:c-eq <enter>
```

E verifique o equilíbrio obtido:

```
POLY_3:l-eq <enter>
```

...

Você irá verificar que o equilíbrio agora corresponde à fase sólida “BCC A2”, portanto o campo de duas fases desejado estará no intervalo de temperaturas entre 2000 e 2200K. Repita o procedimento anterior até determinar uma temperatura onde as duas fases estejam em equilíbrio.

Após determinar a condição de equilíbrio envolvendo duas fases, devemos estabelecer que tipo de diagrama desejamos calcular. No presente caso desejamos calcular um diagrama $T \times w_{Nb}$, para tanto devemos estabelecer os eixos coordenados do diagrama, usando o comando **SET_AXIS_VARIABLE**:

```
POLY_3:s-a-v
Axis number: /1/: <enter>
Condition: /NONE/: w(Nb) <enter>
Min value: /0/: <enter>
Max value: /1/: <enter>
Increment: /0.025/: 0.005 <enter>
```

Isto fixou a abcissa como representando a variável w_{Nb} , no intervalo $0 \leq w_{Nb} \leq 1$ com incremento $\Delta w_{Nb} = 0,005$. Para a ordenada teremos:

```
POLY_3:s-a-v 2 t 500 3000 15 <enter>
```

Isto fixou o eixo das ordenadas como representando a variável T , no intervalo $500K \leq T \leq 3000K$ com incremento $\Delta T = 15K$. Verifique usando o comando **LIST_AXIS_VARIABLE**.

Agora podemos adicionar o equilíbrio adicionando anteriormente como ponto inicial do cálculo, usando o comando **ADD_INITIAL_EQUILIBRIUM**:

```
POLY_3: add <enter>
Direction: /Default/: <enter>
```

E finalmente calculamos o diagrama de fases utilizando o comando **MAP**:

```
POLY_3: map <enter>
```

O programa irá agora executar uma série de cálculos mapeando as variáveis da abcissa e da ordenada, até atingir os limites estabelecidos previamente nos comandos **SET_AXIS_VARIABLE**.

Ao final resta-nos observar o gráfico, para tanto usamos o pós-processador **POST**:

```
POLY_3: post <enter>
```

```
POLY_3 POSTPROCESSOR VERSION 3.2 last update 1996-12-12
```

```
Setting automatic diagram axis
```

```
POST:
```

Note que o pós-processador já definiu os eixos do gráfico que será plotado (novos eixos, entretanto, podem ser definidos usando-se o comando **SET_DIAGRAM_AXIS** do pós-processador).

Caso o sistema usado seja Linux, é necessário ajusta o formato de impressão do gráfico para que ele apareça na tela (usando o comando **SET_PLOT_FORMAT**):

```
POST: s-p-f 9 <enter>
```

Se você estiver trabalhando em um sistema Windows isto será desnecessário.

A seguir podemos plotar o gráfico (com o comando **PLOT_DIAGRAM**):

```
POST: plot <enter>
```

Observamos que o gráfico apresenta apenas os equilíbrios envolvendo as fases "BCC_A2" e "Liquid". Isto é uma deficiência do Thermo-calc, ele não procura campos bifásicos desconexos por conta própria, desta forma é necessário que o operador tenha uma boa idéia de como será o diagrama de fases antes da realização do cálculo, para evitar perder algum campo de duas fases.

Vamos agora manipular um pouco o gráfico:

- Definindo um título (comando **SET_TITLE**)

```
POST:s-tit PMT2512 - Exercicio 1 <enter>
```

- Plotando uma *tie-line* a cada nove calculadas (comando **SET_TIELINE_STATUS**)

```
POST:s-tie-st 9 <enter>
```

- Finalmente imprimindo uma cópia para teste (novamente usando o comando **SET_PLOT_FORMAT**)

```
POST:s-p-f 16 <enter>
```

```
NEW DEVICE: Color Postscript portrait mode
```

```
USE POSTSCRIPT FONT: /Yes/ <enter>
```

```
SELECT FONTNUMBER: /9/ <enter>
```

```
FONT SIZE: /0.35833/ <enter>
```

O diagrama em questão encontra-se na figura 1:

Para dar continuidade ao cálculo retornamos ao módulo POLY_3 com o comando **BACK**:

```
POST:back <enter>
```

Nota-se que a linha de comando indica que agora estamos no módulo POLY_3 (o comando BACK só está definido no Postprocessor, portanto não adianta tentar usá-lo em outro módulo).

Como todos lembramos, o Zircônio possui dois alótropos estáveis: uma fase hexagonal de baixa temperatura e uma fase cúbica de corpo centrado de alta temperatura. Procuraremos agora o campo de duas fases “HCP_A3” + “BCC_A2” que deve existir para ligas ricas em Zr a baixas temperaturas. Selecionamos portanto novas condições iniciais:

