The Open Calphad software, a status report

Bo Sundman, Ursula R Kattner, Mauro Palumbo and Suzana G Fries

INSTN CEA Saclay, France; NIST USA; SISSA Trieste, Italy; Ruhr University Bochum, Germany

Calphad June 2016

Objective

The Open Calphad (OC) provides a free thermodynamic software for:

- development of new thermodynamic models (only CEF and ionic liquid model are implemented at present),
- comparing algorithms for equilibrium calculations,
- assessments and development of assessment techniques,
- integration in simulation software.

Objective

The Open Calphad (OC) provides a free thermodynamic software for:

- development of new thermodynamic models (only CEF and ionic liquid model are implemented at present),
- comparing algorithms for equilibrium calculations,
- assessments and development of assessment techniques,
- integration in simulation software.

Additionally OC is fast as it can calculate equilibria in parallel.

Releases at http://www.opencalphad.org

1. March 2013
2. February 2015
3. February 2016

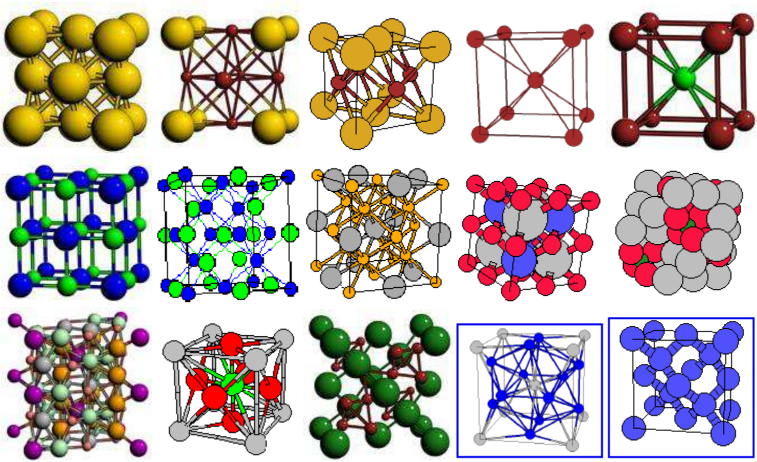
Content

- Data structure for storage, retrieval and calculation of the Gibbs energy for a single phase,
- Extended data structure to handle other properties,
- Databases and database formats,
- Algorithm for calculating equilibrium,
- STEP, MAP and PLOT routines,
- OC Application Software Interface (OCASI)
- Assessment of experimental and theoretical data

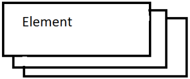
Content

- Data structure for storage, retrieval and calculation of the Gibbs energy for a single phase,
- Extended data structure to handle other properties,
- Databases and database formats,
- Algorithm for calculating equilibrium,
- STEP, MAP and PLOT routines,
- OC Application Software Interface (OCASI)
- Assessment of experimental and theoretical data
- Current development status
- Conclusions

Long range order models are needed for crystalline phases



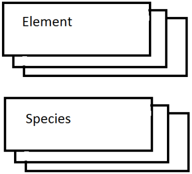
Data structure, the elements



A thermodynamic system is built from elements. Vacancies and electrons are included.

.

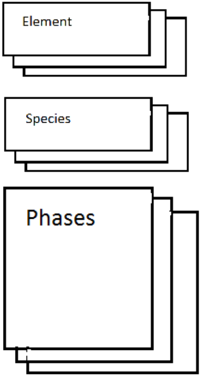
Data structure, the elements and species



A thermodynamic system is built from elements. Vacancies and electrons are included.
The species are stoichiometric combinations of elements.

.

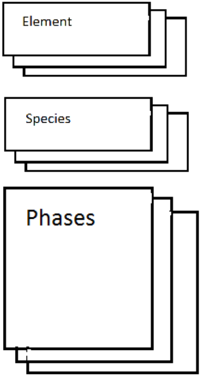
Data structure, the elements, species and phases



A thermodynamic system is built from elements. Vacancies and electrons are included.
The species are stoichiometric combinations of elements.

