The implementation of an algorithm to calculate the thermodynamic equilibrium for multi-component systems with many different models for non-ideal phases.

Bo Sundman

Gif sur Yvette, France

DRAFT Version April 25, 2015

Abstract

Thermodynamics is a central part in materials science. Thermodynamic models provides a unique method to combine experimental data and results from first principle calculations in databases. These databases are essential to provide values of many different thermodynamic properties in software tools for simulating materials processes and to predict their final properties.

A well established algorithm to calculate thermodynamic equilibria for multi-component systems using different kinds of conditions and with many non-ideal solution phases modelled in different ways is explained in detail. In particular how to handle phases with variable amount of atoms and how to handle different types of conditions and changes of the set of stable phases during iterations.

The algorithm can also be used to calculate properties outside the equilibrium state as required for the simulation of phase transitions.

This is part of the documentation of the Open Calphad initiative. There is also documentation available for the model package, General Thermodynamic Package (GTP) and there is ongoing work to develop and document the step/map/plot procedures.

There is also development of an application software interface and attemps to develop an assessment module.

Contents

3

1 Introduction

The algorithm presented here was derived and explained in slightly different forms by Gunnar Erikssom [75Eri], Mats Hillert [81Hil] and in the thesis by Bo Jansson [84Jan] and in a paper by Lukas [82Luk]. These papers give the theoretical background of the algorithm but they are very dense and difficult to understand. In this paper the algorithm and its implementation is explained in more detail, see also Sundman et al.[15Sun2]. Some corrections and extentions comparated to the published version are included in this text.

The implementation is done as part of the Open Calphad initiative [15Sun1] to provide a free software for thermodynamic calculations. This software will provide a useful link between experimental work, first principle calculations and applications like simulatons of phase transformations and microstructures using phase field methods.

2 The thermodynamic model

Phases in a thermodynamic system can be very different, examples are gas, liquid, amorpheous and many different crystalline forms. In any system the Gibbs energy, G, is given at equilibrium by

$$G = \sum_{\alpha} \aleph^{\alpha} G_M^{\alpha}(T, P, Y) \tag{1}$$

where $\aleph^{\alpha}$ is the number of moles of formula units of the phase α and G_M^{α} is the Gibbs energy per mole formula unit of α. Each phase can be modelled differently but its Gibbs energy depend on the temperature, pressure and its constitution, here denoted by T, P and Y.

Note that this definition of the molar Gibbs energy is different from the normal

$$G_m = \frac{G}{N} \tag{2}$$

where N is the total number of moles of components. But in this paper G_M will always be used according to the definition in eq. 1, using the formula unit explained in the next section.

At equilibrium at constant T, P and overall composition will be at a minimum in the Gibbs energy. For other conditions like volume, chemical potential etc we can add Lagrangian constraints to obtain the appropriate function to minimize. Internal constraints, that the sum of constituent fraction on each sublattice can also be handelled by Lagarangian multipliers.

Each phase can be described with a different model but the explanations in this paper will mainly concern phases modelled with the compound energy formalism (CEF) [81Sun, 01Hil]. This include as special gases ideal gases, substitutional regular soultions, interstitial solutions, sublattice models etc. Some additional explanations will be given for the ionic liquid model [84Hil].

4

2.1   The formula unit of a phase

For a phase modelled with CEF we can have sublattices and the ratio of these are deonoted a_s, where s is the sublattice. For example the σ phase has 5 sublattices with a_s equal to 2, 4, 8, 8 and 8 sites. Sites can be whole numbers or fractions but the recommendation is to use the smallest integer values. Thus a Laves phase should be modelled $(A,B)_2(A,B)_1$ rather than $(A,B)_{0.666667}(A,B)_{0.333333}$ as the number of decimal digits may affect the mass balance as well as the Gibbs energy.

The constituent fraction for a phase α are denoted y_i^α. For a phase with sublattices there will also be a second index, s, y_{is}^α indicating the sublattice, as the same species i may be a constituent of several sublattices. When there is no possible confusion the phase and sublattice superscripts are often omitted and a summation over i will mean for all constituents in all sublattices.

For ease of understanding we will from now on use indices A, B etc. to denote components or elements and indices i, j etc. to denote constituents of a phase. In many cases a constituent of a phase is the same as an element but it can be a molecule or ion. The meaning of the term component is according the the Gibbs phase rule but a component will often be the same as an element.

The number of moles of a component A per mole formula unit of the phase, M_A^α is calculated as

$$M_A^\alpha = \sum_s a_s^\alpha \sum_i b_{Ai} y_{is}^\alpha \tag{3}$$

where b_{Ai} is the stoichiometric factor of component A in constituent i and y_{is}^α is the fraction of constituent i in sublattice s of phase α. As before a_s is the number of sites on sublattice s. The sum of constituent fractions on each sublattice is unity.

The total number of moles of components in a formula unit of the phase is thus:

$$M^\alpha = \sum_A M_A^\alpha \tag{4}$$

and the mole fraction is

$$x_A^\alpha = \frac{M_A^\alpha}{M^\alpha} \tag{5}$$

The total number of moles of component A, N_A, in a system with several phases is

$$N_A = \sum_\alpha \aleph^\alpha M_A^\alpha \tag{6}$$

where $\aleph^\alpha$ is the number of moles formula unit of the phase α. The total number of moles of components, N, in the system is

$$N = \sum_A N_A = \sum_A \sum_\alpha \aleph^\alpha M_A^\alpha \tag{7}$$

and the mole fraction of A in the whole system is

$$x_A = \frac{N_A}{N} \tag{8}$$

We have to be careful to distinguish between $N, \aleph, M, x, y$ and other composition related variables.

2.2 Differentials

The differential of M_A^α is

$$dM_A^\alpha = \sum_s \sum_i \left(\frac{\partial M_A^\alpha}{\partial y_{is}^\alpha} \right)_{y_{j \neq i}} dy_{is}^\alpha \tag{9}$$

where each partial derivative of M_A^α is

$$\left(\frac{\partial M_A^\alpha}{\partial y_{is}} \right)_{y_{j \neq i}} = a_s^\alpha b_{Ai} \tag{10}$$

Most of these derivatives will be zero as M_A often depends on a single or just a few y_{is}. But we can have molecules with several atoms as constituents and also vacancies in a sublattice and the number of moles of components per formula unt of the phase is often not fixed.

For the ionic liquid model the situation is different as the number of sites for cations and anions depend on the average charge on the other sublattice. That will be discussed in the context of that model, It does not affect the general procedure for the minimization of the Gibbs energy described here.

2.3 Examples

Understanding the meaning of formula unit is essential to the algorithm so a few examples will be given.

2.3.1 A gas phase with H and O

In a gas with the molecules (H_2, O_2, H_2O) the moles formula unit of the components H and O and the total number of moles in a formula unit, M, are:

$$
\begin{aligned}
M_H &= 2y_{H_2} + 2y_{H_2O} \\
M_O &= 2y_{O_2} + y_{H_2O} \\
M &= 2y_{H_2} + 2y_{O_2} + 3y_{H_2O}
\end{aligned}
$$

The number of moles of atoms per formula unit of this gas can thus very between 2 and 3. The mole fractions are:

$$
\begin{aligned}
x_H = \frac{M_H}{M} &= \frac{2y_{H_2} + 2y_{H_2O}}{2y_{H_2} + 2y_{O_2} + 3y_{H_2O}} \\
x_O = \frac{M_O}{M} &= \frac{2y_{O_2} + y_{H_2O}}{2y_{H_2} + 2y_{O_2} + 3y_{H_2O}}
\end{aligned}
$$

2.3.2 A crystalline phase with substitutional and interstitial constituents

An interstitial solution of C and N in the bcc phase in a steel with Cr is modelled (Cr, Fe)$_1$(C, N, Va)$_3$, where Va denotes a vacancy or vacant site. The moles formula units of the elements are:

$$
\begin{aligned}
M_C &= 3y_{C,2} \\
M_{Cr} &= y_{Cr,1} \\
M_{Fe} &= y_{Fe,1} \\
M_N &= 3y_{N,2} \\
M &= 1 + 3y_{C,2} + 3y_{N,2}
\end{aligned}
$$

We use a comma between the constituent and the sublattice when they are explicit. If there are only two sublattices one frequently use one or two primes, y'_{Cr} and y''_C to indicate the sublattice but with more than two sublattices that become cumbersome. The number of moles of atoms per formula unit of the bcc phase can thus very between 1 and 4. He mole fractions are:

$$
\begin{aligned}
x_C &= \frac{3y_{C,2}}{1 + 3y_{C,2} + 3y_{N,2}} \\
x_{Cr} &= \frac{y_{Cr,1}}{1 + 3y_{C,2} + 3y_{N,2}} \\
x_{Fe} &= \frac{y_{Fe,1}}{1 + 3y_{C,2} + 3y_{N,2}} \\
x_N &= \frac{3y_{N,2}}{1 + 3y_{C,2} + 3y_{N,2}}
\end{aligned}
$$

2.3.3 A crystalline phase with long range ordering

Finally for a σ phase in the Cr, Fe, Mo and Ni system modelled with only 3 sublattices and with restricted solublities, (Cr, Fe, Ni)$_{10}$(Cr, Mo)$_4$(Cr, Fe, Mo, Ni)$_{16}$, the moles formula units are

$$
\begin{aligned}
M_{Cr} &= 10y_{Cr,1} + 4y_{Cr,2} + 16y_{Cr,3} \\
M_{Fe} &= 10y_{Fe,1} + 16y_{Fe,3} \\
M_{Mo} &= 4y_{Mo,2} + 16y_{Mo,3} \\
M_{Ni} &= 10y_{Ni,1} + 16y_{Ni,3} \\
M &= 30
\end{aligned}
$$

The number of moles per formula unit is constant and equal to 30. The mole fractions are

$$x_{\text{Cr}} = \frac{10y_{\text{Cr},1} + 4y_{\text{Cr},2} + 16y_{\text{Cr},3}}{30}$$

$$x_{\text{Fe}} = \frac{10y_{\text{Fe},1} + 16y_{\text{Fe},3}}{30}$$

$$x_{\text{Mo}} = \frac{4y_{\text{Mo},2} + 16y_{\text{Mo},3}}{30}$$

$$x_{\text{Ni}} = \frac{10y_{\text{Ni},1} + 16y_{\text{Ni},3}}{30}$$

The reason for this somewhat lenghty explanation is it is very common to make misstakes, or be uncertain using different kinds of variables for the amount of material in a system or phase.

2.4   End-members of models

An important concept when modelling with CEF is the *end-member* which for a crystalline phase specifies one constituent in each sublattice of the phase. A phase can consist o a single end-member and for the gas phase each molecule is an end-member. In the bcc phase example above we have 6 end-members which can be denoted: (Cr:C), (Cr:N), (Cr:Va), (Fe:C), (Fe:N) and (Fe:Va). In the σ phase example there are 16 end-members, for example (Fe:Cr:Cr), (Fe:Cr:Fe), (Fe:Cr:Mo), (Fe:Cr:Ni) etc. Note that in most cases end-members represent compounds with several elements.

3   The Gibbs energy

The Gibbs energy, G is an extensive property and can be subdivided in may different way. One well known formula relates the Gibbs energy to the chemical potentials, μ_{A}, and the number of moles, N_{A} of the components:

$$G = \sum_{\text{A}} N_{\text{A}} \mu_{\text{A}} \tag{11}$$

The definition of the chemical potentials is

$$\mu_{\text{A}} = \left(\frac{\partial G}{\partial N_{\text{A}}} \right)_{T,P,N_{\text{B} \neq \text{A}}} \tag{12}$$

If a phase has a fixed composition it has only a Gibbs energy and we cannot calculate the individual chemical potentials of the components for this phase alone. But if we can vary the fraction of constituents in one or more sublattices it is possible to calculate some relations between chemical potentials as will be discussed below in the context of eq. 21.

We have already introduced a different definition of the molar Gibbs energy for each phase which is related to the formula unit of the phase defined by structure of the phase as described above. Using eq. 3 the molar Gibbs energy for a formula unit, G_M^α and for one mole of components, G_m^α, for a phase α are equal to:

$$G_M^\alpha = \sum_A M_A^\alpha \mu_A \tag{13}$$

$$G_m^\alpha = \sum_A x_A^\alpha \mu_A \tag{14}$$

So it is important to know what kind of "molar" quantity we use.