```
POLY_3:s-con t=800 w(Nb)=0.1 <enter>
```

```
POLY_3:c-e <enter>
```

```
135 ITS, CPU TIME USED 0 SECONDS
```

```
POLY_3:c-e <enter>
```

```
60 ITS, CPU TIME USED 0 SECONDS
```

```
POLY_3:c-e <enter>
```

```
12 ITS, CPU TIME USED 0 SECONDS
```

```
POLY_3:c-e <enter>
```

```
12 ITS, CPU TIME USED 0 SECONDS
```

```
POLY_3:li-eq <enter>
```

```
Output file: /TERMINAL/: <enter>
```

```
Options: /VWCS/: <enter>
```

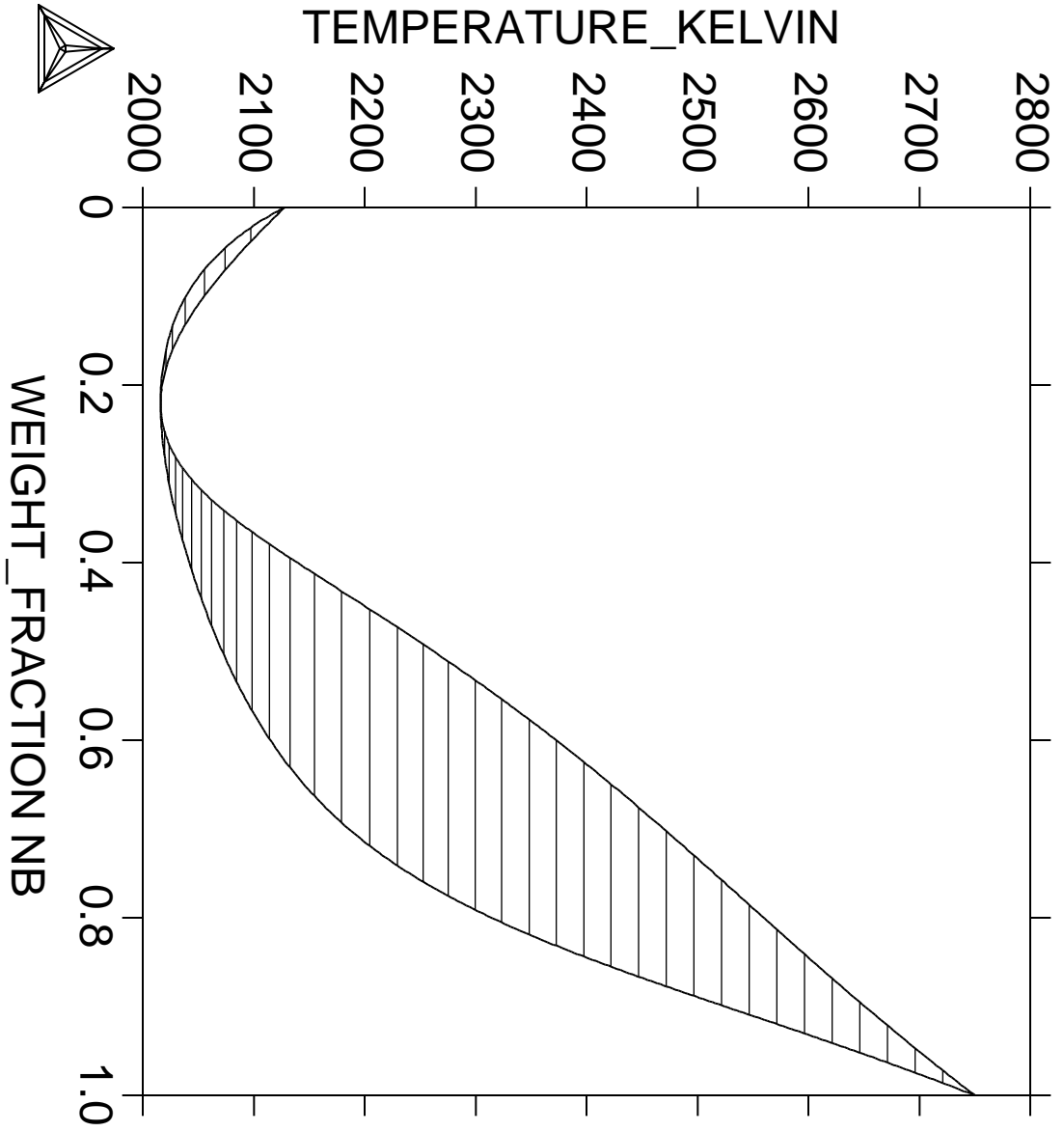
Vemos que conforme esperado encontramos o campo de duas fases entre “BCC_A2” e “HCP_A3”. A partir daí simplesmente adicionamos esta nova condição inicial e rodamos **MAP**:

