

Documentation of the implementation of an algorithm to calculate thermodynamic equilibria in multi-component systems for flexible sets of conditions.

Bo Sundman

Gif sur Yvette, France

DRAFT Version December 21, 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 (OC) initiative. There is also documentation available for the model 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 called TQOC. Finally there is a user guide to the command oriented user interface to OC [OC-UI].

The current release of the OC software and its documentation is available at [opencalphad] and the last update at [github].

Contents

1	Introduction	4
2	The thermodynamic model	4
2.1	The formula unit of a phase	5
2.2	Differentials	6
2.3	Examples	6
2.3.1	A gas phase with H and O	6
2.3.2	A crystalline phase with substitutional and interstitial constituents	7
2.3.3	A crystalline phase with long range ordering	7
2.4	End-members of models	8
3	The Gibbs energy	8
3.1	The differential of the Gibbs energy	9
3.2	The partial Gibbs energy with for an end-member	9
3.3	The Gibbs-Duhem relation	10
4	Minimization with constraints	11
4.1	The constraints	11
4.2	The Lagrangian equation	11
4.3	The derivative of the Lagrangian with respect to phase amounts	12
4.4	The derivative of the Lagrangian with respect to constituent fractions .	12
5	Calculating the equilibrium	13
5.1	Step 0, Obtaining start values by grid minimizer	13
5.2	Step 1, the phase matrix	15
5.2.1	The phase matrix for a binary system	16
5.2.2	The general equation for the correction of constituent fractions .	17
5.3	Charge balance	17
5.4	Step 2, the external conditions	18
5.4.1	Condition on the amount of the components	18
5.4.2	Example: a binary system with a single stable phasse	19
5.4.3	Example: a binary system with two stable phases	20
5.4.4	Example: a binary system with unknown T and one stable phase prescribed.	20
5.4.5	Example: a binary system with one condition of a chemical po- tential	21
5.5	Condition on volume	21
5.6	Constant Gibbs energy or entropy	22
5.7	Heat balance calculation	24
5.8	Normallized state variables as conditions	27
5.9	Changing the set of stable phases	30
5.10	Potentials as conditions	30
5.11	Generallizing the equilibrium matrix	30
5.12	Special treatment of the ionic liquid model	31

6	Calculating derivatives using the result of an equilibrium calculation, the dot derivative	32
6.1	Calculating heat capacity	32
6.2	The liquidus slope	34
7	Assessment of model parameters	35
8	Software documentation	36
8.1	Data structures	36
8.1.1	Data for each phase	36
8.1.2	Data for the system	37
8.1.3	Data needed for applications like STEP and MAP	38
8.2	Top level calculation routines	38
8.2.1	The simplest one, for single equilibrium calculation	38
8.2.2	Equilibrium calculations during step/map calculations	39
8.3	Subroutine to change the set of stable phases	39
8.4	Iterations with same set of stable phases	39
8.5	Formulating the equilibrium matrix	40
8.6	Routine to calculate the inverse phase matrix	40
8.7	Some utility routines	40
8.7.1	Correction of second derivatives for the ionic liquid model	41
8.7.2	Test of same composition	41
8.8	The coefficients in the Δy equation	41
8.9	The remaining subroutines deals with state variable function and in particular dot derivatives	43
8.9.1	Evaluate all state variable functions	43
8.9.2	Get the value of one or more state variable or function	44
8.9.3	Evaluate a state variable function	44
8.9.4	Initiate the equilibrium matrix for a derivative calculation	44
8.9.5	Calculate the value of a state variable derivative	45
8.9.6	Calculate the value of a state variable derivative for a single phase	45
8.10	Subroutine called in assessments	45
8.11	Subroutine called for listing extra data	46
9	Summary	46

1 Introduction

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

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 simulations 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, amorphous and many different crystalline forms. In any system the Gibbs energy, G , is given at equilibrium by

$$G = \sum_{\alpha} \mathbb{N}^{\alpha} G_M^{\alpha}(T, P, Y) \quad (1)$$

where \mathbb{N}^{α} 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 depends 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

$$G_m = \frac{G}{N} \quad (2)$$

where N is the total number of moles of components. But in this paper G_M , with a capital M as subscript 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 handled by Lagrangian 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 includes special gases, ideal gases, substitutional regular solutions, interstitial solutions, sublattice models etc. Some additional explanations will be given for the ionic liquid model [84Hil].

2.1 The formula unit of a phase

For a phase modelled with CEF we can have sublattices and the ratio of these are denoted 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 to 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

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

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:

$$M^\alpha = \sum_A M_A^\alpha \quad (4)$$

and the mole fraction is

$$x_A^\alpha = \frac{M_A^\alpha}{M^\alpha} \quad (5)$$

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

$$N_A = \sum_\alpha \aleph^\alpha M_A^\alpha \quad (6)$$

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

$$N = \sum_A N_A = \sum_A \sum_\alpha \aleph^\alpha M_A^\alpha \quad (7)$$

and the mole fraction of A in the whole system is

$$x_A = \frac{N_A}{N} \quad (8)$$

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

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

where each partial derivative of M_A^α is

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

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 unit 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 vary 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 $(\text{Cr}, \text{Fe})_1(\text{C}, \text{N}, \text{Va})_3$, where Va denotes a vacancy or vacant site. The moles formula units of the elements are:

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

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 vary between 1 and 4. The mole fractions are:

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

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 solubilities, $(\text{Cr}, \text{Fe}, \text{Ni})_{10}(\text{Cr}, \text{Mo})_4(\text{Cr}, \text{Fe}, \text{Mo}, \text{Ni})_{16}$, the moles formula units are

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

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

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

The reason for this somewhat lengthy explanation is it is very common to make mistakes, 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 of 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 many different ways. One well known formula relates the Gibbs energy to the chemical potentials, μ_A , and the number of moles, N_A of the components:

$$G = \sum_A N_A \mu_A \quad (11)$$

The definition of the chemical potentials is

$$\mu_A = \left(\frac{\partial G}{\partial N_A} \right)_{T,P,N_{B \neq A}} \quad (12)$$

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:

$$G_M^\alpha = \sum_A M_A^\alpha \mu_A \quad (13)$$

$$G_m^\alpha = \sum_A x_A^\alpha \mu_A \quad (14)$$

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

$$dG = -SdT + VdP + \sum_A \mu_A dN_A \quad (15)$$

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

$$dG = \sum_\alpha (\aleph^\alpha dG_M^\alpha + G_M^\alpha d\aleph^\alpha) \quad (16)$$

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 :

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

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:

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

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 $(\text{Fe}, \text{Cr})_1(\text{C}, \text{N}, \text{Va})_3$ 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:

$$G_{\text{Fe:Va}} = G_M + \left(\frac{\partial G_M}{\partial y_{\text{Fe},1}} \right)_{T,P,y_j \neq \text{Fe}} + \left(\frac{\partial G_M}{\partial y_{\text{Va},2}} \right)_{T,P,y_j \neq \text{Va}} - \sum_s \sum_j y_{js} \left(\frac{\partial G_M}{\partial y_{js}} \right)_{T,P,y_k \neq j} \quad (19)$$

$$G_{\text{Fe:C}} = G_M + \left(\frac{\partial G_M}{\partial y_{\text{Fe},1}} \right)_{T,P,y_j \neq \text{Fe}} + \left(\frac{\partial G_M}{\partial y_{\text{C},2}} \right)_{T,P,y_j \neq \text{C}} - \sum_s \sum_j y_{js} \left(\frac{\partial G_M}{\partial y_{js}} \right)_{T,P,y_k \neq j} \quad (20)$$