3.1 The differential of the Gibbs energy

The differential of the Gibbs energy is

$$dG = -SdT + VdP + \sum_A \mu_A dN_A \tag{15}$$

and at equilibrium in a closed system $dG = 0$. If we have several stable phases

$$dG = \sum_\alpha (\aleph^\alpha dG_M^\alpha + G_M^\alpha d\aleph^\alpha) \tag{16}$$

where dG_M^α for each phase can, using the molar Gibbs energy per formula unit, be expressed as differences of the independent variables T, P and M_A or the dependent y_{is}^α:

$$
\begin{aligned}
dG_M^\alpha &= -S_M^\alpha dT + V_M^\alpha dP + \sum_A \mu_A dM_A^\alpha \\
&= -S_M^\alpha dT + V_M^\alpha dP + \sum_A \mu_A \sum_s \sum_i \left(\frac{\partial M_A^\alpha}{\partial y_{is}^\alpha} \right)_{T,P,y_{j\neq i}} dy_{is}^\alpha
\end{aligned} \tag{17}
$$

3.2 The partial Gibbs energy with for an end-member

For a phase with sublattices it is not always possible to calculate directly the chemical potentials for the components but we can always calculate the partial Gibbs energies of the end-members, I, of the model. An end-member specifies one constituent in each sublattice:

$$G_I = G_M + \sum_s \left(\frac{\partial G_M}{\partial y_{is\in I}} \right)_{T,P,y_{j\neq i}} - \sum_s \sum_j y_{js} \left(\frac{\partial G_M}{\partial y_{js}} \right)_{T,P,y_{k\neq j}} \tag{18}$$

where the first summation is for the constituents i specified by the end-member I for each sublattice s. The second double summation is for all constituents j. For a phase with fixed composition $G_I = G_M$.

In the example for the bcc interstitial solution $(Fe, Cr)_1(C, N, Va)_3$ above it is not possible to express directly the partial Gibbs energy for C in the model. However, we have the end-members (Fe:Va) and (Fe:C) and we can calculate these two partials from the model:

$$G_{\text{Fe:Va}} = G_M + \left(\frac{\partial G_M}{\partial y_{\text{Fe},1}}\right)_{T,P,y_{j\neq\text{Fe}}} + \left(\frac{\partial G_M}{\partial y_{\text{Va},2}}\right)_{T,P,y_{j\neq\text{Va}}} - \sum_s \sum_j y_{js} \left(\frac{\partial G_M}{\partial y_{js}}\right)_{T,P,y_{k\neq j}}$$

(19)

$$G_{\text{Fe:C}} = G_M + \left(\frac{\partial G_M}{\partial y_{\text{Fe},1}}\right)_{T,P,y_{j\neq\text{Fe}}} + \left(\frac{\partial G_M}{\partial y_{\text{C},2}}\right)_{T,P,y_{j\neq\text{C}}} - \sum_s \sum_j y_{js} \left(\frac{\partial G_M}{\partial y_{js}}\right)_{T,P,y_{k\neq j}}$$

(20)

At equilibrium the partial Gibbs energies of the end-members are related to the chemical potentials of the components as

$$G_{\text{FeVa}_3} = G_{\text{Fe:Va}} = G_{\text{Fe}} = \mu_{\text{Fe}}$$
$$G_{\text{FeC}_3} = G_{\text{Fe:C}} = G_{\text{Fe}} + 3G_{\text{C}} = \mu_{\text{Fe}} + 3\mu_{\text{C}}$$

as the chemical potential of Va, $\mu_{\text{Va}} = 0$ at equilibrium. We can rearrange to obtain the chemical potential of C:

$$\mu_{\text{C}} = G_{\text{C}} = \frac{1}{3}(G_{\text{Fe:C}} - G_{\text{Fe:Va}}) = \frac{1}{3}\left[\left(\frac{\partial G_M}{\partial y_{\text{C},2}}\right)_{T,P,y_{\text{Va}}} - \left(\frac{\partial G_M}{\partial y_{\text{Va},2}}\right)_{T,P,y_{\text{C}}}\right]$$

(21)

As can be seen by the last part of eq. 21 it is independent of the constituent in the first sublattice so it does not matter if we had chosen the end-members (Cr:C) and (Cr:Va). At equilibrium the difference will be the same.

Even of there is no Cr so Fe is alone on its sublattice we can obtain the chemical potentials of both Fe and C because the fraction of C can vary.

But for some CEF models it is not possible to combine the end-members in such a way that one can extract the chemical potentials of the elements, in the model for for the σ phase above we cannot obtain the individual chemical potentials of the elements from the model. But the method described below to calculate the equilibrium can be applied also for such phases.

3.3 The Gibbs-Duhem relation

In eq. 15 there is no differential of the chemical potentials because the Gibbs-Duhem relation for each phase is:

$$\sum_A d\mu_A M_A^\alpha + S_M^\alpha dT + V_M^\alpha dP = 0$$

(22)

which must be valid at equilibrium.

4 Minimization with constraints

To minimize a function with constraints we apply a Lagarangian equation where each equality constraints has a multiplier. When the constraint is obeyed the minimim of the Lagrangian is the same as the origibal function. The multipliers can be used to find the method to vary the variables to fulfill the constraints.

4.1 The constraints

The variables in the Gibbs energy expression have several constraints. The first is that the sum of the site fractions on each sublattice is unity:

$$g_s^\alpha = 1 - \sum_i y_{is}^\alpha = 0 \tag{23}$$

For phases with ions the net charge must be zero and an external charge balance constraint must be added:

$$Q^\alpha = \sum_s a_s \sum_i \nu_i y_{is}^\alpha = 0 \tag{24}$$

where ν_i is the charge on constituent i.

The total Gibbs energy for a system, G, is given by eq. 1. The number of formula units of a phase α, $\aleph^\alpha$, must be equal to or larger than zero for a stable phase. If it becomes negative during iterations it will be removed from the stable set of phases.

For a closed system we have the constraint on the amount of components

$$f_A = \tilde{N}_A - N_A = \tilde{N}_A - \sum_\alpha \aleph^\alpha M_A^\alpha = 0 \tag{25}$$

where $\tilde{N}_A$ is the prescribed amount of component A. For a phase modelled $(A,B)_{0.75}(A,B)_{0.25}$ the value of $\aleph^\alpha$ is 4 times larger compared to the case that the phase had been modelled $(A,B)_3(A,B)_1$. Both models are allowed and the thermodynamic parameters in the second case must be 4 times larger than those in the first.

4.2 The Lagrangian equation

To minimize the Gibbs energy of a system with constraints we can use a Lagrangian equation as:

$$L = G + \sum_A f_A \mu_A + \sum_\alpha \eta_s^\alpha g_s^\alpha + \sum_\alpha \lambda^\alpha Q^\alpha + \sum_\psi \gamma^\psi \aleph^\psi \tag{26}$$

where $\mu_A, \eta_s^\alpha, \lambda^\alpha$ are multipliers for all phases and γ^ψ are multipliers for all phases ψ that are unstable with $\aleph^\psi = 0$. The important property of this Lagrantian is that it will have the same extremum points as the Gibbs energy G when the constraints are fullfilled. From now on it will rarely be indicated which variables are kept constant at the partial derivatives, the reader is expected to understand this from the text.

4.3 The derivative of the Lagrangian with respect to phase amounts

For the partial derivative of L with respect to the amount of a stable phase α we get:

$$\frac{\partial L}{\partial \aleph^\alpha} = G_M^\alpha - \sum_A f_A M_A^\alpha = 0 \tag{27}$$

and from this equation we can understand that the Lagrangian multiplier $f_A = \mu_A$, i.e. the chemical potential of component A. For a rigorous proof see [81Hil].

For an unstable phase ψ which is not included in the stable phase set we can calculate the derivative:

$$\frac{\partial L}{\partial \aleph^\psi} = G_M^\psi - \sum_i \mu_i M^{\psi_i} + \gamma^\psi = 0 \tag{28}$$

This means that the driving force, γ^ψ, for an unstable phase will be calculated as part of the minimization. An unstable phase may become stable during the iterations for the equilibrium and that is indicated by γ^ψ becomming positive.

On the other hand, if $\aleph^\alpha$ for a stable phase α becomes negative it means this phase has become unstable and should be removed from the stable set. In both cases we must change the set of stable phases as will be discussed in section 5.8.

4.4 The derivative of the Lagrangian with respect to constituent fractions

The partial derivative of L with respect to a constituent fraction y_{is}^α, keeping all other variables constant, we get:

$$\frac{\partial L}{\partial y_{is}^\alpha} = \aleph^\alpha \frac{\partial G_M^\alpha}{\partial y_{is}^\alpha} - \aleph^\alpha \sum_A \mu_A \frac{\partial M_A^\alpha}{\partial y_{is}^\alpha} - \eta_s^\alpha + \lambda^\alpha \frac{\partial Q^\alpha}{\partial y_{is}^\alpha} = 0 \tag{29}$$

We would like to use this equation in an iterative procedure to find the equilibrium and to obtain a linear correction of the difference between the current value of the constituent fractions and those of the equilibrium we expand the partial derivative of $\frac{\partial G_M^\alpha}{\partial y_{is}^\alpha}$ in a Taylor series of its idependent variables dT, dP and dy_{jt}^α:

$$\frac{\partial G_M^\alpha}{\partial y_{is}^\alpha} = \left(\frac{\partial G_M^\alpha}{\partial y_{is}^\alpha}\right) + \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha \partial T}\right) dT + \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha \partial P}\right) dP + \sum_t \sum_j \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha y_{jt}^\alpha}\right) dy_{jt}^\alpha \tag{30}$$

where the terms on the right hand side are calculated for the current T, P and Y_i and the term on the left hand side is the linearly "extrapolated" value. The last term on the right hand side is a summation over all constituents j on all sublattices t. In the rest of this paper it will often be written as just a summustion over j.

Inserting this in eq. 29 and changing to finite differences we get a system of linear equations depending on the corrections in $\Delta T, \Delta P$ and Δy_{is}^α:

$$\sum_t \sum_j \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha y_{jt}^\alpha} \right) \Delta y_{jt}^\alpha + \frac{\eta_s^\alpha}{\aleph^\alpha} =$$

$$\sum_A \mu_A \left(\frac{\partial M_A^\alpha}{\partial y_{is}^\alpha} \right) - \left(\frac{\partial G_M^\alpha}{\partial y_{is}^\alpha} \right) - \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha \partial T} \right) \Delta T - \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha \partial P} \right) \Delta P$$

$$(31)$$

In the following we will normally make the simplification that P is constant, i.e. $\Delta P = 0$, as P and T enter the equations in the same way. We are intersted to solve this for the fraction corrections Δy_i and can rearrange this in matrix notation, omitting the superscripts:

$$\begin{pmatrix} \frac{\partial^2 G_M}{\partial y_1^2} & \frac{\partial^2 G_M}{\partial y_1 \partial y_2} & \cdots & 1 & \cdots \\ \frac{\partial^2 G_M}{\partial y_1 \partial y_2} & \frac{\partial^2 G_M}{\partial y_2^2} & \cdots & 1 & \cdots \\ \vdots & & & & \\ 1 & 1 & \cdots & 0 & \cdots \\ \vdots & & & & \end{pmatrix} \begin{pmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \frac{\eta_1}{\aleph} \\ \vdots \end{pmatrix} = \begin{pmatrix} \sum_A \mu_A \frac{\partial M_A}{\partial y_1} - \frac{\partial G_M}{\partial y_1} - \frac{\partial^2 G_M}{\partial y_1 \partial T} \Delta T \\ \sum_A \mu_A \frac{\partial M_A}{\partial y_2} - \frac{\partial G_M}{\partial y_2} + \frac{\partial S_M}{\partial y_2} \Delta T \\ \vdots \\ 0 \\ \vdots \end{pmatrix} \quad (32)$$

The matrix on the left hand side is symmetric and the columns and rows with "1" representing the constraint that sum of constituent fractions on each sublattice is unity. Inverting this *phase matrix* we can express the corrections of the constuent fractions of each phase in the global potentials $\mu_A, \Delta T$ and ΔP. The use of the inverted phase matrix will be explained in more detail for several specific cases below.