```
POLY_3: add <enter>
```

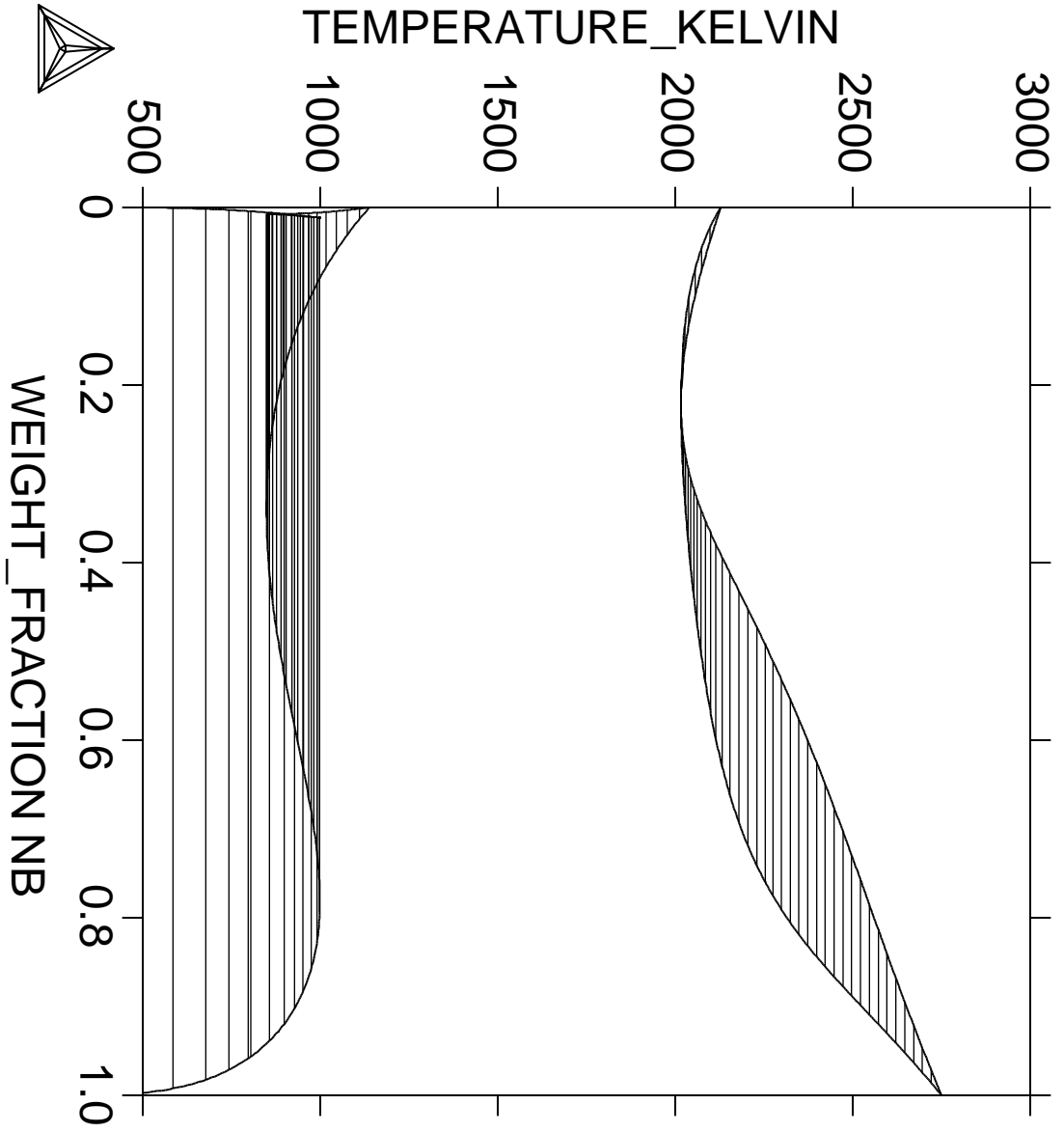
```
Direction /Default/: <enter>
```