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

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

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

$$\mu_{\text{C}} = G_{\text{C}} = \frac{1}{3}(G_{\text{Fe:C}} - G_{\text{Fe:Va}}) = \frac{1}{3} \left[\left(\frac{\partial G_M}{\partial y_{\text{C},2}} \right)_{T,P,y_{\text{Va}}} - \left(\frac{\partial G_M}{\partial y_{\text{Va},2}} \right)_{T,P,y_{\text{C}}} \right] \quad (21)$$

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 if 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 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:

$$\sum_{\text{A}} d\mu_{\text{A}} M_{\text{A}}^{\alpha} + S_M^{\alpha} dT + V_M^{\alpha} dP = 0 \quad (22)$$

which must be valid at equilibrium.

4 Minimization with constraints

To minimize a function with constraints we apply a Lagrangian equation where each equality constraints has a multiplier. When the constraint is obeyed the minimum of the Lagrangian is the same as the original 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:

$$g_s^\alpha = 1 - \sum_i y_{is}^\alpha = 0 \quad (23)$$

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

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

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^α , 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

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

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^α 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:

$$L = G + \sum_A f_A \mu_A + \sum_\alpha \eta_s^\alpha g_s^\alpha + \sum_\alpha \lambda^\alpha Q^\alpha + \sum_\psi \gamma^\psi \aleph^\psi \quad (26)$$

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 Lagrangian is that it will have the same extremum points as the Gibbs energy G when the constraints are fulfilled. 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:

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

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:

$$\frac{\partial L}{\partial \aleph^\psi} = G_M^\psi - \sum_i \mu_i M_i^\psi + \gamma^\psi = 0 \quad (28)$$

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 becoming positive.

On the other hand, if \aleph^α 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 5.9.

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:

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

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 independent variables dT, dP and dy_{jt}^α :

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

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 summation 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 :

$$\begin{aligned}
\sum_t \sum_j \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha \partial y_{jt}^\alpha} \right) \Delta y_{jt}^\alpha + \frac{\eta_s^\alpha}{N^\alpha} &= \\
\sum_A \mu_A \left(\frac{\partial M_A^\alpha}{\partial y_{is}^\alpha} \right) - \left(\frac{\partial G_M^\alpha}{\partial y_{is}^\alpha} \right) &- \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha \partial T} \right) \Delta T - \left(\frac{\partial^2 G_M^\alpha}{\partial y_{is}^\alpha \partial P} \right) \Delta P
\end{aligned} \tag{31}$$

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 interested to solve this for the fraction corrections Δy_i and can rearrange this in matrix notation, omitting the superscripts:

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

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 constituent fractions of each phase in the global potentials μ_A , ΔT 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 initial 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.

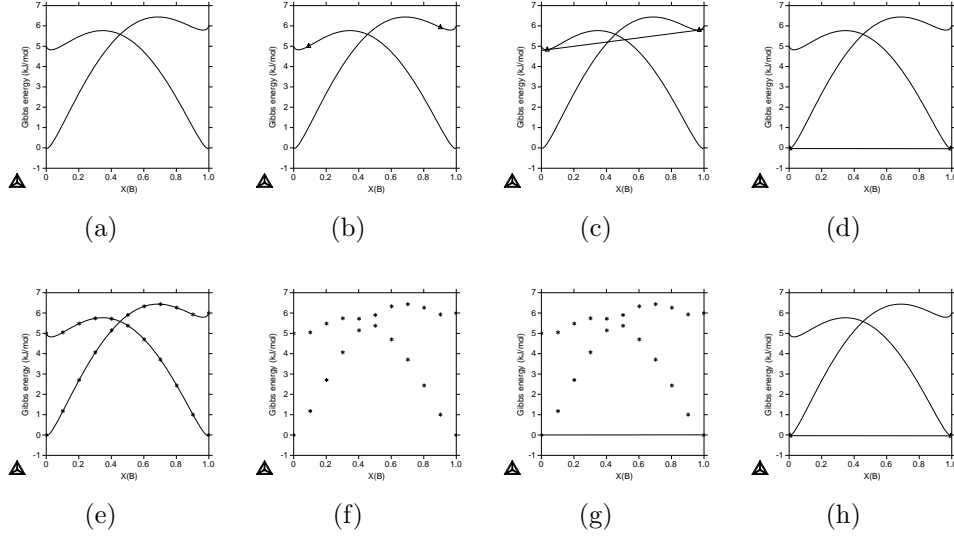


Figure 1: A case where start values matters. In (a) the Gibbs energy curves for two phases with miscibility gaps in a binary system are shown. If we have the initial constitutions marked in (b) the present algorithm will find the local equilibrium shown in (c) which is not the global minimum, the global 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, each 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 systems with solution phases by calculating a number of gridpoints in each phase and then treat each of these as a separate phase. 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 representing 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.

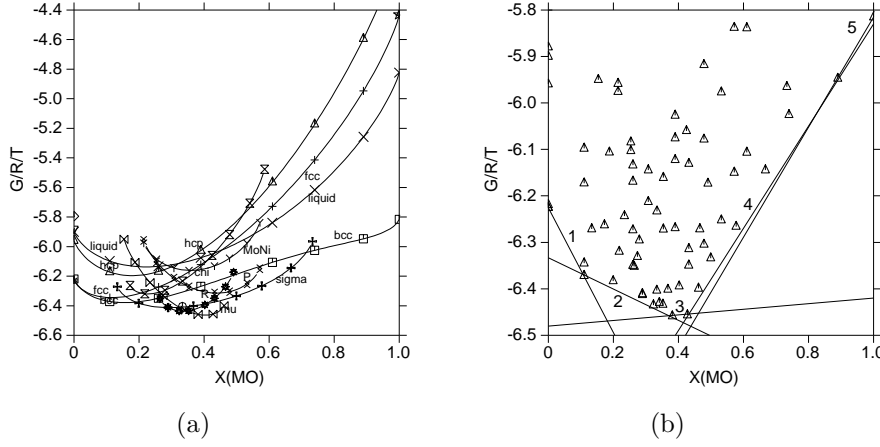


Figure 2: The Gibbs energy curves if the phases in the Fe-Mo system calculated at 1400 K in (a) together with the selected gridpoints. In (b) the gridpoints are treated as individual phases and the “convex hull” is drawn between the gridpoint pairs representing the lowest Gibbs energy at various compositions

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 generalised.