5 Calculating the equilibrium

The solution must be calculated by an iterative process and each iteration is divided into two steps. To simplify the explanations a substitutional binary system (A,B) is used in many cases. As discussed below there is an initial step 0 to find a good start set of stable phases and their constitutions.

5.1 Step 0, Obtaining start values by grid minimizer

An iterative procedure for calculating the equilibria can only find a local equilibrium which depend on the intitial constitution of the phases and it is thus necessary to provide a reasonable estimate of the stable phases and constitution of all phases for the first step.

There are very many techniques to find a global minimum [09Flo] but one problem with the thermodynamic equilibrium is that one does not know beforehand how many

composition sets of a phase that is needed. If a phase has a miscibility gap it may be necessary to create additional composition sets for several phases with different compositions. The solution that problem, implemented in OC, is based on the method to solve the equilibrium for the case when all phases have a fixed composition. Such an equilibrium is easily found by a combination of simplex and steepest decent methods and in such a calculation we will at equilibrium always have as many stable phases as we have components, and that is the maximum.
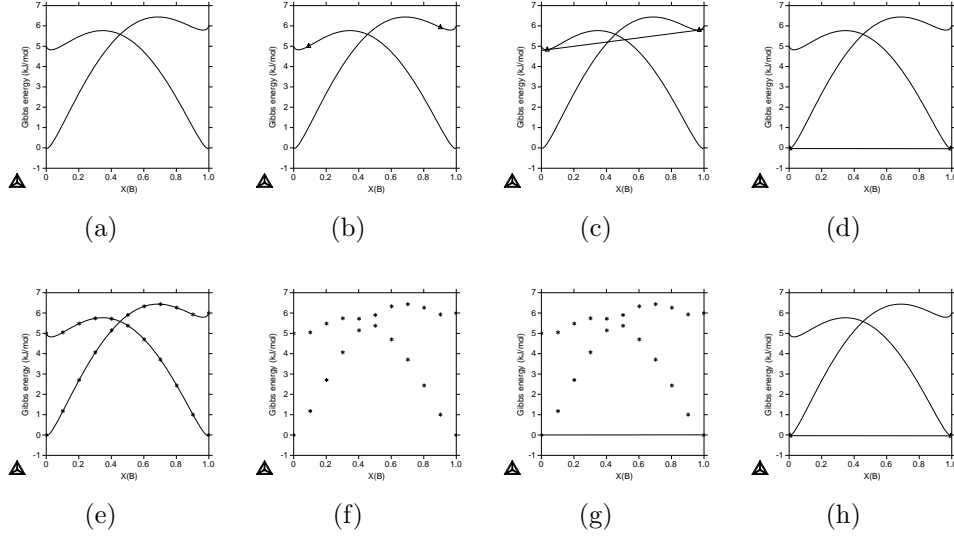
Figure 1: A case where start values matters. In (a) the Gibbs energy curves for two phases with miscibility gaps in a binary system are shown. If we have the initial constitutions marked in (b) the present algorithm will find the local equilibrium shown in (c) which is not the global minimum, the global minimum is shown in (d). In order to find good start values of the phases we can replace the Gibbs energy curves with calculated gridpoints as shown in (e) and then minimize these Gibbs energy of these gridpoints, exah treated as a separate phase with fixed composition as shown in (f). The grid minimizer will find the two gridpoints joined by a line in (g) as a minimum and using these compositions as startpoint for the calculation will give the correct global minimum using the present algorithm as in (h).

A more realistic case for the grid minimization is shown in Fig. 2 for the Fe-Mo system at 1400 K. At the stable equilibrium there is no miscibility gap but one of the phases has a metastable miscibility gap.

The same technique can be adopted to eystems with solution phases by calculating a number of gridpoints in each phase and then treat each of these as a separate phse. The number of gridpoint does not have to be very large even in multicomponent systems, normally 2000 points in each phase is sufficient even with more than 10 components. After finding the gridpoints respresenting the equilibrium we must check if some of them are in the same solution phase and check if they can be merged to a single point.
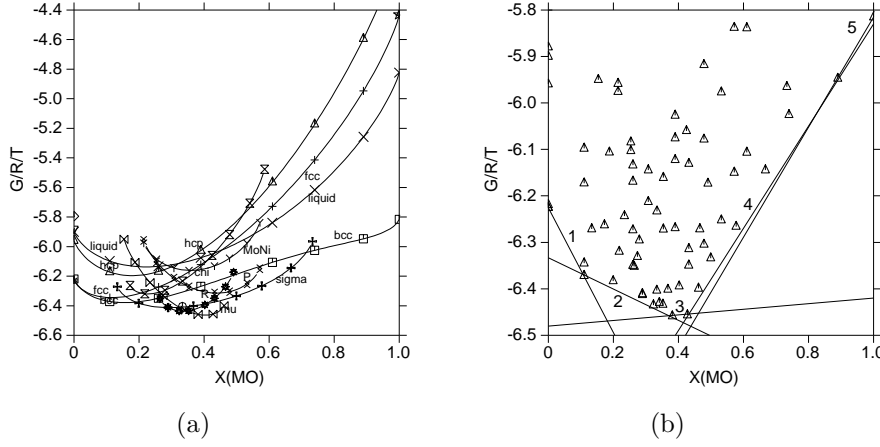
Figure 2: The Gibbs energy curves if the phases in the Fe-Mo system calculated at 1400 K in (a) together with the selected gridpoints. In (b) the gridpoints are treated as individual phases and the "convex hull" is drawn between the gridpoint pairs representing the lowest Gibbs energy at various compositions

That is not always the case if there are miscibility gaps in the solution phase. In Fig. 1 a case with two phases with miscibility gaps are shown.

The grid minimizer can also be used after an iterative equilibrium calculation to check if there are gridpoints below the calculated equilibrium surface. Such a technique is useful if the conditions does not allow an initial search for a grid minimum, for example if the value of T is not a condition.

If an equilibrium has already been calculated with almost the same conditions, like while performing a STEP calculation for a property diagram or MAP calculation for a phase diagram, it is not necessary to perform a grid minimization again but we can use the already calculated constitutions of the phases as start values. But it is important that now and again check if the current equilibrium is the global one as we may step into a miscibility gaps that is not stable initially and which is not detected by an iterative method.

It is also possible that the set of conditions does not allow a global gridminimization, for example if T is not known. In such cases we can start from a default initial constitution of the phases and after the equilibrium has been calculated, and the value of T is known and we can make a grid minimization to find if the calculated equilibrium is indeed the globally stable. If not the set of phases found by the grid minimizer are used to calculate the equilibrium again.

5.2 Step 1, the phase matrix

The derivation will first be given for a binary system, then it will be generallized.

15

5.2.1 The phase matrix for a binary system

For a substitutional binary phase (A,B) we can write the system of equations from eq. 31, denoting the constituents with index 1 and 2 to emphasize that components and constituents are not the same and at constant P, as

$$
\begin{pmatrix}
\frac{\partial^2 G_M^\alpha}{\partial y_1^2} & \frac{\partial^2 G_M^\alpha}{\partial y_1 \partial y_2} & 1 \\[2mm]
\frac{\partial^2 G_M^\alpha}{\partial y_1 \partial y_2} & \frac{\partial^2 G_M^\alpha}{\partial y_2^2} & 1 \\[2mm]
1 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
\Delta y_1 \\ \Delta y_2 \\ \eta
\end{pmatrix}
=
\begin{pmatrix}
-\frac{\partial G_M}{\partial y_1} - \frac{\partial^2 G_M}{\partial y_1 \partial T}\Delta T + \mu_A \frac{\partial M_A}{\partial y_1} + \mu_B \frac{\partial M_B}{\partial y_1} \\[2mm]
-\frac{\partial G_M}{\partial y_2} - \frac{\partial^2 G_M}{\partial y_2 \partial T}\Delta T + \mu_A \frac{\partial M_A}{\partial y_1} + \mu_B \frac{\partial M_B}{\partial y_2} \\[2mm]
0
\end{pmatrix}
\quad (33)
$$

On the left hand side we have the phase matrix for the binary (A,B) system, including the constraint that the sum of constituent fractions is unity:

$$
\begin{pmatrix}
\frac{\partial^2 G_M^\alpha}{\partial y_1^2} & \frac{\partial^2 G_M^\alpha}{\partial y_1 \partial y_2} & 1 \\[3mm]
\frac{\partial^2 G_M^\alpha}{\partial y_1 \partial y_2} & \frac{\partial^2 G_M^\alpha}{\partial y_2^2} & 1 \\[3mm]
1 & 1 & 0
\end{pmatrix}
$$

Before calculating the second derivatives in this matrix the constituent fractions should be checked that they are larger than a minimal (positive) value and normalized so the sum is unity. We cannot solve eq. 33 now as $\Delta T, \Delta P$ and μ_A are not known but we can invert the phase matrix and as we will use that several times below we write the inverted matrix as:

$$
\begin{pmatrix}
e_{11} & e_{12} & e_{13} \\
e_{21} & e_{22} & e_{23} \\
e_{31} & e_{32} & e_{33}
\end{pmatrix}
=
\begin{pmatrix}
\frac{\partial^2 G_M}{\partial y_i \partial y_j}
\end{pmatrix}^{-1}
$$

Only a part of this matrix is important because we are not interested in the η multiplier. We can write the solution for Δy_1 and Δy_2 as:

$$
\begin{pmatrix}
\Delta y_1 \\ \Delta y_2
\end{pmatrix}
=
\begin{pmatrix}
e_{11} & e_{12} \\
e_{21} & e_{22}
\end{pmatrix}
\begin{pmatrix}
-\frac{\partial G_M}{\partial y_A} - \frac{\partial^2 G_M}{\partial y_A \partial T}\Delta T + \mu_A \frac{\partial M_A}{\partial y_A} + \mu_B \frac{\partial M_B}{\partial y_A} \\[2mm]
-\frac{\partial G_M}{\partial y_B} - \frac{\partial^2 G_M}{\partial y_B \partial T}\Delta T + \mu_A \frac{\partial M_A}{\partial y_B} + \mu_B \frac{\partial M_B}{\partial y_B}
\end{pmatrix}
\quad (34)
$$

This gives a very important relation between the finite difference of a site fraction expressed as a function of several derivatives of the Gibbs energy and the "multipliers" μ_A, which are identical to the chemical potentials. This relation be used several times below. Writing the equation explicitly for constituent 1, using e_{ij} for the inverted phase matrix, gives:

$$
\begin{aligned}
\Delta y_1 \;=\; & e_{11}\left(-\frac{\partial G_M}{\partial y_1} - \frac{\partial^2 G_M}{\partial y_1 \partial T}\Delta T + \mu_A \frac{\partial M_A}{\partial y_1} + \mu_B \frac{\partial M_B}{\partial y_1} \right) + \\
& e_{12}\left(-\frac{\partial G_M}{\partial y_2} - \frac{\partial^2 G_M}{\partial y_2 \partial T}\Delta T + \mu_A \frac{\partial M_A}{\partial y_2} + \mu_B \frac{\partial M_B}{\partial y_2} \right)
\end{aligned}
\quad (35)
$$

5.2.2 The general equation for the correction of constituent fractions

Generallizing eq. 35 to any number of constituents, and also including variable P, gives for each constituent i on sublattice s:

$$\Delta y_{is} \;=\; c_{iG} + c_{iT}\Delta T + c_{iP}\Delta P + \sum_{A} c_{iA}\;\mu_{A} \tag{36}$$

where the coefficients in this equation can be calculated as

$$
\begin{aligned}
c_{iG} &= -\sum_{j} e_{ij}\frac{\partial G_M}{\partial y_j} \\[4pt]
c_{iT} &= -\sum_{j} e_{ij}\frac{\partial^2 G_M}{\partial T \partial y_j} \\[4pt]
c_{iP} &= -\sum_{j} e_{ij}\frac{\partial^2 G_M}{\partial P \partial y_j} \\[4pt]
c_{iA} &= \sum_{j} e_{ij}\frac{\partial M_A}{\partial y_j}
\end{aligned}
\tag{37}
$$

where i is the constituent in sublattice s and the summation over j is for all constituents all sublattices. A is a component. These coefficients will be used when formulating the equations in section 5.4.1 and also later to calculate partial derivators of state variables in section 6.