```
POLY_3: map <enter>
```

THERMO-CALC (**.02.22:14.59) :PMT2512 - Exercício 1



THERMO-CALC (**.02.22:15.19) :PMT2512 Exercício 1



O programa calculará um novo conjunto de equilíbrios (correspondente à condição inicial especificada). Vamos plotar o diagrama obtido, repetindo o procedimento usado anteriormente. O resultado está apresentado na Figura 2.

Nota-se que o diagrama calculado não se assemelha ao diagrama experimental e ainda apresenta características muito inusitadas, como por exemplo *tie-lines* que cruzam o campo monofásico “BCC_A2”.

Isto ocorre porque este sistema apresenta um *domo de imiscibilidade* ou seja, um campo bifásico entre duas fases de mesma estrutura, mas com composições diferentes. A presença de um domo de imiscibilidade indica que a energia livre de Gibbs da fase “BCC_A2” possui uma segunda derivada negativa para alguns valores de composição para temperaturas abaixo do máximo do domo (ou seja, ela é um máximo ao invés de um mínimo do equilíbrio). Isto viola as leis da termodinâmica e por isto a solução sólida se separa espontaneamente em regiões ricas em zircônio e outras ricas em nióbio (na composição de equilíbrio de cada fase a segunda derivada de G é positiva e as leis da termodinâmica são preservadas).

O Thermo-calc não consegue identificar por conta própria a presença de domos de imiscibilidade¹ e o usuário deve especificá-los no início do cálculo.

Para tanto retornamos novamente ao módulo POLY_3 e, rodamos o comando **SPECIAL_OPTIONS**:

```
POLY_3:sp-op <enter>
Which option? /SET_MISCIBILITY_GAP/:
Phase with miscibility gap: bcc <enter>
New highest set number /2/:
Give for composition set 2
Major constituent(s) for sublattice 1: /NB/: Zr <enter>
Major constituent(s) for sublattice 2: /VA/: <enter>
```

O comando acima diz ao thermo-calc que a fase “BCC_A2” deve ser separada em duas variantes, que serão identificadas como “BCC_A2” e “BCC_A2#2” e que o conjunto 2 é rico em zircônio.

Notamos (usando os comandos **LIST_CONDITIONS** e **LIST_AXIS_VARIABLE**) que o comando **SPECIAL_OPTIONS** apaga toda a memória, precisamos, portanto redefinir as duas condições iniciais que havíamos usado anteriormente e os eixos do gráfico:

```
POLY_3:s-con t=2017 p=101325 n=1 w(nb) = 0.2 <enter>
POLY_3:c-e <enter>
POLY_3:c-e <enter>
POLY_3:add <enter>
Direction /Default/: <enter>
POLY_3:s-con t=800 w(nb)=0.1 <enter>
POLY_3:c-e <enter>
```

¹Exceto para o caso dos carbonetonitretos $M(C,N)$ e $M_2(C,N)$, que são definidos como domos de imiscibilidade das soluções sólidas intersticiais com estrutura CFC e HCP respectivamente, como veremos mais adiante neste curso.

Vemos que após o último comando **COMPUTE_EQUILIBRIUM** o programa resulta em uma mensagem de erro. Isto ocorre porque o programa está confuso, sem saber o que fazer com os dois conjuntos de fases "BCC_A2". Vamos então forçar um equilíbrio inicial com o comando **SET_ALL_START_VALUES**:

```
POLY_3: s-a-s-v <enter>
Automatic start values for constituents? /Y/: N <enter>

Should LIQUID be stable? /N/: <enter>
Major constituent(s) /*/: <enter>

Should BCC_A2 be stable? /N/: Y <enter>
Major constituent(s) /*/: Zr <enter>

Should BCC_A2#2 be stable? /Y/: <enter>
Major constituent(s) /*/: Nb <enter>

Should FCC_A1 be stable? /N/: N <enter>
Major constituent(s) /*/: Zr <enter>

Should HCP_A3 be stable? /N/: <enter>
Major constituent(s) /*/: Zr <enter>
```

Este comando força o programa a considerar uma condição inicial para o cálculo onde uma fase BCC_A2 rica em Nióbio está em equilíbrio com uma fase BCC_A2 rica em Zircônio. Agora o comando **COMPUTE_EQUILIBRIUM** não retorna mais a mensagem de erro² e esta nova condição de equilíbrio pode ser adicionada. A seguir definimos os eixos, rodamos **MAP** e imprimimos o gráfico usando o Postprocessor (como já fizemos anteriormente). O resultado, agora mais que satisfatório, está na Figura 3.

²Observe que o equilíbrio neste temperatura é entre uma fase HCP e uma fase BCC rica em Zr e não entre duas fases BCC como impusemos acima, mas isto não importa: o comando foi usado apenas para gerar condições iniciais para os dois conjuntos de fases BCC.

THERMO-CALC (**.02.22:16.15) :PMT2512 Exercício 1