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

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

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:

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

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:

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

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:

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

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

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

$$\Delta y_{is} = c_{iG} + c_{iT}\Delta T + c_{iP}\Delta P + \sum_A c_{iA} \mu_A \quad (36)$$

where the coefficients in this equation can be calculated as

$$\begin{aligned} c_{iG} &= -\sum_j e_{ij} \frac{\partial G_M}{\partial y_j} \\ c_{iT} &= -\sum_j e_{ij} \frac{\partial^2 G_M}{\partial T \partial y_j} \\ c_{iP} &= -\sum_j e_{ij} \frac{\partial^2 G_M}{\partial P \partial y_j} \\ c_{iA} &= \sum_j e_{ij} \frac{\partial M_A}{\partial y_j} \end{aligned} \quad (37)$$

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 ΔT , ΔP 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 electrically neutral.

$$Q^\alpha = \sum_s a_s \sum_i \nu_i y_{is}^\alpha = 0 \quad (38)$$

$$\Delta Q^\alpha = \sum_s a_s \sum_i \frac{\partial Q}{\partial y_{is}^\alpha} \Delta y_{is}^\alpha \quad (39)$$

where

$$\frac{\partial Q^\alpha}{\partial y_{is}^\alpha} = a_s \nu_i \quad (40)$$

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

$$\Delta y_{is}^\alpha = c_{iG} + c_{iT}\Delta T + c_{iP}\Delta P + \sum_A c_{iA} \mu_A - e_{iQ}Q^\alpha \quad (41)$$

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 balance but we cannot use that the derivatives $\frac{\partial M_A}{\partial y_{is}}$ 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:

$$G_M^\alpha = \sum_A M_A^\alpha \mu_A \quad (42)$$

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:

$$G_M^\alpha = \sum_A M_A^\alpha \mu_A - \frac{\partial G_M}{\partial T} \Delta T - \frac{\partial G_M}{\partial P} \Delta P \quad (43)$$

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:

$$N_A - \tilde{N}_A = \sum_\alpha \aleph^\alpha M_A^\alpha - \tilde{N}_A = 0 \quad (44)$$

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

$$\Delta N_A = \sum_\alpha \aleph^\alpha \Delta M_A^\alpha + \sum_\alpha \Delta \aleph^\alpha M_A^\alpha = 0 \quad (45)$$

From eq. 9 we have:

$$\Delta M_A^\alpha = \sum_i \frac{\partial M_A^\alpha}{\partial y_i^\alpha} \Delta y_i^\alpha \quad (46)$$

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

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

5.4.2 Example: a binary system with a single stable phase

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:

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

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

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

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 $\Delta \aleph$ 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 sufficiently small. The matrix in eq. 52 is called the *equilibrium matrix*.

Note that unstable phases also will have their constitution updated 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^α 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 5.9.

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):

$$\begin{pmatrix} M_A^\alpha & M_B^\alpha & 0 & 0 \\ M_A^\beta & M_B^\beta & 0 & 0 \\ \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} c_{iA}^\varphi & \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} c_{iB}^\varphi & M_A^\alpha & M_A^\beta \\ \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_B^\varphi}{\partial y_i^\varphi} c_{iA}^\varphi & \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_B^\varphi}{\partial y_i^\varphi} c_{iB}^\varphi & M_B^\alpha & M_B^\beta \end{pmatrix} \begin{pmatrix} \mu_A \\ \mu_B \\ \Delta \aleph^\alpha \\ \Delta \aleph^\beta \end{pmatrix} = \begin{pmatrix} G_M^\alpha \\ G_M^\beta \\ \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} c_{iG}^\varphi + N_A - \tilde{N}_A \\ \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_B^\varphi}{\partial y_i^\varphi} c_{iG}^\varphi + N_B - \tilde{N}_B \end{pmatrix} \quad (53)$$

Iteratively 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:

$$\begin{pmatrix} M_A^\alpha & M_B^\alpha & 0 & -\frac{\partial G_M^\alpha}{\partial T} \\ M_A^\beta & M_B^\beta & 0 & -\frac{\partial G_M^\beta}{\partial T} \\ \aleph^\alpha \sum_i \frac{\partial M_A^\alpha}{\partial y_i^\alpha} c_{iA}^\alpha & \aleph^\alpha \sum_i \frac{\partial M_A^\alpha}{\partial y_i^\alpha} c_{iB}^\alpha & M_A^\alpha & -\sum_i \frac{\partial M_A^\alpha}{\partial y_i^\alpha} c_{iT}^\alpha \\ \aleph^\alpha \sum_i \frac{\partial M_B^\alpha}{\partial y_i^\alpha} c_{iA}^\alpha & \aleph^\alpha \sum_i \frac{\partial M_B^\alpha}{\partial y_i^\alpha} c_{iB}^\alpha & M_B^\alpha & -\sum_i \frac{\partial M_B^\alpha}{\partial y_i^\alpha} c_{iT}^\alpha \end{pmatrix} \begin{pmatrix} \mu_A \\ \mu_B \\ \Delta \aleph^\alpha \\ \Delta T \end{pmatrix} = \begin{pmatrix} G_M^\alpha \\ G_M^\beta \\ \aleph^\alpha \sum_i \frac{\partial M_A^\alpha}{\partial y_i^\alpha} c_{iG}^\alpha + N_A - \tilde{N}_A \\ \aleph^\alpha \sum_i \frac{\partial M_B^\alpha}{\partial y_i^\alpha} c_{iG}^\alpha + N_B - \tilde{N}_B \end{pmatrix} \quad (54)$$

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.

$$\begin{pmatrix} M_A^\alpha & 0 & 0 \\ M_A^\beta & 0 & 0 \\ \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} c_{iA}^\varphi & M_A^\alpha & M_A^\beta \end{pmatrix} \begin{pmatrix} \mu_A \\ \Delta \aleph^\alpha \\ \Delta \aleph^\beta \end{pmatrix} = \begin{pmatrix} G_M^\alpha - M_B^\alpha \mu_B \\ G_M^\beta - M_B^\beta \mu_B \\ \sum_\varphi \aleph^\varphi \sum_i \frac{\partial M_A^\varphi}{\partial y_i^\varphi} (c_{iG}^\varphi - c_{iB}^\varphi \mu_B) + N_A - \tilde{N}_A \end{pmatrix} \quad (55)$$

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:

$$dV = V - \tilde{V} = 0 \quad (56)$$

where

$$V = \sum_\alpha \aleph^\alpha V_M^\alpha \quad (57)$$

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

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 ΔT , ΔP and μ_A using eq. 36. Rearranging the equation for the equilibrium matrix for the unknown ΔT , ΔP and μ_A gives:

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

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 preparation 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:

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

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

$$dS = S - \tilde{S} = 0 \quad (66)$$

where

$$S = \sum_{\alpha} \aleph^{\alpha} S_M^{\alpha} \quad (67)$$

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

$$dS_M^{\alpha} = - \sum_A dM_A^{\alpha} \frac{\partial \mu_A}{\partial T} \quad (69)$$

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

$$dS_M^{\alpha} = - \frac{\partial^2 G_M^{\alpha}}{\partial T^2} \Delta T - \frac{\partial^2 G_M^{\alpha}}{\partial T \partial P} \Delta P - \sum_i \frac{\partial^2 G_M^{\alpha}}{\partial T \partial y_i} \Delta y_i \quad (70)$$