As already mentioned we cannot calculate Δy_{is} at present because the values of $\Delta T, \Delta P$ and μ_A are not known.

5.3 Charge balance

If some constituents have a net charge we must add the differential of eq. 24 to ensure that the phase is electrcally neutral.

$$Q^{\alpha} \;=\; \sum_{s} a_s \sum_{i} \nu_i y_{is}^{\alpha} = 0 \tag{38}$$

$$\Delta Q^{\alpha} \;=\; \sum_{s} a_s \sum_{i} \frac{\partial Q}{\partial y_{is}^{\alpha}} \Delta y_{is}^{\alpha} \tag{39}$$

where

$$\frac{\partial Q^{\alpha}}{\partial y_{is}^{\alpha}} \;=\; a_s \nu_i \tag{40}$$

This equation is part of the phase matrix, it should be the last row and column, call it q. After inverting the phase matrix the correction of the constituent fractions, eq. 36, will have an additional term

$$\Delta y_{is}^{\alpha} \;=\; c_{iG} + c_{iT}\Delta T + c_{iP}\Delta P + \sum_{\mathrm{A}} c_{i\mathrm{A}}\;\mu_{\mathrm{A}} - e_{iQ}Q^{\alpha} \tag{41}$$

where e_{iQ} are the last column in the inverted phase matrix and Q^{α} is the current charge of the phase.

As mentioned above a more complicated case is the ionic liquid model where the the cation and anion site ratios depend on the charge of the opposite site. In such a case there is no extra equation for the charge balanace but we cannot use that the derivatives $\frac{\partial M_{\mathrm{A}}}{\partial y_{is}}$ are constant.

5.4 Step 2, the external conditions

After calculating the inverted phase matrices for all phases and saving them we must calculate new values of the intensive variables ($\Delta T, \Delta P$ and μ_i) by making use of the external conditions. The most common set of external conditions are fixed T, P and amount of the components, i.e. mass balance conditions. Here we now describe how to formulate the equations that can determine these.

For each stable phase α we will have an equation:

$$G_M^{\alpha} = \sum_{\mathrm{A}} M_{\mathrm{A}}^{\alpha}\mu_{\mathrm{A}} \tag{42}$$

This ensures that all stable phases are on the same hyperplane of chemical potentials. For equilibrium calculations with variable T and P we must take into account any changes in these:

$$G_M^{\alpha} = \sum_{\mathrm{A}} M_{\mathrm{A}}^{\alpha}\mu_{\mathrm{A}} - \frac{\partial G_M}{\partial T}\Delta T - \frac{\partial G_M}{\partial P}\Delta P \tag{43}$$

The other equations depend on the conditions and the number of stable phases.

5.4.1 Condition on the amount of the components

The amount of each element summed over all phases is:

$$N_{\mathrm{A}} - \tilde{N}_{\mathrm{A}} = \sum_{\alpha} \aleph^{\alpha} M_{\mathrm{A}}^{\alpha} - \tilde{N}_{\mathrm{A}} = 0 \tag{44}$$

where $\tilde{N}_{\mathrm{A}}$ is the prescribed amount of moles of component A. The differential of N is

$$\Delta N_{\mathrm{A}} \;=\; \sum_{\alpha} \aleph^{\alpha}\Delta M_{\mathrm{A}}^{\alpha} + \sum_{\alpha} \Delta\aleph^{\alpha} M_{\mathrm{A}}^{\alpha} = 0 \tag{45}$$

From eq. 9 we have:

$$\Delta M_{\mathrm{A}}^{\alpha} \;=\; \sum_{i} \frac{\partial M_{\mathrm{A}}^{\alpha}}{\partial y_i^{\alpha}}\Delta y_i^{\alpha} \tag{46}$$

where the summation over i is for all constituents so we have omitted the sublattice index. We can approximate the differentials with finite differences and for Δy_i^α we now use eq. 36 and can write, omitting the phase superscripts and using the coefficients c_{iZ} defined in eq. 37:

$$\Delta M_A = \sum_B \mu_B \sum_i \frac{\partial M_A}{\partial y_i} c_{iB} - \sum_i \frac{\partial M_A}{\partial y_i} c_{iG} - \Delta T \sum_i \frac{\partial M_A}{\partial y_i} c_{iT} - \Delta P \sum_i \frac{\partial M_A}{\partial y_i} c_{iP} \quad (47)$$

where the sum over B is for all components. For fixed T and P this becomes:

$$\Delta M_A = \sum_B \mu_B \sum_i \frac{\partial M_A}{\partial y_i} c_{iB} - \sum_i \frac{\partial M_A}{\partial y_i} c_{iG} \quad (48)$$

Inserting the expression for c_{iX} gives

$$\Delta M_A = \sum_B \mu_B \sum_i \frac{\partial M_A}{\partial y_i} \sum_j \frac{\partial M_B}{\partial y_j} e_{ij} - \sum_i \frac{\partial M_A}{\partial y_i} \sum_j \frac{\partial G_M}{\partial y_j} e_{ij}$$

$$\Delta M_A = \sum_B \left(\sum_i \sum_j e_{ij} \frac{\partial M_A}{\partial y_i} \frac{\partial M_B}{\partial y_j} \right) \mu_B - \sum_i \sum_j e_{ij} \frac{\partial M_A}{\partial y_i} \sum_j \frac{\partial G_M}{\partial y_j} \quad (49)$$

5.4.2   Example: a binary system with a single stable phasse

If apply this to a binary A-B system with just one stable phase the sum of the fractions of A and B in this must fullfill the mass balance for each component we can insert this in eq. 45:

$$\Delta N_A = \aleph \left(\sum_B \mu_B \sum_i \frac{\partial M_A}{\partial y_i} c_{iB} - \sum_i \frac{\partial M_A}{\partial y_i} c_{iG} \right) + \Delta\aleph M_A \quad = \quad N_A - \tilde{N}_A = 0 \quad (50)$$

In the published paper, [15Sun2], the difference $N_A - \tilde{N}_A$ was forgotten. Rearranging the terms we have for each element:

$$\aleph \sum_B \mu_B \sum_i \frac{\partial M_A}{\partial y_i} c_{iB} + \Delta\aleph M_A \quad = \quad \aleph \sum_i \frac{\partial M_A}{\partial y_i} c_{iG} + N_A - \tilde{N}_A \quad (51)$$

Again, the sum over i should be for all constituents in all sublattices. We can now combine this with eq. 42 to a system of linear equations:

$$\begin{pmatrix} M_A & M_B & 0 \\ \aleph \sum_i \frac{\partial M_A}{\partial y_i} c_{iA} & \aleph \sum_i \frac{\partial M_A}{\partial y_i} c_{iB} & M_A \\ \aleph \sum_i \frac{\partial M_B}{\partial y_i} c_{iA} & \aleph \sum_i \frac{\partial M_B}{\partial y_i} c_{iB} & M_B \end{pmatrix} \begin{pmatrix} \mu_A \\ \mu_B \\ \Delta\aleph \end{pmatrix} = \begin{pmatrix} G_M \\ \aleph \sum_i \frac{\partial M_A}{\partial y_i} c_{iG} + N_A - \tilde{N}_A \\ \aleph \sum_i \frac{\partial M_B}{\partial y_i} c_{iG} + N_B - \tilde{N}_B \end{pmatrix} \quad (52)$$

All the terms in eq. 52 except μ_A, μ_B and $\aleph$ are known and this means we can calculate new values of them which can be inserted in eq. 36 so we get new constitution

of α and can calculate the terms in eq. 52 again and solve this to get new values of the potentials. We can continue to iterate until the changes are sufficently small. The matrix in eq. 52 is called the *equilibrium matrix*.

Note that unstable phases also will have their constitution updatated for each iteration using eq. 36 which is valid for all phases. The driving force, γ^ψ for an unstable phase ψ phase is calculated by eq. 28.

If the driving force become positive for a phase that initially is unstable the set of stable phases should be changed. If we have several phases and $\aleph^\alpha$ becomes negative for a phase α that phase should be set as unstable. Some care must be taken when changing the set of stable phases as discussed in section 5.8.

5.4.3 Example: a binary system with two stable phases

We always use eq. 36 to calculate the corrections to the phase constitutions. The only thing that varies with the conditions and the set of stable phases is the equilibrium matrix.

For a binary system with fixed T and P and two stable phases α and β the system equations are (note that φ is used as phase summation index):

$$
\begin{pmatrix}
M_A^\alpha & M_B^\alpha & 0 & 0 \\
M_A^\beta & M_B^\beta & 0 & 0 \\
\sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} c_{iA}^\varphi & \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} c_{iB}^\varphi & M_A^\alpha & M_A^\beta \\
\sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_B^\varphi}{\partial y_i^\varphi} c_{iA}^\varphi & \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_B^\varphi}{\partial y_i^\varphi} c_{iB}^\varphi & M_B^\alpha & M_B^\beta
\end{pmatrix}
\begin{pmatrix}
\mu_A \\
\mu_B \\
\Delta\aleph^\alpha \\
\Delta\aleph^\beta
\end{pmatrix}
=
$$

$$
\begin{pmatrix}
G_M^\alpha \\
G_M^\beta \\
\sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} c_{iG}^\varphi + N_A - \tilde{N}_A \\
\sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_B^\varphi}{\partial y_i^\varphi} c_{iG}^\varphi + N_B - \tilde{N}_B
\end{pmatrix}
\quad (53)
$$

Iterativly solving this system for $\mu_A, \mu_B, \Delta\aleph^\alpha$ and $\Delta\aleph^\beta$ together with eq. 36, considering that the set of stable phases may change, will eventually lead to the equilibrium.

With fixed T and P and massbalance conditions there must always be a stable equilibrium even if we may have to change the set of stable phases to find it. But it is possible to prescribe conditions that have no solution as may happen in the next example.

5.4.4 Example: a binary system with unknown T and one stable phase prescribed.

It is possible to prescribe that a phase should be stable and this is treated as an external condition. Such a condition can either be set explicitly by the user or set automatically when following a line in a phase diagram. All lines in a phase diagram represent values of the axis variables where the amount of a phase zero. So when mapping a phase diagram one of the axis variables is calculated by the condition that a phase should be stable with zero amount.

We assume that the remaining conditions are that we have fixed amounts of the components A and B and fixed P. We must have one phase stable with variable amount so we have two phases stable, one with zero amount. Note that with these conditions it is possible that no equilibrium exists.

For a binary system with unknown T and two stable phases, one of which, β, that is prescribed to be stable with amount zero, will have an equilibrium matrix as:

$$
\begin{pmatrix}
M_{\mathrm{A}}^{\alpha} & M_{\mathrm{B}}^{\alpha} & 0 & -\frac{\partial G_M^{\alpha}}{\partial T} \\
M_{\mathrm{A}}^{\beta} & M_{\mathrm{B}}^{\beta} & 0 & -\frac{\partial G_M^{\beta}}{\partial T} \\
\aleph^{\alpha} \sum_i \frac{\partial M_{\mathrm{A}}^{\alpha}}{\partial y_i^{\alpha}} c_{i\mathrm{A}}^{\alpha} & \aleph^{\alpha} \sum_i \frac{\partial M_{\mathrm{A}}^{\alpha}}{\partial y_i^{\alpha}} c_{i\mathrm{B}}^{\alpha} & M_{\mathrm{A}}^{\alpha} & -\sum_i \frac{\partial M_{\mathrm{A}}^{\alpha}}{\partial y_i^{\alpha}} c_{iT}^{\alpha} \\
\aleph^{\alpha} \sum_i \frac{\partial M_{\mathrm{B}}^{\alpha}}{\partial y_i^{\alpha}} c_{i\mathrm{A}}^{\alpha} & \aleph^{\alpha} \sum_i \frac{\partial M_{\mathrm{B}}^{\alpha}}{\partial y_i^{\alpha}} c_{i\mathrm{B}}^{\alpha} & M_{\mathrm{B}}^{\alpha} & -\sum_i \frac{\partial M_{\mathrm{B}}^{\alpha}}{\partial y_i^{\alpha}} c_{iT}^{\alpha}
\end{pmatrix}
\begin{pmatrix}
\mu_A \\
\mu_B \\
\Delta\aleph^{\alpha} \\
\Delta T
\end{pmatrix} =
$$

$$
\begin{pmatrix}
G_M^{\alpha} \\
G_M^{\beta} \\
\aleph^{\alpha} \sum_i \frac{\partial M_{\mathrm{A}}^{\alpha}}{\partial y_i^{\alpha}} c_{iG}^{\alpha} + N_{\mathrm{A}} - \tilde{N}_{\mathrm{A}} \\
\aleph^{\alpha} \sum_i \frac{\partial M_{\mathrm{B}}^{\alpha}}{\partial y_i^{\alpha}} c_{iG}^{\alpha} + N_{\mathrm{B}} - \tilde{N}_{\mathrm{B}}
\end{pmatrix} \quad (54)
$$

In this system of equations there is only one phase with variable amount, $\Delta\aleph^{\alpha}$, as $\aleph^{\beta} = 0$. Note that we take into account that the Gibbs energy of the phases varies with ΔT according to eq. 43. We must also take into account that the equation for Δy_i depend on ΔT according to eq. 36.