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

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

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

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

This means we should not use eq. 65 for a condition on G, but write the differential of G_M^{α} as:

$$dG_M^{\alpha} = \frac{\partial G_M^{\alpha}}{\partial T} \Delta T + \frac{\partial G_M^{\alpha}}{\partial P} \Delta P + \sum_i \frac{\partial G_M^{\alpha}}{\partial y_i} \Delta y_i \quad (73)$$

and the equation for a constant Gibbs energy condition would be written

$$\begin{aligned} dG &= \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial G_M^{\alpha}}{\partial T} \Delta T + \frac{\partial G_M^{\alpha}}{\partial P} \Delta P + \sum_i \frac{\partial G_M^{\alpha}}{\partial y_i} \Delta y_i \right) + \sum_{\alpha} G_M^{\alpha} \Delta \aleph^{\alpha} \\ &= G - \tilde{G} = 0 \end{aligned} \quad (74)$$

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

$$\begin{aligned} \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial G_M^{\alpha}}{\partial T} + \sum_i \frac{\partial G_M^{\alpha}}{\partial y_i} c_{iT} \right) \Delta T + \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial G_M^{\alpha}}{\partial P} + \sum_i \frac{\partial G_M^{\alpha}}{\partial y_i} c_{iP} \right) \Delta P \\ + \sum_{\alpha} \aleph^{\alpha} \sum_A \sum_i \frac{\partial G_M^{\alpha}}{\partial y_i} c_{iA} \mu_A + \sum_{\alpha} G_M^{\alpha} \Delta \aleph^{\alpha} = - \sum_{\alpha} \aleph^{\alpha} \sum_i \frac{\partial G_M^{\alpha}}{\partial y_i} c_{iG} + G - \tilde{G} \end{aligned} \quad (75)$$

Just to remind you

$$c_{iA} = \sum_j e_{ij} \frac{\partial M_A}{\partial y_j} \quad (76)$$

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

$$mamu(A) = \sum_i F_i c_{iA} = \sum_i \sum_j F_i e_{ij} \frac{\partial M_A}{\partial y_j} \quad (77)$$

$$maT = \sum_i F_i c_{iT} = \sum_i \sum_j F_i e_{ij} \frac{\partial^2 G_M}{\partial T \partial y_j} \quad (78)$$

$$maP = \sum_i F_i c_{iP} = \sum_i \sum_j F_i e_{ij} \frac{\partial^2 G_M}{\partial P \partial y_j} \quad (79)$$

$$maG = \sum_i F_i c_{iG} = \sum_i \sum_j F_i e_{ij} \frac{\partial G_M}{\partial y_j} \quad (80)$$

where F_i is an array passed to this subroutine and $mamu(A)$ is an array with one value for each component. 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 a mass balance equation on the amount of component B, N_B , it is $F_i = \frac{\partial M_B}{\partial y_i}$.

5.7 Heat balance calculation

The enthalpy is $H = G + TS$ and we are frequently interested to calculate the equilibrium at constant enthalpy. We have as before the equation:

$$dH = H - \tilde{H} = 0 \quad (81)$$

where \tilde{H} is the prescribed value of the enthalpy and

$$H = \sum_{\alpha} \aleph^{\alpha} H_M^{\alpha} \quad (82)$$

$$dH = \sum_{\alpha} \aleph^{\alpha} dH_M^{\alpha} + \sum_{\alpha} H_M^{\alpha} d\aleph^{\alpha} \quad (83)$$

$$H_M^{\alpha} = G_M^{\alpha} + TS_M^{\alpha} \quad (84)$$

$$dH_M^{\alpha} = dG_M^{\alpha} + TdS_M^{\alpha} + S_M^{\alpha}dT \quad (85)$$

Thermodynamics is confusing, maybe we should use:

$$dH_M^{\alpha} = dG_M^{\alpha} + TdS_M^{\alpha} \quad (86)$$

as we use such an equation together with $\Delta G = 0$ when we have a phase transformation to obtain $\Delta H = T\Delta S$. But this relation is only valid at fixed T , i.e. $dT = 0$. In the general case we must also include dT .

In eq. 85 the differentials for G_M and S_M are given by eqs 70 and 73 respectively for a single phase. Combining there we have:

$$\begin{aligned} dH_M^{\alpha} = & \left(\frac{\partial G_M}{\partial T} - T \frac{\partial^2 G_M}{\partial T^2} \right) \Delta T + \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} \right) \Delta P + \\ & \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) \Delta y_i - \frac{\partial G_M}{\partial T} \Delta T \end{aligned} \quad (87)$$

Summed over all phases and including the change in the phase amount, $\left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^{\alpha}$, this gives

$$\begin{aligned} \Delta H = & \sum_{\alpha} \aleph^{\alpha} \left[\left(\frac{\partial G_M}{\partial T} - T \frac{\partial^2 G_M}{\partial T^2} \right) \Delta T + \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} \right) \Delta P + \right. \\ & \left. \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) \Delta y_i - \frac{\partial G_M}{\partial T} \Delta T \right] + \\ & \sum_{\alpha} \left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^{\alpha} = H - \tilde{H} = 0 \end{aligned} \quad (88)$$

where we calculate the partial derivatives with the appropriate variables T, P and y_i kept fixed. We can do this for a single phase as we have already separated out the change in the amount of the phases, $\Delta \aleph^{\alpha}$.

Realizing that the two terms $\frac{\partial G}{\partial T}$ cancel and inserting the expression for Δy_i from eq. 36 we get:

$$\begin{aligned} & \sum_{\alpha} \aleph^{\alpha} \left[\left(-T \frac{\partial^2 G_M}{\partial T^2} \right) \Delta T + \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} \right) \Delta P + \right. \\ & \left. \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) (c_{iG} + c_{iT} \Delta T + c_{iP} \Delta P + \sum_A c_{iA} \mu_A) \right] + \\ & \sum_{\alpha} \left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^{\alpha} = H - \tilde{H} \end{aligned} \quad (89)$$

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 :

$$\begin{aligned}
& \sum_{\alpha} \aleph^{\alpha} \left(-T \frac{\partial^2 G_M}{\partial T^2} + \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iT} \right) \Delta T + \\
& \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} + \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iP} \right) \Delta P + \\
& \sum_{\alpha} \aleph^{\alpha} \sum_A \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iA} \mu_A + \sum_{\alpha} \left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^{\alpha} = \\
& - \sum_{\alpha} \aleph^{\alpha} \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iG} + H - \tilde{H} \quad (90)
\end{aligned}$$

and if we prescribed the enthalpy of a specific phase α , \tilde{H}^α :

$$\begin{aligned}
& \left(-T \frac{\partial^2 G_M}{\partial T^2} + \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iT} \right) \Delta T + \\
& \left(\frac{\partial G_M}{\partial P} - T \frac{\partial^2 G_M}{\partial T \partial P} + \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iP} \right) \Delta P + \\
& \sum_A \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iA} \mu_A + \left(G_M - T \frac{\partial G_M}{\partial T} \right) \Delta \aleph^{\alpha} = \\
& - \sum_i \left(\frac{\partial G_M}{\partial y_i} - T \frac{\partial^2 G_M}{\partial T \partial y_i} \right) c_{iG} + H^{\alpha} - \tilde{H}^{\alpha} \quad (91)
\end{aligned}$$

Note that the enthalpy of the α phase in eq. 91 depends on the amount of the phase. If it is not stable H^{α} will be zero.

If we prescribe the volume of a system, \tilde{V} , we use

$$V = \left(\frac{\partial G_M}{\partial P} \right)_{T, y_i} \quad (92)$$

$$\Delta V = \sum_i \frac{\partial^2 G_M}{\partial P \partial y_i} \Delta y_i + \frac{\partial^2 G_M}{\partial P \partial T} \Delta T + \frac{\partial^2 G_M}{\partial P^2} \Delta P \quad (93)$$

and eq. 90 can be transformed to:

$$\begin{aligned}
& \sum_{\alpha} \aleph^{\alpha} \sum_A \sum_i \frac{\partial^2 G_M}{\partial P \partial y_i} c_{iA} \mu_A + \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial^2 G_M}{\partial T \partial P} + \sum_i \frac{\partial^2 G_M}{\partial P \partial y_i} c_{iT} \right) \Delta T + \\
& \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial^2 G_M}{\partial P^2} + \sum_i \frac{\partial^2 G_M}{\partial P \partial y_i} c_{iP} \right) \Delta P + \sum_{\alpha} \frac{\partial G_M}{\partial P} \Delta \aleph^{\alpha} = \\
& - \sum_{\alpha} \aleph^{\alpha} \sum_i \frac{\partial^2 G_M}{\partial P \partial y_i} c_{iG} + V - \tilde{V} \quad (94)
\end{aligned}$$

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. It makes it easy to set up equations for each equation independent of all other conditions. In section 5.8 we show how to handle normalized state variables as conditions.

5.8 Normalized state variables as conditions

In most of the examples above the total amounts of the components has been used as condition. This simplifies the equations because if we use normalized state variables like mole fractions rather than total amounts we must take into account the derivatives of the normalizing 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 normalized one (normalized per moles, mass or volume). For a global total property we can formulate the condition like for N_A in eq. 44.

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

$$z^\alpha = \frac{Z_M^\alpha}{K_M^\alpha} \quad (95)$$

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

$$\Delta z^\alpha = \frac{\Delta Z_M^\alpha - z^\alpha \Delta K_M^\alpha}{K_M^\alpha} = z^\alpha - \tilde{z}^\alpha = 0 \quad (96)$$

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:

$$\Delta Z_M^\alpha = \frac{\partial Z_M^\alpha}{\partial T} \Delta T + \frac{\partial Z_M^\alpha}{\partial P} \Delta P + \sum_t \sum_j \frac{\partial Z_M^\alpha}{\partial y_{js}^\alpha} \Delta y_{js}^\alpha \quad (97)$$

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:

$$Z = \sum_\alpha \aleph^\alpha Z_M^\alpha \quad (98)$$

Using a normalizing factor K for the whole system gives z :

$$z = \frac{Z}{K} \quad (99)$$

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

$$\Delta z = \tilde{z} - z = \frac{1}{K} \left(\sum_\alpha \aleph^\alpha \Delta Z_M^\alpha + Z_M^\alpha \Delta \aleph^\alpha \right) - \frac{Z}{K^2} \left(\sum_\alpha \aleph^\alpha \Delta K_M^\alpha + K_M^\alpha \Delta \aleph^\alpha \right) \quad (100)$$

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. 97 and eq. 36. A phase with prescribed fixed amount have $\Delta \aleph = 0$.

A typical use of eq. 100 is when there is a condition on the mole or mass fractions of a component as shown in eq. 47.

Thermodynamic properties like V and H can also be normalized with respect to the amount of components N or the mass B with suffix M and W respectively. The enthalpy, H , can also be normalized with respect to the volume with suffix V . A property with a phase index can also be normalized with respect to the formula unit using the suffix F . Without suffix and phase index the value of the property is for the current size of the system. With a phase index and without suffix the value is for the current amount of the phase and if the phase is not stable that is zero.

As an example take the equation for normalizing the enthalpy per moles of component, H_m , where as before lower case m means per mole of components and upper case M means per mole formula unit:

$$Z = H = \sum_{\alpha} \aleph^{\alpha} H_M^{\alpha} \quad (101)$$

$$K = N = \sum_{\alpha} \aleph^{\alpha} M^{\alpha} \quad (102)$$

$$z = \frac{Z}{K} = H_m = \frac{H}{N} = \frac{\sum_{\alpha} \aleph^{\alpha} H^{\alpha}}{\sum_{\alpha} \aleph^{\alpha} M^{\alpha}} \quad (103)$$

$$H_M^{\alpha} = G_M^{\alpha} - T \left(\frac{\partial G_M^{\alpha}}{\partial T} \right)_{P, y_i} \quad (104)$$

$$M^{\alpha} = \sum_A \sum_s a_s^{\alpha} \sum_i b_{Ai} y_{is}^{\alpha} \quad (105)$$

where G_M^{α} is the Gibbs energy and M^{α} (according to eq. 4) is the amount of moles of components, in both cases per mole formula unit of phase α .

When used as condition we need the differential of this and for dH we use eq. 90:

$$\begin{aligned} dH = \Delta H = & \sum_{\alpha} \aleph^{\alpha} \sum_A \sum_i \frac{\partial H_M^{\alpha}}{\partial y_i} c_{iA} \mu_A + \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial H_M^{\alpha}}{\partial T} + \sum_i \frac{\partial H_M^{\alpha}}{\partial y_i} c_{iT} \right) \Delta T + \\ & \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial H_M^{\alpha}}{\partial P} + \sum_i \frac{\partial H_M^{\alpha}}{\partial y_i} c_{iP} \right) \Delta P + \sum_{\alpha} \aleph^{\alpha} \sum_i \frac{\partial H_M^{\alpha}}{\partial y_i} c_{iG} + \sum_{\alpha} H_M^{\alpha} \Delta \aleph^{\alpha} \end{aligned} \quad (106)$$

and for dN we follow (the slightly confusing notation) in eq. 47:

$$\begin{aligned} dN = \Delta N = & \sum_{\alpha} \aleph^{\alpha} \Delta M^{\alpha} = \sum_{\alpha} \sum_A M_A^{\alpha} \Delta \aleph^{\alpha} + \\ & \sum_{\alpha} \aleph^{\alpha} \left(\sum_A \sum_i \frac{\partial M_A^{\alpha}}{\partial y_i} c_{iA} \mu_A + \sum_i \frac{\partial M_A^{\alpha}}{\partial y_i} c_{iT} \Delta T + \sum_i \frac{\partial M_A^{\alpha}}{\partial y_i} c_{iP} \Delta P + \sum_i \frac{\partial M_A^{\alpha}}{\partial y_i} c_{iG} \right) \end{aligned} \quad (107)$$