5.4.5   Example: a binary system with one condition of a chemical potential

The final example is for a binary system with fixed T, P, the total amount of one components, N_{A}, and the chemical potential, μ_{B} of the other. Two phases are stable.

$$
\begin{pmatrix}
M_{\mathrm{A}}^{\alpha} & 0 & 0 \\
M_{\mathrm{A}}^{\beta} & 0 & 0 \\
\sum_{\varphi} \aleph^{\varphi} \sum_i \frac{\partial M_{\mathrm{A}}^{\varphi}}{\partial y_i^{\varphi}} c_{i\mathrm{A}}^{\varphi} & M_{\mathrm{A}}^{\alpha} & M_{\mathrm{A}}^{\beta}
\end{pmatrix}
\begin{pmatrix}
\mu_{\mathrm{A}} \\
\Delta\aleph^{\alpha} \\
\Delta\aleph^{\beta}
\end{pmatrix} =
\begin{pmatrix}
G_M^{\alpha} - M_{\mathrm{B}}^{\alpha}\mu_{\mathrm{B}} \\
G_M^{\beta} - M_{\mathrm{B}}^{\beta}\mu_{\mathrm{B}} \\
\sum_{\varphi} \aleph^{\varphi} \sum_i \frac{\partial M_{\mathrm{A}}^{\varphi}}{\partial y_i^{\varphi}} (c_{iG}^{\varphi} - c_{i\mathrm{B}}^{\varphi}\mu_{\mathrm{B}}) + N_{\mathrm{A}} - \tilde{N}_{\mathrm{A}}
\end{pmatrix} (55)
$$

The summation over φ is over all stable phases.

5.5   Condition on volume

If the volume is prescribed as constant, $\tilde{V}$, we have an equation:

$$
dV \;=\; V - \tilde{V} = 0 \tag{56}
$$

where

$$
V \;=\; \sum_{\alpha} \aleph^{\alpha} V_M^{\alpha} \tag{57}
$$

$$
V_M^{\alpha} \;=\; \left(\frac{\partial G_M^{\alpha}}{\partial P}\right)_{T,Y} \tag{58}
$$

It is not necessary to have variable P, we may be able to change the volume even at constant P, for example by varying the amount of phases with different molar volumes or having a condition on a chemical potential which can change the amount of material in the system. We expand the differential of $dV = \Delta V$ as:

$$
\begin{aligned}
\Delta V &= \sum_\alpha \aleph^\alpha \left(\frac{\partial^2 G_M^\alpha}{\partial P \partial T} \Delta T + \frac{\partial^2 G_M^\alpha}{\partial P^2} \Delta P + \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} \Delta y_i^\alpha \right) + \sum_\alpha \frac{\partial G_M^\alpha}{\partial P} \Delta \aleph^\alpha \\
&= \sum_\alpha \aleph^\alpha \left(\sum_A \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iA} \mu_A + \left[\frac{\partial^2 G_M^\alpha}{\partial P \partial T} + \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iT} \right] \Delta T + \right. \\
&\quad \left. \left[\frac{\partial^2 G_M^\alpha}{\partial P^2} \Delta P + \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iP} \right] \Delta P + \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iG} \right) + \sum_\alpha \frac{\partial G_M^\alpha}{\partial P} \Delta \aleph^\alpha \\
&= V - \tilde{V} = 0
\end{aligned}
\tag{59}
$$

where Δy_i can be expressed as a function of $\Delta T, \Delta P$ and μ_A using eq. 36. Rearranging the equation for the equilibrium matrix for the unknown $\Delta T, \Delta P$ and μ_A gives:

$$
\begin{aligned}
\sum_\alpha \aleph^\alpha \sum_A \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iA} \mu_A + \sum_\alpha \aleph^\alpha \left(\frac{\partial^2 G_M^\alpha}{\partial P \partial T} + \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iT} \right) \Delta T \quad + \\
\sum_\alpha \aleph^\alpha \left(\frac{\partial^2 G_M^\alpha}{\partial P^2} + \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iP} \right) \Delta P + \sum_\alpha \frac{\partial G_M^\alpha}{\partial P} \Delta \aleph^\alpha \quad = \\
- \sum_\alpha \aleph^\alpha \sum_i \frac{\partial^2 G_M^\alpha}{\partial P \partial y_i^\alpha} c_{iG} + V - \tilde{V}
\end{aligned}
\tag{60}
$$

In the next sections we will not specify the phase when obvious.

5.6   Constant Gibbs energy or entropy

It would be rare to have a condition on the Gibbs energy or entropy of the system but as preparetion for a heat balance equation I will start by that. In the same way as for the volume the equation is:

$$
dG = G - \tilde{G} = 0
\tag{61}
$$

where

$$
G = \sum_\alpha \aleph^\alpha G_M^\alpha
\tag{62}
$$

$$
G_M^\alpha = \sum_A M_A^\alpha \mu_A
\tag{63}
$$

$$
dG_M^\alpha = \sum_A dM_A^\alpha \mu_A
\tag{64}
$$

where we can approximate $dM_A^\alpha = \Delta M_A^\alpha$ as in eq. 47. As we are only dealing with linear changes all terms multiplied with two potentials or potential differences are ignored and in the equation we keep only:

$$
dG = \sum_\alpha \aleph^\alpha \sum_A \sum_i \frac{\partial M_A^\alpha}{\partial y_i} c_{iG} \mu_A = G - \tilde{G} = 0
\tag{65}
$$

22

This looks nice and simple but maybe not so useful. For a condition on the entropy we have

$$dS \;=\; S - \tilde{S} = 0 \tag{66}$$

where

$$S \;=\; \sum_{\alpha} \aleph^{\alpha} S_M^{\alpha} \tag{67}$$

$$S_M^{\alpha} \;=\; -\left(\frac{\partial G_M^{\alpha}}{\partial T}\right)_{P,Y} = -\frac{\partial}{\partial T}\left(\sum_{A} M_A^{\alpha}\mu_A\right) = -\sum_{A} M_A^{\alpha}\frac{\partial \mu_A}{\partial T} \tag{68}$$

$$dS_M^{\alpha} \;=\; -\sum_{A} dM_A^{\alpha}\frac{\partial \mu_A}{\partial T} \tag{69}$$

but I have no idea how to calculate $\frac{\partial \mu_A}{\partial T}$. This must be the wrong track. If we do not introduce the chemical potentials we can write

$$dS_M^{\alpha} = -\frac{\partial^2 G_M^{\alpha}}{\partial T^2}\Delta T - \frac{\partial^2 G_M^{\alpha}}{\partial T\partial P}\Delta P - \sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}\Delta y_i \tag{70}$$

and we can formulate an equation as we did for the volume in eq. 59:

$$
\begin{aligned}
dS \;&=\; -\sum_{\alpha}\aleph^{\alpha}\left(\frac{\partial^2 G_M^{\alpha}}{\partial T^2}\Delta T + \frac{\partial^2 G_M^{\alpha}}{\partial T\partial P}\Delta P + \sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}\Delta y_i\right) - \sum_{\alpha}\frac{\partial G_M^{\alpha}}{\partial T}\Delta\aleph^{\alpha} \\
&=\; S - \tilde{S} = 0
\end{aligned}
\tag{71}
$$

where Δy_i can be expressed as a function of $\Delta T, \Delta P$ and μ_A using equation 36.

$$
\begin{aligned}
-\sum_{\alpha}\aleph^{\alpha}\frac{\partial^2 G_M^{\alpha}}{\partial T^2}\Delta T - \sum_{\alpha}\aleph^{\alpha}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial P}\Delta P - \sum_{\alpha}\aleph^{\alpha}\sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}\Delta y_i - \sum_{\alpha}\frac{\partial G_M^{\alpha}}{\partial T}\Delta\aleph^{\alpha} \;&=\; S - \tilde{S} \\
-\sum_{\alpha}\aleph^{\alpha}\frac{\partial^2 G_M^{\alpha}}{\partial T^2}\Delta T - \sum_{\alpha}\aleph^{\alpha}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial P}\Delta P - \sum_{\alpha}\frac{\partial G_M^{\alpha}}{\partial T}\Delta\aleph^{\alpha} & \\
-\sum_{\alpha}\aleph^{\alpha}\sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}\left(\sum_{A}c_{iA}\mu_A + c_{iT}\Delta T + c_{iP}\Delta P + c_{iG}\right) \;&=\; S - \tilde{S} \\
-\sum_{\alpha}\aleph^{\alpha}\left(\frac{\partial^2 G_M^{\alpha}}{\partial T^2} + \sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}c_{iT}\right)\Delta T - \sum_{\alpha}\aleph^{\alpha}\left(\frac{\partial^2 G_M^{\alpha}}{\partial T\partial P} + \sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}c_{iP}\right)\Delta P & \\
-\sum_{\alpha}\aleph^{\alpha}\sum_{A}\sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}c_{iA}\mu_A - \sum_{\alpha}\frac{\partial G_M^{\alpha}}{\partial T}\Delta\aleph^{\alpha} = \sum_{\alpha}\aleph^{\alpha}\sum_{i}\frac{\partial^2 G_M^{\alpha}}{\partial T\partial y_i}c_{iG} \;&+\; S - \tilde{S}
\end{aligned}
\tag{72}
$$

This means we should not use eq. 65 for a condition on G, but write such an equation as:

$$
\begin{aligned}
dG \;&=\; \sum_{\alpha}\aleph^{\alpha}\left(\frac{\partial G_M^{\alpha}}{\partial T}\Delta T + \frac{\partial G_M^{\alpha}}{\partial P}\Delta P + \sum_{i}\frac{\partial G_M^{\alpha}}{\partial y_i}\Delta y_i\right) + \sum_{\alpha} G_M^{\alpha}\Delta\aleph^{\alpha} \\
&=\; G - \tilde{G} = 0
\end{aligned}
\tag{73}
$$

23

or written as an equation of the variables $\Delta\aleph, \Delta T, \Delta P$ and μ:

$$\sum_\alpha \aleph^\alpha(\frac{\partial G_M^\alpha}{\partial T} + \sum_i \frac{\partial G_M^\alpha}{\partial y_i}c_{iT})\Delta T + \sum_\alpha \aleph^\alpha(\frac{\partial G_M^\alpha}{\partial P} + \sum_i \frac{\partial G_M^\alpha}{\partial y_i}c_{iP})\Delta P$$

$$+\sum_\alpha \aleph^\alpha \sum_A \sum_i \frac{\partial G_M^\alpha}{\partial y_i}c_{iA}\mu_A + \sum_\alpha G_M^\alpha \Delta\aleph^\alpha = -\sum_\alpha \aleph^\alpha \sum_i \frac{\partial G_M^\alpha}{\partial y_i}c_{iG} \quad + \quad G - \tilde{G}$$

$$(74)$$

Just to remind you

$$c_{iA} \quad = \quad \sum_j e_{ij}\frac{\partial M_A}{\partial y_j} \tag{75}$$

where e_{ij} is the inverted phase matrix. The subroutine **calc_dgdytermsh** calculates

$$mamu(A) \quad = \quad \sum_i F_i c_{iA} = \sum_i F_i \sum_j e_{ij}\frac{\partial M_A}{\partial y_j} \tag{76}$$

$$maT \quad = \quad \sum_i F_i c_{iT} = \sum_i F_i \sum_j e_{ij}\frac{\partial^2 G_M}{\partial T \partial y_j} \tag{77}$$

$$maP \quad = \quad \sum_i F_i c_{iT} = \sum_i F_i \sum_j e_{ij}\frac{\partial^2 G_M}{\partial P \partial y_j} \tag{78}$$

$$maG \quad = \quad \sum_i F_i c_{iG} = \sum_i F_i \sum_j e_{ij}\frac{\partial G_M}{\partial y_j} \tag{79}$$

where F_i is an array passed to this subroutine. For a condition on G we have $F_i = \frac{\partial G_M}{\partial y_i}$.