Note that c_{iA}, c_{iT} etc. are of course different for each phase even if that is not indicated. This should be inserted in eq. 100 together with $z = \frac{H}{N}$:

$$\frac{1}{N} \sum_{\alpha} \aleph^{\alpha} \left(\sum_A \sum_i \frac{\partial H_M^{\alpha}}{\partial y_i} c_{iA} \mu_A + \left(\frac{\partial H_M^{\alpha}}{\partial T} + \sum_i \frac{\partial H_M^{\alpha}}{\partial y_i} c_{iT} \right) \Delta T + \right.$$

$$\begin{aligned}
& \left(\frac{\partial H_M^\alpha}{\partial P} + \sum_i \frac{\partial H_M^\alpha}{\partial y_i} c_{iP} \right) \Delta P + \sum_i \frac{\partial H_M^\alpha}{\partial y_i} c_{iG} \Bigg) + \\
& \frac{1}{N} \left(\sum_\alpha H_M^\alpha - \frac{H}{N} \sum_\alpha \sum_A M_A^\alpha \right) \Delta \aleph^\alpha + \\
& \frac{H}{N^2} \sum_\alpha \aleph^\alpha \left(\sum_A \sum_i \frac{\partial M_A^\alpha}{\partial y_i} c_{iA} \mu_A + \sum_i \frac{\partial M_A^\alpha}{\partial y_i} c_{iT} \Delta T + \right. \\
& \quad \left. \sum_i \frac{\partial M_A^\alpha}{\partial y_i} c_{iP} \Delta P + \sum_i \frac{\partial M_A^\alpha}{\partial y_i} c_{iG} \right) = H_m - \tilde{H}_m
\end{aligned} \tag{108}$$

Rearranging for the independent variables:

$$\begin{aligned}
& \frac{1}{N} \sum_\alpha \aleph^\alpha \left[\sum_A \sum_i \left(\frac{\partial H_M^\alpha}{\partial y_i} - \frac{H}{N} \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iA} \mu_A + \right. \\
& \quad \left(\frac{\partial H_M^\alpha}{\partial T} + \sum_i \left(\frac{\partial H_M^\alpha}{\partial y_i} - \frac{H}{N} \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iT} \right) \Delta T + \\
& \quad \left. \left(\frac{\partial H_M^\alpha}{\partial P} + \sum_i \left(\frac{\partial H_M^\alpha}{\partial y_i} - \frac{H}{N} \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iP} \right) \Delta P \right] + \\
& \quad \frac{1}{N} \sum_\alpha \left(H_M^\alpha - \frac{H}{N} \sum_A M_A^\alpha \right) \Delta \aleph^\alpha = -\frac{1}{N} \sum_\alpha \aleph^\alpha \sum_i \left(\frac{\partial H_M^\alpha}{\partial y_i} - \frac{H}{N} \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iG} + H_m - \tilde{H}_m
\end{aligned} \tag{109}$$

And finally replace $H_M = G_M - T \frac{\partial G}{\partial T}$ and use $H_m = \frac{H}{N}$:

$$\begin{aligned}
& \frac{1}{N} \sum_\alpha \aleph^\alpha \left[\sum_A \sum_i \left(\frac{\partial G_M^\alpha}{\partial y_i} - T \frac{\partial^2 G^\alpha}{\partial T \partial y_i} - H_m \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iA} \mu_A + \right. \\
& \quad \left(-T \frac{\partial^2 G^\alpha}{\partial T^2} + \sum_i \left(\frac{\partial G_M^\alpha}{\partial y_i} - T \frac{\partial^2 G}{\partial y_i \partial T} - H_m \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iT} \right) \Delta T + \\
& \quad \left. \left(\frac{\partial G_M^\alpha}{\partial P} - T \frac{\partial^2 G_M^\alpha}{\partial T \partial P} + \sum_i \left(\frac{\partial G_M^\alpha}{\partial y_i} - T \frac{\partial^2 G}{\partial y_i \partial T} - H_m \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iP} \right) \Delta P \right] + \\
& \quad \frac{1}{N} \sum_\alpha \left(G_M^\alpha - T \frac{\partial G_M^\alpha}{\partial T} - H_m \sum_A M_A^\alpha \right) \Delta \aleph^\alpha = \\
& \quad -\frac{1}{N} \sum_\alpha \aleph^\alpha \sum_i \left(\frac{\partial G_M^\alpha}{\partial y_i} - T \frac{\partial^2 G^\alpha}{\partial T \partial y_i} - H_m \frac{\partial M_A^\alpha}{\partial y_i} \right) c_{iG} + H_m - \tilde{H}_m
\end{aligned} \tag{110}$$

The condition can also be set for a specific phase. In this case the molar enthalpy, H_m , will be nonzero even if the phase is not stable. Similar equations can be derived for other normalizing properties.

All of this may seem quite complicated but the coding is general and it is an advantage that these equations make each condition independent of any other condition.

5.9 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.

$$\gamma^\psi = \sum_A \mu_A M_A^{\psi_i} - G_M^\psi \quad (111)$$

where the sum over A is for all components. If T and P are variable the terms with 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 cause problem is when more phases become stable than allowed by the Gibbs phase rule:

$$f = n - p + 2 \quad (112)$$

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 or we must allow T to vary in order to determine the invariant T .

5.10 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 because then we are trying to minimize the Gibbs-Duhem relation which is always zero. At least one condition must be an extensive property.

A potential set to a fixed value means we have one variable less in the equilibrium matrix. If there is a relation between two potentials that must be added as an equation.

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 during

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

$$(C^{+\nu_C})_P(D^{-\nu_D}, \text{Va}, N)_Q \quad (113)$$

where $C^{+\nu_C}$ represent cations with charge $+\nu_C$, $D^{-\nu_D}$ represent anions with charge $-\nu_D$, Va represent hypothetical 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:

$$P = \sum_A y_A \nu_A + y_{\text{Va}} Q \quad (114)$$

$$Q = \sum_C y_C \nu_C \quad (115)$$

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

$$(Fe^{+2}, Cu^{+1})_P(S^{-2}, \text{Va}, S)_Q \quad (116)$$

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 Cu₂S when the second sublattice has mainly S²⁻ and the first a single metallic ion.

The mass balance equation is like for the CEF model:

$$M_A = P \sum_i b_{iA} y_i + Q \left(\sum_j b_{jA} y_j + \sum_k b_{kA} y_k \right) \quad (117)$$

where the b_{iA} , b_{jA} and b_{kA} represent the stoichiometric factor of element A in cations, anions and neutrals respectively and all y_i the fraction of the constituent i . The derivative of this is slightly more complicated than for the CEF model as P and Q are not constant:

$$dM_A = dP \sum_i b_{iA} y_i + P \sum_i b_{Ai} dy_i + dQ \left(\sum_j b_{Aj} y_j + \sum_k b_{Ak} y_k \right) + Q \left(\sum_k b_{Ak} dy_k + \sum_j b_{Aj} dy_j \right) \quad (118)$$

where

$$dP = \sum_A dy_A \nu_A + dy_{V_a} Q + y_{V_a} dQ \quad (119)$$

$$dQ = \sum_C dy_C \nu_C \quad (120)$$

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 type 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 appropriate 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

$$C_P = \left(\frac{\partial H}{\partial T} \right)_{P, N_i} \quad (121)$$

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:

$$G(T, P, N_i) = \sum_{\alpha} \aleph^{\alpha} G_M^{\alpha}(T, P, y_{js}) \quad (122)$$

$$N_i = \sum_{\alpha} \aleph^{\alpha} N_i^{\alpha} \quad (123)$$

$$N_i^{\alpha} = \sum_s a_s^{\alpha} \sum_j b_{ij} y_{js}^{\alpha} \quad (124)$$

where N_i is the amount in moles of component i , \aleph^{α} 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:

$$G = H - TS = H + T \left(\frac{\partial G}{\partial T} \right)_{P, N_i} \quad (125)$$

$$H = G - T \left(\frac{\partial G}{\partial T} \right)_{P, N_i} \quad (126)$$

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

$$\begin{aligned} \left(\frac{\partial G}{\partial T} \right)_{P, N_i} &= \sum_{\alpha} \frac{\partial \aleph^{\alpha}}{\partial T} G_M^{\alpha} + \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial G_M^{\alpha}}{\partial T} \right)_{P, N_i} \\ &= \sum_{\alpha} \frac{\partial \aleph^{\alpha}}{\partial T} G_M^{\alpha} + \sum_{\alpha} \aleph^{\alpha} \left[\left(\frac{\partial G_M^{\alpha}}{\partial T} \right)_{P, y_{js}} + \sum_{js} \left(\frac{\partial G_M^{\alpha}}{\partial y_{js}} \right)_{P, y_{kt} \neq js} \frac{\partial y_{js}}{\partial T} \right] \end{aligned} \quad (127)$$

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

$$C_P = \left(\frac{\partial H}{\partial T} \right)_{P, N_i} \quad (128)$$

$$\left(\frac{\partial H}{\partial T} \right)_{P, N_i} = \sum_{\alpha} \frac{\partial \aleph^{\alpha}}{\partial T} H_M^{\alpha} + \sum_{\alpha} \aleph^{\alpha} \left(\frac{\partial H_M^{\alpha}}{\partial T} \right)_{P, N_i} \quad (129)$$

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

$$\begin{aligned} \left(\frac{\partial H}{\partial T} \right)_{P, N_i} &= \sum_{\alpha} \frac{\partial \aleph^{\alpha}}{\partial T} H_M^{\alpha} + \sum_{\alpha} \aleph^{\alpha} \frac{\partial}{\partial T} \left[G_M^{\alpha} - T \left(\frac{\partial G_M^{\alpha}}{\partial T} \right)_{P, N_i} \right] \\ &= \sum_{\alpha} \frac{\partial \aleph^{\alpha}}{\partial T} H_M^{\alpha} - T \sum_{\alpha} \aleph^{\alpha} \frac{\partial}{\partial T} \left[\left(\frac{\partial G_M^{\alpha}}{\partial T} \right)_{P, N_i} \right] \\ &= \sum_{\alpha} \frac{\partial \aleph^{\alpha}}{\partial T} H_M^{\alpha} - T \sum_{\alpha} \aleph^{\alpha} \frac{\partial}{\partial T} \left[\left(\frac{\partial G_M^{\alpha}}{\partial T} \right)_{P, y_{js}} + \sum_j \left(\frac{\partial G_M^{\alpha}}{\partial y_{js}} \right)_{P, y_{k \neq j}} \frac{\partial y_{js}}{\partial T} \right] \\ &= \sum_{\alpha} \frac{\partial \aleph^{\alpha}}{\partial T} H_M^{\alpha} - T \sum_{\alpha} \aleph^{\alpha} \left[\left(\frac{\partial^2 G_M^{\alpha}}{\partial T^2} \right)_{P, y_{js}} + \sum_j \left(\frac{\partial^2 G_M^{\alpha}}{\partial T \partial y_{js}} \right)_{P, y_{k \neq j}} \frac{\partial y_{js}}{\partial T} \right] \end{aligned} \quad (130)$$

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.

$$\left(\frac{\partial H}{\partial T} \right)_{P, N_i} = \sum_{\alpha} \Delta \aleph^{\alpha} H_M^{\alpha} - T \sum_{\alpha} \aleph^{\alpha} \left[\left(\frac{\partial^2 G_M^{\alpha}}{\partial T^2} \right)_{P, y_j} + \sum_j \left(\frac{\partial^2 G_M^{\alpha}}{\partial T \partial y_j} \right)_{P, y_{k \neq j}} \Delta y_j \right] \quad (131)$$

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:

$$\begin{pmatrix} M_A & M_B & 0 & s_{14} \\ \aleph \sum_i \frac{\partial M_A}{\partial y_i} c_{iA} & \aleph \sum_i \frac{\partial M_A}{\partial y_i} c_{iB} & M_A & s_{24} \\ \aleph \sum_i \frac{\partial M_B}{\partial y_i} c_{iA} & \aleph \sum_i \frac{\partial M_B}{\partial y_i} c_{iB} & M_B & s_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta\mu_A \\ \Delta\mu_B \\ \Delta\aleph \\ \Delta T \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (132)$$

with the terms in 4'th column calculated as:

$$s_{14} = \frac{\partial G_M}{\partial T} \quad (133)$$

$$s_{24} = \aleph \sum_i \frac{\partial M_A}{\partial y_i} c_{iT} \quad (134)$$

$$s_{34} = \aleph \sum_i \frac{\partial M_B}{\partial y_i} c_{iT} \quad (135)$$

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 $\Delta\mu_A, \Delta\mu_B$ and $\Delta\aleph$, the value of ΔT is unity of course. The values of $\Delta\mu_A$ and $\Delta\mu_B$ are related to this change and the value of Δy_i can be calculated from eq. 36 using these $\Delta\mu_A$ and $\Delta\mu_B$:

$$\Delta y_i = c_{iT} \Delta T + \sum_A c_{iA} \Delta\mu_A \quad (136)$$

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.

7 Assessment of model parameters

This package also contain a subroutine used for assessment of model parametrs using experimental and theoretical data. In an assessment the difference between experimental data for various thermodynamic properties like enthalpies, heat capacities, phase diagram data etc. and the corresponding properites obtained from an equilibrium calculation using different sets of model parameters are minimimzed using model parameters varied by a least square fitting routine.

$$F(q_j) = \sum_i \left(\frac{x_i^{\text{experimental}} - x_i^{\text{model}}(q_j)}{\rho_i} w_i \right)^2 \quad (137)$$

where the summation is over all experiments, q_j are the j model parameters, $x_i^{\text{experimental}}$ are the i experimental data for the thermodynamic properties, $x_i^{\text{model}}(q_i)$ the same properties extracted from an equilibrium calculation using the model parameters q_i . ρ_i is the experimental uncertainty and w_i is a weight assigned to this information by the assessor.