For the heat balance explained below it is $F_i = \frac{\partial G_M}{\partial y_i} - T\frac{\partial^2 G_M}{\partial T \partial y_i}$ and for the mass balance equations it is $F_i = \frac{\partial M_A}{\partial y_i}$.

5.7   Heat balance calculation

The enthalpy is $H = G + TS$ and conditions on H are quite frequent. We have as before the equation:

$$dH \quad = \quad H - \tilde{H} = 0 \tag{80}$$

where

$$H \quad = \quad \sum_\alpha \aleph^\alpha H_M^\alpha \tag{81}$$

$$H_M^\alpha \quad = \quad G_M^\alpha + TS_M^\alpha \tag{82}$$

$$dH_M^\alpha \quad = \quad dG_M^\alpha + TdS_M^\alpha \tag{83}$$

Thermodynamics is confusing, maybe we should use?

$$dH_M^\alpha \quad = \quad dG_M^\alpha + TdS_M^\alpha + S_M^\alpha dT \tag{84}$$

but this cannot be correct. Why? Because $H = H(S, P)$ is a function of S but of T?

In eq. 83 the differentials for G_M and S_M are given by eqs 73 and 71 respectivly. Combining there we have:

$$
\begin{aligned}
dH &= \sum_\alpha \aleph^\alpha \left[\left(\frac{\partial G_M}{\partial T} - T \frac{\partial^2 G_M}{\partial T^2} \right) \Delta T + \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} \right) \Delta P + \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) \Delta y_i \right] + \\
&\quad \sum_\alpha \left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^\alpha = H - \tilde{H} = 0
\end{aligned}
\tag{85}
$$

This must be the equation! If we insert eq. 36 we get

$$
\begin{aligned}
\sum_\alpha \aleph^\alpha \left[\left(\frac{\partial G_M}{\partial T} - T \frac{\partial^2 G_M}{\partial T^2} \right) \Delta T + \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} \right) \Delta P + \right. \\
\left. \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) \left(c_{iG} + c_{iT} \Delta T + c_{iP} \Delta P + \sum_A c_{iA} \mu_A \right) \right] + \\
\sum_\alpha \left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^\alpha = H - \tilde{H}
\end{aligned}
\tag{86}
$$

More rearrangements to have the coefficients in the equilibrium matrix for the independent variables $\Delta T, \Delta P, \mu_A$ and $\Delta \aleph^\alpha$ in the equation for fixed H:

$$
\begin{aligned}
\sum_\alpha \aleph^\alpha \left(\frac{\partial G_M}{\partial T} - T \frac{\partial^2 G_M}{\partial T^2} + \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iT} \right) \Delta T \quad + \\
\sum_\alpha \aleph^\alpha \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} + \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iP} \right) \Delta P \quad + \\
\sum_\alpha \aleph^\alpha \sum_A \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iA} \mu_A + \sum_\alpha \left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^\alpha \quad = \\
-\sum_\alpha \aleph^\alpha \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iG} + H - \tilde{H}
\end{aligned}
\tag{87}
$$

We see again how useful it is to have expressed Δy_i as a function of the potentials and that all second derivatives are calculated analytically in the model package.

5.8   Changing the set of stable phases

When the amount of a stable phase becomes negative at an iteration it means this phase should be removed from the set of stable phases. And if the driving force for an unstable phase according to eq. 28 becomes positive that phase should be added.

$$
\gamma^\psi = \sum_A \mu_A M_A^{\psi_i} - G_M^\psi
\tag{88}
$$

where the sum over A is for all components. If T and P are variable the ΔT and ΔP are also included in this equation as in eq. 43.

This is basically a trivial operation but we should take care that the same phase is removed and added at every second iteration. Normally we should allow a few iterations after a change in the set of stable phases before another change is allowed.

It can also happen that the amount of the single stable phase becomes negative and it is clearly impossible to have a system without a single stable phase. This may indicate that the set of external conditions are unreasonable or that the model parameters are wrong.

A third case that may case problem is when more phases become stable than allowed by the Gibbs phase rule:

$$f = n - p + 2 \tag{89}$$

where f is the degrees of freedom, n number of components, p number of stable phases and 2 represent variable T and P. In order to calculate an equilibrium we must have set so many conditions that f is zero. For a binary system that means 4 conditions. If one condition is constant P, we can at most have 3 phases stable.

For a calculation in a binary system with fixed T and P cannot have more than 2 stable phases. If a third phase wants to become stable this one must replace one of the already stable phases.

5.9   Potentials as conditions

In most cases above T and P have been fixed and in one example a chemical potential is fixed. We cannot have all conditions as potentials becuase then we are trying to minimize the Gibbs-Duhem relation which is always zero. At keast one condition must be an extensive property.

5.10   Normallized state variables as conditions

In the examples above the total amounts of the components has been fixed. This simplifies the equations because if we use normallized state variables like mole fractions rather than total amounts we must take into account the derivatives of the normallizing property when deriving the equations for the system and that these may change during the iterations. In order to handle this the following equations must be used as derived by [84Jan].

Conditions on extensive properties can in general be set on a global variable (summed over all stable phases) or for a specific phase and it can be a total value or a normallized one (normallized per moles, mass or volume). For a global total property we can formulate the condition like for N_A in eq. 44.

For a normallized phase specific variable we have, using Z for the variable and K for the normallizing quantity:

$$z^\alpha = \frac{Z_M^\alpha}{K_M^\alpha} \tag{90}$$

If the prescribed value is $\tilde{z}^\alpha$ we have

$$\Delta z^{\alpha} = \tilde{z}^{\alpha} - z^{\alpha} = \frac{\Delta Z_M^{\alpha} - z^{\alpha} \Delta K_M^{\alpha}}{K_M^{\alpha}} \tag{91}$$

where the subscript m means per mole formula unit of the phase. The finite difference ΔZ^{α} can be expressed in the model variables of the phase, T, P and y_{is}^{α}, as:

$$\Delta Z_M^{\alpha} = \frac{\partial Z_M^{\alpha}}{\partial T} \Delta T + \frac{\partial Z_M^{\alpha}}{\partial P} \Delta P + \sum_t \sum_j \frac{\partial Z_M^{\alpha}}{\partial y_{js}^{\alpha}} \Delta y_{js}^{\alpha} \tag{92}$$

and similarly for ΔK_M^{α}. Here we can again use eq. 36 to express Δy_{js}^{α} as a function of ΔT, ΔP and the chemical potentials μ_A.

To prescribe a global value for the whole system we can simply sum for all stable phases:

$$Z = \sum_{\alpha} \aleph^{\alpha} Z_M^{\alpha} \tag{93}$$

Using a normallizing factor K gives z:

$$z = \frac{Z}{K} \tag{94}$$

Expressing the deviation from the prescribed value $\tilde{z}$ using finite differences is:

$$\Delta z = \tilde{z} - z = \sum_{\alpha} \aleph^{\alpha} \left(\frac{\Delta Z_M^{\alpha} - z \Delta K_M^{\alpha}}{K} \right) + \sum_{\alpha} \left(\frac{Z_M^{\alpha} - z K_M^{\alpha}}{K} \right) \Delta \aleph^{\alpha} \tag{95}$$

As before we can replace ΔZ_M^{α} and ΔK_M^{α} to express this as a function of ΔT, ΔP and the chemical potentials μ_A using eq. 92 and eq. 36. A phase with prescribed fixed amount have $\Delta \aleph = 0$.

A typical use of eq. 95 is when there is a condition on the mole or mass fractions of a component.

5.11 Generallizing the equilibrium matrix

All examples shown above has been for binary cases. For the phase matrix and the way to calculate corrections to the constituent fractions, eq. 36, is completely general and can be used for all phases in a multicomponent system.

As shown with the examples the system equations must be adopted to the conditions set by the user and it will also change if the number of stable phases changes dureing the iterations. The coefficients calculated from the inverted phase matrix, eq. 37, are important in constructing the system equations.

5.12 Special treatment of the ionic liquid model

The two-sublattice partially ionic liquid model is a unified model for liquid with or without tendency for ionization. It is derived and explained in detail in [84Hil]. In this model the short range ordering is in fact modelled as long range ordering using two different sites for cations and anions. But in the liquid the sites does not represent fixed

positions in space but is a way to calculate the configurational entropy with separate mixing of anions and cations.

The notation is

$$(C^{+\nu_C})_P (D^{-\nu_D}, \text{Va}, N)_Q \tag{96}$$

where $C^{+\nu_C}$ represent cations with charge $+\nu_C$, $D^{-\nu_D}$ represent anions with charge $-\nu_D$, Va represent hypthetical vacancies with an induced charge $-Q$ and N represent neutral constituents. P is used in a new meaning here as P and Q are site ratios which are equal to the average charge on the opposite sublattice:

$$P = \sum_A y_A \nu_A + y_{\text{Va}} Q \tag{97}$$

$$Q = \sum_C y_C \nu_C \tag{98}$$

An example of this model is the liquid Cu-Fe-S modelled as

$$(Fe^{+2}, Cu^{+1})_P (S^{-2}, \text{Va}, S)_Q \tag{99}$$

where the binary metallic liquid Cu-Fe is described with just Va on the second sublattice to compensate for the charge. The pure sulphur liquid has just neutral S. Note that in that case $P = 0$. There is short range ordering in the liquid at the compositions FeS and Cu_2S when the second sublattice has mainly S^2 and the first a single metallic ion.

The mass balance equation is like for the CEF model:

$$M_A = P \sum_i b_{Ai} y_i + Q(\sum_j b_{Aj} y_j + \sum_k b_{Ak} y_k) \tag{100}$$

where all b_{Ai} represent the stoichiometric factor of element A in any cation, anion or neutral and all y_i the fraction of the constituent i. The derivative of this slightly more complicated than for the CEF model as P and Q are nort constant:

$$dM_A = dP \sum_i b_{Ai} y_i + P \sum_i b_{Ai} dy_i + dQ(\sum_j b_{Aj} y_j + \sum_k b_{Ak} y_k) + Q(\sum_k b_{Ak} dy_k + \sum_j b_{Aj} dy_j) \tag{101}$$

where

$$dP = \sum_A dy_A \nu_A + dy_{\text{Va}} Q + y_{\text{Va}} dQ \tag{102}$$

$$dQ = \sum_C dy_C \nu_C \tag{103}$$

So the partial derivatives $\frac{\partial M_A}{\partial y_i}$ are no longer constants which makes the equations for the equilibrium calculation slightly more complicated to implement.

6 Calculating derivatives using the result of an equilibrium calculation, the dot derivative

After an equilibrium calculation it is possible to obtain additional properties using the calculated values of G and its first and second derivatives. This tyoe of calculations was first implemented in Thermo-Calc by Bo Jansson but he never documented his work.

6.1 Calculating heat capacity

The heat capacity is not directly available after an equilibrium calculation but by combining the values of appropiate coefficients in eq. 37 we can calculate this without making any new equilibrium calculation with a small step in T. The heat capacity at constant P is defined as

$$C_P = \left(\frac{\partial H}{\partial T} \right)_{P,N_i} \tag{104}$$

We must derive a way to obtain this from the Gibbs energy, G, for a system at equilibrium distributed over a set of stable phases α is:

$$G(T,P,N_i) = \sum_\alpha \aleph^\alpha G_M^\alpha(T,P,y_{js}) \tag{105}$$

$$N_i = \sum_\alpha \aleph^\alpha N_i^\alpha \tag{106}$$

$$N_i^\alpha = \sum_s a_s^\alpha \sum_j b_{ij} y_{js}^\alpha \tag{107}$$

where N_i is the amount in moles of component i, $\aleph^\alpha$ is the amount of formula units of phase α and G_M^α is the Gibbs energy of α for one formula unit.