The least square routine provides the subroutine with a set of variables, q_j and these are transferred by the subroutine to model coefficents in the thermodynamic model package, GTP. The experimental data have been obtained at an equilibrium state and thus the subroutine must make several equilibrium calculations to obtain the same data from the thermodynamic models. The information about the conditions for these equilibria is transferred to the subroutine by a structured datatype defined in GTP. The individually calculated values are returned to the least square routine:

$$F_i(q_j) = (x_i^{\text{experimental}} - x_i^{\text{model}}(q_j)) \frac{w_i}{\rho_i} \quad (138)$$

Inside the least square routine various nummerical techniques are used to minimize the sum of squares in eq. 137 by varying the variables q_j .

It is possibly to use inequalities for the experimental data also, i.e. that a property should be larger or smaller that a certain value. Finally one may also prescribe limits of the model parameters.

8 Software documentation

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

8.1 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.

8.1.1 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 :: sumxmole,sumwmole
! 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 iliqcharge,yva
! end specific ionic liquids
```

```
end TYPE meq_phase
```

8.1.2 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
! status for various things
! nrel number of elements (components)
! 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,status
    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)
! 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

```

8.1.3 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

```

8.2 Top level calculation routines

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

8.2.1 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.

The difference between calceq2 and 3 are that the latter does not write anything, except possible error messages.

```

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
    subroutine calceq3(mode,confirm,ceq)
! calculates the equilibrium for the given set of conditions
! mode=0 means no global minimization
! confirm is TRUE if output of CPU time

```

```

! ceq is a datastructure with all relevant thermodynamic data
  implicit none
  integer mode
  logical confirm
  TYPE(gtp_equilibrium_data), pointer :: ceq

```

8.2.2 Equilibrium calculations 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 variables
  implicit none
  integer mode
  TYPE(meq_setup), pointer :: meqrec
  type(map_fixph), pointer :: mapfix
  TYPE(gtp_equilibrium_data), pointer :: ceq

```

8.3 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.
  implicit none
  TYPE(meq_setup) :: meqrec
  TYPE(gtp_equilibrium_data), pointer :: ceq
  logical formap

```

8.4 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,inmap,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,inmap
TYPE(meq_setup) :: meqrec
TYPE(meq_phase), dimension(*), target :: phr
TYPE(gtp_equilibrium_data), pointer :: ceq

```

8.5 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_equlmatrix(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 equilibrium
! 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

```

8.6 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

```

8.7 Some utility routines

These are called when necessary for each iteration.

8.7.1 Correction 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_d2gdydyj(nkl, knr, curmu, pmi, ncc, nd1, pmat, ceq)
! correction of d2G/dy1dy2 for ionic liquid because the formula unit is
! not fixed. This contributes ONLY to the second derivatives 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, nd1, nkl(*), knr(*)
double precision curmu(*), pmat(nd1,*)
```

8.7.2 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
```

8.8 The coefficients in the Δy equation

These subroutines returns 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”.

The calc_dgdytemsh is the most recent needed to handle conditions on enthalpy. It is more flexible and the others should no longer be used.

```
subroutine calc_dgdyterms1(nrel, ia, tpindep, mamu, mag, mat, map, pmi, &
```

```

        curmux,noofits)
! any change must also be made in subroutine calc_dyterms2 and calc_dgdytermsh
! calculate the terms in the deltax 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 similar to calc_dgdyterms1
    implicit none
    integer iy,nrel
    double precision mag,mat,map,mamu(*)
    type(meq_phase), pointer :: pmi
    subroutine calc_dgdytermsh(nrel,ia,tpindep,hval,mamu,mag,mat,map,pmi,&
        curmux,noofits)
! This is a variant of dgdyterms1 including a term multiplied with each
! term (hval) in the summation over the constituents as needed when calculating
! an equation for fix V or H. If hval(i)=1.0 it should give the same
! results as dgdyterms1
!
! calculate the terms in the deltax 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(*) :: hval,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_dgdytermshm(nrel,ia,tpindep,hval,mamu,mag,mat,map,&
        mamu1,mag1,mat1,map1,pmi,curmux,noofits)
! This is a variant of dgdyterms1 including a term multiplied with each
! term (hval) in the summation over the constituents as needed when calculating
! an equation for fix V or H. If hval(i)=1.0 it should give the same
! results as dgdyterms1
!
! 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(*) :: hval,mamu,mamu1
    double precision mag,mat,map,mag1,mat1,map1
    double precision curmux(*)
! pmi is the phase data record for this phase
    type(meq_phase), pointer :: pmi

```

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

The dot derivatives requires 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 ...

8.9.1 Evaluate all state variable functions

This is called from the user interface to calculate values of all entered state variable functions. It writes the functions 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
```

8.9.2 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 OCPLLOT 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
```

8.9.3 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
```

8.9.4 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
```

8.9.5 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
```

8.9.6 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 returned
! 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(*)
```

8.10 Subroutine called in assessments

This subroutine calculates the difference between the experimental data and the same property calculated from the model.

The subroutine makes use of the data structure firstash declared in the GTP package and which must have vaules for all relevant coefficients and data. The values of X must be transferred to the relevant TPfun constants and F must be calculated for all experiments.

```

subroutine assessment_calfun(nexp,nvcoeff,f,x)
! nexp is number of experiments, nvcoeff number of coefficients
! f is the differences between experiments and value calculated by model
! returned by this subroutine
! x are the current model parameter values set by VA05AD
  implicit none
  integer nexp,nvcoeff
  double precision F(*),X(*)
!   type(gtp_assessmenthead), pointer :: ash

```

8.11 Subroutine called for listing extra data

It is possible to add extra output to calculations in connection with the command “enter many_equil” used in a “calculate all” command. These are calculated by this routine and the values listed on the output unit lut.

```

subroutine list_equilibrium_extra(lut,ceq)
! list the extra character variables for calculate symboles and
! list characters (if any), It is used in pmon and part of matsmin
! because it calls subroutines which need access to calculated results
  implicit none
  integer lut
  TYPE(gtp_equilibrium_data), pointer :: ceq

```

9 Summary

That’s all.

References

- [GTP] The General Thermodynamic Package, part of the OC software documentation, available at the opencalphad website.
- [OC-UI] The User Interface to the Open Calphad software, part of the OC software documentation, available at the opencalphad website.
- [opencalphad] <http://www.opencalphad.org>
- [github] opencalphad repository at <http://www.github.com>
- [09Flo] C A Floudas and C E Gounaris, J of Global Optimization **45** (2009) 3-38

- [75Eri] G Eriksson and E Rosén, Chem Scripta **4** (1973) 193
- [81Sun] B Sundman and J Ågren, J Chem Phys **42** (1981) 297
- [81Hil] M Hillert, Physica B, **103B** (1981) 31
- [82Luk] H L Lukas, J Weiss and E-Th Henig, Calphad **6** (1982) 229
- [84Jan] B Jansson, Ph D thesis, KTH, Stockholm (1984)
- [84Hil] M Hillert, B Jansson, B Sundman and J Ågren Metall Trans A **16A** (1985) 261
- [01Hil] M Hillert, J Alloys Compd **320** (2001) 161
- [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 **101**(2015) 127