The constitution of the α phase is modelled using constituent fractions y_{js} for fraction of constituent j on sublattice s, a_s is the number of sites on sublattice s and b_{ij} is the stoichiometric factor of component i in constituent j.

The enthalpy, H, can be calculated from the Gibbs energy as:

$$G = H - TS = H + T \left(\frac{\partial G}{\partial T} \right)_{P,N_i} \tag{108}$$

$$H = G - T \left(\frac{\partial G}{\partial T} \right)_{P,N_i} \tag{109}$$

From the model we cannot calculate a derivative with respect to constant N_i so the derivative of G must be expanded as:

$$\begin{aligned}
\left(\frac{\partial G}{\partial T} \right)_{P,N_i} &= \sum_\alpha \frac{\partial \aleph^\alpha}{\partial T} G_M^\alpha + \sum_\alpha \aleph^\alpha \left(\frac{\partial G_M^\alpha}{\partial T} \right)_{P,N_i} \\
&= \sum_\alpha \frac{\partial \aleph^\alpha}{\partial T} G_M^\alpha + \sum_\alpha \aleph^\alpha \left[\left(\frac{\partial G_M^\alpha}{\partial T} \right)_{P,y_{js}} + \sum_{js} \left(\frac{\partial G_M^\alpha}{\partial y_{js}} \right)_{P,y_{kt \neq js}} \frac{\partial y_{js}}{\partial T} \right]
\end{aligned} \tag{110}$$

The last equation comes from the fact that the equilibrium is calculated for fixed amounts of the components but the Gibbs energy of the phase depend on the constituent fractions, the term $\left(\frac{\partial G_M^\alpha}{\partial T}\right)_{P,y_{js}}$ is calculated for fixed constituents fractions and the sum over the derivatives with respect to y_{js} takes care of the contribution from the variation of the constitution with the temperature.

The heat capacity finally is defined and calculated as

$$C_P = \left(\frac{\partial H}{\partial T}\right)_{P,N_i} \tag{111}$$

$$\left(\frac{\partial H}{\partial T}\right)_{P,N_i} = \sum_\alpha \frac{\partial \aleph^\alpha}{\partial T} H_M^\alpha + \sum_\alpha \aleph^\alpha \left(\frac{\partial H_M^\alpha}{\partial T}\right)_{P,N_i} \tag{112}$$

Again changing the derivative from constant N_i to constant y_{is} and replacing H by G gives:

$$
\begin{aligned}
\left(\frac{\partial H}{\partial T}\right)_{P,N_i} &= \sum_\alpha \frac{\partial \aleph^\alpha}{\partial T} H_M^\alpha + \sum_\alpha \aleph^\alpha \frac{\partial}{\partial T}\left[G_M^\alpha - T\left(\frac{\partial G_M^\alpha}{\partial T}\right)_{P,N_I}\right] \\
&= \sum_\alpha \frac{\partial \aleph^\alpha}{\partial T} H_M^\alpha - T\sum_\alpha \aleph^\alpha \frac{\partial}{\partial T}\left[\left(\frac{\partial G_M^\alpha}{\partial T}\right)_{P,N_i}\right] \\
&= \sum_\alpha \frac{\partial \aleph^\alpha}{\partial T} H_M^\alpha - T\sum_\alpha \aleph^\alpha \frac{\partial}{\partial T}\left[\left(\frac{\partial G_M^\alpha}{\partial T}\right)_{P,y_{js}} + \sum_j \left(\frac{\partial G_M^\alpha}{\partial y_{js}}\right)_{P,y_{k\neq j}} \frac{\partial y_{js}}{\partial T}\right] \\
&= \sum_\alpha \frac{\partial \aleph^\alpha}{\partial T} H_M^\alpha - T\sum_\alpha \aleph^\alpha \left[\left(\frac{\partial^2 G_M^\alpha}{\partial T^2}\right)_{P,y_{js}} + \sum_j \left(\frac{\partial^2 G_M^\alpha}{\partial T \partial y_{js}}\right)_{P,y_{k\neq j}} \frac{\partial y_{js}}{\partial T}\right]
\end{aligned}
\tag{113}
$$

For a small change in T, we replace the derivatives of $\aleph$ and y_{js} by finite differences. We also drop the subscript s as the summation over j is for all constituents.

$$
\left(\frac{\partial H}{\partial T}\right)_{P,N_i} = \sum_\alpha \Delta\aleph^\alpha H_M^\alpha - T\sum_\alpha \aleph^\alpha \left[\left(\frac{\partial^2 G_M^\alpha}{\partial T^2}\right)_{P,y_j} + \sum_j \left(\frac{\partial^2 G_M^\alpha}{\partial T \partial y_j}\right)_{P,y_{k\neq j}} \Delta y_j\right]
\tag{114}
$$

In order to find the appropriate values for $\Delta\aleph$ and Δy_j we have to calculate one iteration of the equilibrium matrix with an extra equation for ΔT as an additional variable. In order to be able to add such an equation the T must be a condition in the original equilibrium calculation.

As an example take the equilibrium matrix for a binary system with conditions on T, P and the amount of the components and with a single stable phase in eq. 52.

In this matrix we can add a line representing the variable ΔT and for all previous lines in the matrix we add a column representing the derivate with respect to T. The system equation for the same case would be:

$$\begin{pmatrix} M_{\mathrm{A}} & M_{\mathrm{B}} & 0 & s_{14} \\ \aleph \sum_i \frac{\partial M_{\mathrm{A}}}{\partial y_i} c_{i\mathrm{A}} & \aleph \sum_i \frac{\partial M_{\mathrm{A}}}{\partial y_i} c_{i\mathrm{B}} & M_{\mathrm{A}} & s_{24} \\ \aleph \sum_i \frac{\partial M_{\mathrm{B}}}{\partial y_i} c_{i\mathrm{A}} & \aleph \sum_i \frac{\partial M_{\mathrm{B}}}{\partial y_i} c_{i\mathrm{B}} & M_{\mathrm{B}} & s_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta\mu_{\mathrm{A}} \\ \Delta\mu_{\mathrm{B}} \\ \Delta\aleph \\ \Delta T \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (115)$$

with the terms in 4'th column calculated as:

$$s_{14} = \frac{\partial G_M}{\partial T} \tag{116}$$

$$s_{24} = \aleph \sum_i \frac{\partial M_{\mathrm{A}}}{\partial y_i} c_{iT} \tag{117}$$

$$s_{34} = \aleph \sum_i \frac{\partial M_{\mathrm{B}}}{\partial y_i} c_{iT} \tag{118}$$

The right hand side is all zeros, except for ΔT, as we are calculating a difference.

We do not iterate this system of equations, just solve it once to obtain the values of $\Delta\mu_{\mathrm{A}}, \Delta\mu_{\mathrm{A}}$ and $\Delta\aleph$, the value of ΔT is unity of course. The values of $\Delta\mu_{\mathrm{A}}$ and $\Delta\mu_{\mathrm{B}}$ are related to this change and the value of Δy_i can be calculated from eq. 36 using these $\Delta\mu_{\mathrm{A}}$ and $\Delta\mu_{\mathrm{B}}$:

$$\Delta y_i = c_{iT}\Delta T + \sum_{\mathrm{A}} c_{i\mathrm{A}} \, \Delta\mu_{\mathrm{A}} \tag{119}$$

In this way we obtain a heat capacity that correspond to the enthalpy change with an increase in of T with one degree. This includes also latent heat if there are several phases stable and contributions from internal degrees of ordering.

6.2 The liquidus slope

Another property that is useful for simulating phase transformations is the slope of the solubility lines. In order to calculate the slope of for example the liquidus versus T in the direction of a component A we can set the appropriate conditions together with specifying a solid phase as fix, and no condition on the temperature, and calculate the equilibrium.

References

[09Flo] C A Floudas and C E Gounaris, J of Global Optimization **45** (2009) 3-38

[75Eri] G Eriksson (1974)

[81Sun] B Sundman and J ï¿½gren (1981)

[81Hil] M Hillert, Physica (1981)

[82Luk] H L Lukas et al, Calphad (1982)

[84Jan] B Jansson, Thesis, KTH (1984)

[84Hil] M Hillert et al, Ionic liquid model (1984)

[01Hil] M Hillert, CEF

[15Sun1] B Sundman, U R Kattner, M Palumbo, S G Fries, IMMI journal (2015) 4:1

[15Sun2] B Sundman, X-G Lu, H Ohtani, Comp Mat. Sci (2015)

7 Software documentation

A very brief explanation of the main parts of the data structures and subroutines in the HMS package is given here.

7.1 Data structures

For each equilibrium calculation a structure called meqrec is created of the type meq_data explained below. In meqrec there is an array with record type meq_phase for each phase to hold intermediate values.

7.1.1 Data for each phase

For each phase in the system a structure with these data is created

```
  TYPE meq_phase
! parts of the data in this structure should be in the gtp_equilibrium_data
! it contains phase specific results from various subroutines during
! equilibrium calculation
! iph: phase number
! ics: composition set number
! idim: the dimension of phase matrix,
! ncc: the number of constituents
! stable: is 1 for a stable phase
! xdone: set to 1 for stoichiometric phases after calculating xmol first time
! dormlink: used to link phases that temporarily been set dormant
      integer iph,ics,idim,stable,ncc,xdone,dormlink
! value of phase status (-1,0=ent, 1=stable, 2=fix, -2=dorm, -3=sus, -4 hidden)
      integer phasestatus
! inverted phase matrix
      double precision, dimension(:,:), allocatable :: invmat
! mole fractions of components and their sum
      double precision, dimension(:), allocatable :: xmol
      double precision :: sumxmol,sumwmol
! Derivatives of moles of component wrt all constituent fractions of the phase
      double precision, dimension(:,:), allocatable :: dxmol
! link to phase_varres record
      TYPE(gtp_phase_varres), pointer :: curd
! value of amount and driving force at previous iteration
      double precision prevam, prevdg
! iteration when phase was added/removed
      integer itadd, itrem
! chargebal is 1 if external charge balance needed, ionliq<0 unless
! ionic liquid when it is equal to nkl(1)=number of cations
      integer chargebal,ionliq,i2sly(2)
      double precision charge,yva
! end specific ionic liquids
```

```
   end TYPE meq_phase
```

7.1.2   Data for the system

The equilibrium calculation starts by filling this structure with data. Each phase that can be stable has an entry in the meq_phase array.

```
  TYPE meq_setup
! one structure of this type is created when an equilibrium calculation
! is started and it holds all global data needed for handling the
! calculation of an equilibrium.  The phase specific data is in meq_phase
! nv: initial guess of number of stable phases
! nphase: total number of phases and composition sets
! nstph: current number of stable phases
! dormlink: is start of list of phases temporarily set dormant
! noofits current number of iterations
! nrel number of elements (omponents)
! typesofcond: types of conditions, =1 only massbal, =2 any conditions
! nfixmu number of fixed chemical potentials
! nfixph number of conditions representing fix phases
      integer nv,nphase,nstph,dormlink,noofits
      integer nrel,typesofcond,maxsph,nfixmu,nfixph
! component numbers of fixed potentials, reference and value
      integer, dimension(:), allocatable :: mufixel
      integer, dimension(:), allocatable :: mufixref
      double precision, dimension(:), allocatable :: mufixval
! fix phases and amounts
      integer, dimension(:,:), allocatable :: fixph
      double precision, dimension(:), allocatable :: fixpham
! fix phases during mapping: phase index and composition set index, zero amount
!      type(gtp_phasetuple), dimension(:), allocatable :: nodefixph
! indices of axis conditions that has been inactivated
!      integer, dimension(:), allocatable :: inactiveaxis
! iphl, icsl: phase and composition sets of intial guess of stable phases
! aphl: initial guess of amount of each stable phase
      integer iphl(maxel+2),icsl(maxel+2)
      double precision aphl(maxel+2)
! this is because I tried to scale the total amount of phases during iterations
      double precision antot
! stphl: current list of stable phases, value is index in phr array
      integer, dimension(maxel+2) :: stphl
! current values of chemical potentials stored in gtp_equilibrium_data
! if variable T and P these are TRUE, otherwise FALSE
      logical tpindep(2)
! these are the maximum allowed changes in T and P during iterations
      double precision tpmaxdelta(2)
```

```
! individual phase information
    type(meq_phase), dimension(:), allocatable :: phr
! information about conditions should be stored here.  Note that conditions
! may change during STEP and MAP
  end TYPE meq_setup
```

7.1.3   Data needed for applications like STEP and MAP

When calculating phase diagram and property diagrams one must have control of when
a new phase wants to be stable or a stable phase will dissapear. This data structure
helps to transfer such information and in addition is used to specify the phase kept fix
with zero amount along the lines in the phase diagram.

```
  TYPE map_fixph
! provides information about phase sets for each line during mapping
    integer nfixph,nstabph
    type(gtp_phasetuple), dimension(:), allocatable :: fixph
    type(gtp_phasetuple), dimension(:), allocatable :: stableph
    double precision, dimension(:), allocatable :: stablepham
  end TYPE map_fixph
```

7.2   Top level calculation routines

There are two routines for equilibrium calculation depening on the context.

7.2.1   The simplest one, for single equilibrium calculation

This subroutine organizes the calculation of a single equilibrium. The user must have
stored all data in the GTP package including all conditions. This in found via the
pointer ceq. Each such datastructure is independent and it should be possible to run
several of these in parallell.

   If mode is nonzero it will call a grid minimizer if possible to find a set of stable
phases. It does some clean up and copy of results at the end if the calculation is
successful.

```
  subroutine calceq2(mode,ceq)
! calculates the equilibrium for the given set of conditions
! mode=0 means no global minimization
! ceq is a datastructure with all relevant thermodynamic data
    implicit none
    integer mode
    TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.2.2   Equilibrium calculatins during step/map calculations

Calling this routine makes it possible to retrieve more information when the set of
stable phases changes etc. It is used in STEP/MAP when creating new lines in a
diagram.

```
  subroutine calceq7(mode,meqrec,mapfix,ceq)
! calculates the equilibrium for the given set of conditions
! mode=0 means no global minimization
! mode=-1 means used during step/map, no gridmin and do not deallocate phr
! ceq is a datastructure with all relevant thermodynamic data
! calling this routine instead of calceq2 makes it possible to extract
! additional information about the equilibrium from meqrec.
! Meqrec is also used for calculation of derivatives of state vatiables
     implicit none
     integer mode
     TYPE(meq_setup), pointer :: meqrec
     type(map_fixph), pointer :: mapfix
     TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.3   Subroutine to change the set of stable phases

This subroutine will call meq_sameset with a list of stable phases. The meq_sameset
subroutine will return to this if there is a phase with negative amount to be removed
or a phase with positive driving force to be added.

```
  subroutine meq_phaseset(meqrec,formap,ceq)
! this subroutine can change the set of stable phase and their amounts
! and constitutions until equilibrium is found for the current conditions.
! nv is the number of stable phases initially, iphl, icsl and aphl are the
! phase number, compset number and amount.
     implicit none
     TYPE(meq_setup) :: meqrec
     TYPE(gtp_equilibrium_data), pointer :: ceq
     logical formap
```

7.4   Iterations with same set of stable phases

This subroutine will iterate updating the chemical potentials, the phase amounts and
constitution of all phases as long as the set of stable phases does not change. Several
criteria is tested if the calculation have converged.

```
  subroutine meq_sameset(irem,iadd,meqrec,phr,ceq)
! iterate until phase set change, converged or error (incl too many its)
! iadd = -1 indicates called from calculating a sequence of equilibria
     implicit none
     integer irem,iadd
     TYPE(meq_setup) :: meqrec
     TYPE(meq_phase), dimension(*), target :: phr
     TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.5 Formulating the equilibrium matrix

This routine sets up the equilibrium matrix depending on the set of stable phases and the conditions set by the user.

```
  subroutine setup_equilmatrix(meqrec,phr,nz1,smat,tcol,pcol,&
      dncol,converged,ceq)
! handels external conditions on extensive variables in the equil matrix
! meqrec and phr contains data for phases, nz1 is dimension of equlibrium
! matrix, smat is the equilibrium matrix, tcol and pcol are columns for
! variable T or P, dncol is the column with phase amount variables.
! converged is used to indicate calling routine and set if not converged
! external variable.
    TYPE(meq_setup) :: meqrec
    TYPE(meq_phase), dimension(*), target :: phr
    double precision smat(nz1,nz1+1)
    integer nz1,tcol,pcol,converged,dncol
    TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.6 Routine to calculate the inverse phase matrix

At each iteration the inverted phase matrix is calculated for all phases. Additionally a lot of data needed for the massbalance equations are calculated and stored here.

```
  subroutine meq_onephase(meqrec,pmi,ceq)
! this subroutine calculates new constituent fractions for a phase iph+ics
! with given T, P and chemical potentials for the components
! For ionic liquids the sites on the sublattices varies with composition
! THIS IS A FIRST VERSION WITHOUT ANY TRICKS FOR SPEED
    implicit none
    TYPE(meq_phase), pointer :: pmi
    TYPE(gtp_equilibrium_data), pointer :: ceq
    TYPE(meq_setup) :: meqrec
```

7.7 Some utility routines

These are called when necessary for each iteration.

7.7.1 Correction of of second derivatives for the ionic liquid model

In the ionic liquid model the size of the formula unit of the phase varies with the composition. This requires some extra care when calculating how the amount of the components varies with the constitution.

```
  subroutine corriliq_d2gdyidyj(nkl,knr,curmu,pmi,ncc,pmat,ceq)
! correction of d2G/dy1dy2 for ionic liquid because the formula unit is
! not fixed.  This contributes ONLY to the second derivaties of G and
```

```
! is not really part of the model itself, only needed when minimizing G
    implicit none
    type(gtp_equilibrium_data), pointer :: ceq
    TYPE(meq_phase), pointer :: pmi
    integer ncc,nkl(*),knr(*)
    double precision curmu(*),pmat(ncc,*)
```

7.7.2 Test of same composition

When there are two or more composition sets of the same phase needed to handle
miscibility gaps, these may sometimes converge to the same composition and try to
become stable. This subroutine is needed to prevent having two composition sets with
the same composition stable.

```
  logical function same_composition(jj,phr,meqrec,ceq,dgm)
! returns .TRUE. if phase phr(jj) has almost exactly the same composition
! as another composition set of the same phase that is stable
! dgm just for debug output
! ==============================================================
! The composition of the phases are compared as ordered phases one can have
! the same constitution but distributed on different sets of sublattices ....
! ==============================================================
    implicit none
    integer jj
    double precision dgm
    TYPE(meq_phase), dimension(*) :: phr
    TYPE(meq_setup) :: meqrec
    TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.8 The coefficients in the Δy equation

These subroutines returnes the terms in the Δy expression, eq. 36. The first one
is used during the equilibrium calculation, the second is used when calculating "dot
derivatives".

```
  subroutine calc_dgdyterms1(nrel,ia,tpindep,mamu,mag,mat,map,pmi,&
      curmux,noofits)
! any change must also be made in subroutine calc_dyterms2
! calculate the terms in the deltay expression for amounts of component ia
!
! DM_A = \sum_B mu_B*MAMU(B) - MAG - MAT*dt - MAP*dp
!
! where MAMU=\sum_i dM_A/dy_i*\sum_j invmat(i,j)*dM_B/dy_j
!       c_iB=\sum_j invmat(i,j)*dM_B/dy_j etc etc
!
! it may not be very efficient but first get it right ....
! tpindep(1) is TRUE if T variable, tpindep(2) is TRUE if P are variable
```

```
      implicit none
      integer ia,nrel,noofits
      logical tpindep(2)
      double precision, dimension(*) :: mamu
      double precision mag,mat,map
      double precision curmux(*)
! pmi is the phase data record for this phase
      type(meq_phase), pointer :: pmi
    subroutine calc_dgdyterms2(iy,nrel,mamu,mag,mat,map,pmi)
! it should be similare to calc_dgdyterms1
      implicit none
      integer iy,nrel
      double precision mag,mat,map,mamu(*)
      type(meq_phase), pointer :: pmi
```

7.9   The remaining subroutines deals with state variable function and in particular dot derivatives

The dot derivatives requres second derivatives of the Gibbs energy from the most recent calculation. These are saved in the ceq record but it also requires the equilibrium matrix to know the current set of conditions. Thus some of the routines originally in the thermodynamic model package had to be moved here.

The names of the subroutines are quite confusing ...

7.9.1   Evaluate all state variable functions

This is called from the user interface to calculate values of all entered state variable functions. It writes the funtions and values on unit kou.

```
 subroutine meq_evaluate_all_svfun(kou,ceq)
! evaluate and list values of all state variable functions
    implicit none
    integer kou
    TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.9.2   Get the value of one or more state variable or function

This is called whenever the value of a state variable or a state variable function is needed. In particular the post processor will call this to obtain values to be plotted.

```
 subroutine meq_get_state_varorfun_value(statevar,value,dummy,ceq)
! used in OCPLOT to extact value of state variable of symbol
! NOTE if a specific function is given only this function evaluated
    implicit none
    character statevar*(*),dummy*(*)
    double precision value
    TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.9.3 Evaluate a state variable function

This function returns the value of a state variable function, The argument lrot is the index where the function is stored. The feature of having formal argument to state variable function has not been implemented.

```
 double precision function meq_evaluate_svfun(lrot,actual_arg,mode,ceq)
! envaluate all funtions as they may depend on each other
! actual_arg are names of phases, components or species as @Pi, @Ci and @Si
! needed in some deferred formal parameters  (NOT IMPLEMENTED YET)
! if mode=1 always evaluate, if mode=0 several options
   implicit none
   integer lrot,mode
   character actual_arg(*)*(*)
   TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.9.4 Initiate the equilibrium matrix for a derivative calculation

When calculating a dot derivative this subroutine must be called to create a temporare equilibrium data structure.

```
   subroutine initiate_meqrec(svr,svar,meqrec,ceq)
! this is to setup data for a state var derivative calculation
! taken from the normal initialization of an equilibrium calculation
! it also solves a modified equil matrix once to get delta-amounts and mu
   TYPE(meq_setup), pointer :: meqrec
   TYPE(gtp_state_variable), pointer :: svr
   double precision, allocatable :: svar(:)
   TYPE(gtp_equilibrium_data), pointer :: ceq
```

7.9.5 Calculate the value of a state variable derivative

This calculates a derivative of state variable svr1 with respect to state variable svr2. At present svr2 must be T or P but the intention is to handle all kinds of derivatives. The values returned depend on the conditions set at the last calculated equilibrium.

```
   subroutine meq_state_var_value_derivative(svr1,svr2,value,ceq)
! calculates a state variable value derivative (in some cases)
! svr1 and svr2 identifies the state variables in (dstv1/dstv2)
! check that svr2 2 is a condition
! value is calculated value
! ceq is current equilibrium
!
   implicit none
   TYPE(gtp_equilibrium_data), pointer :: ceq
   TYPE(gtp_state_variable), pointer :: svr1,svr2
   double precision value
```

7.9.6   Calculate the value of a state variable derivative for a single phase

This is called by meq_state_var_value_derivative for each stable phase in the system.

```
  subroutine meq_calc_phase_derivative(svr1,svr2,meqrec,iph,iel,&
      svar,jj,value,ceq)
! Calculate contribution for one phase, one or all elements
! svr1 and svr2 identifies the state variables in (dstv1/dstv2)
! value is calculated value
! iph and iel indicate possible phase or element
! svar is solution to equil matrix, potentials and phase amounts
! jj is an attempt to index phases in svar, starting with 1
! ceq is current equilibrium
!
! THIS IS UNFINISHED can only handle H.T
!
   implicit none
   TYPE(gtp_equilibrium_data), pointer :: ceq
   TYPE(gtp_state_variable), target :: svr1,svr2
   TYPE(meq_setup), pointer :: meqrec
   integer iph,iel,jj
   double precision value,svar(*)
```

8   Summary

That's all.