The phases have species as constituents.
The phases can have different models, constituents, sublattices, magnetic contribution.

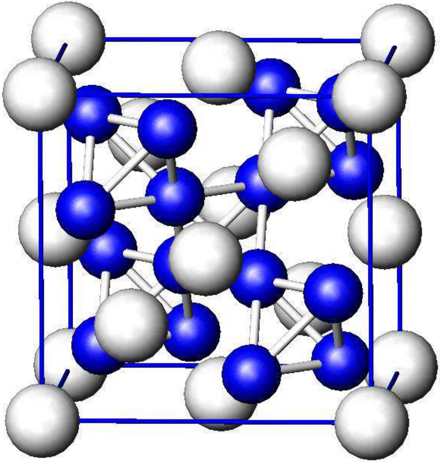
Data structure, the elements, species and phases



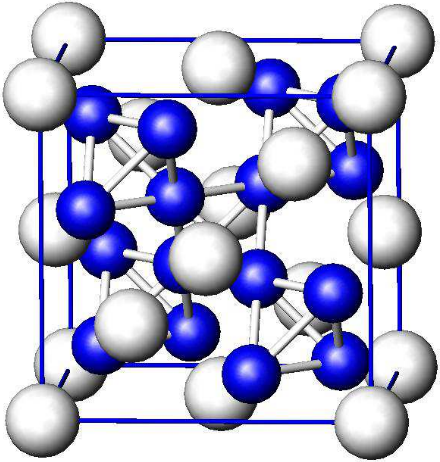
A thermodynamic system is built from elements. Vacancies and electrons are included.
The species are stoichiometric combinations of elements.

The phases have species as constituents.
The phases can have different models, constituents, sublattices, magnetic contribution.
Model parameters describe the Gibbs energy but also other composition dependent properties like Curie temperature, elastic constants, mobilities, etc.

Efficient storing of parameters for a phase $(A,D)_a(B,C)_b$



Efficient storing of parameters for a phase (A,D)$_a$(B,C)$_b$



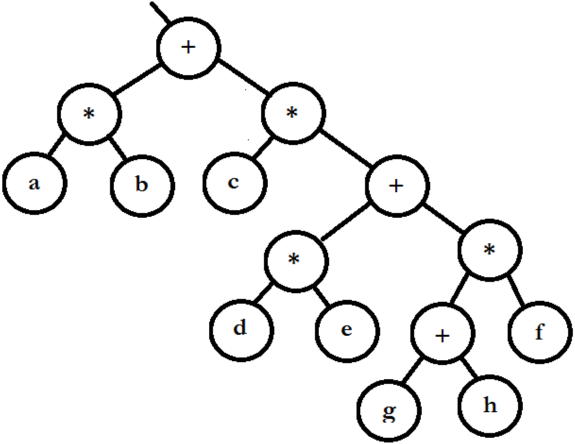
To handle this I have made use of what I learned about structured programming >>>

Storing a mathematical function as binary tree

$>>>$ where I had discovered that a function like

$$a * b + c * (d * e + f * (g + h))$$

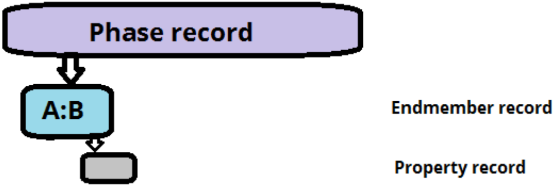
can be stored and calculated as a binary tree



Efficient storing of parameters for a phase (A,D)$_a$(B,C)$_b$

When entering a phase with or without sublattices and constituents OC does not automatically reserve any storage for model parameters.

Efficient storing of parameters for a phase (A,D)$_a$(B,C)$_b$



Phase record

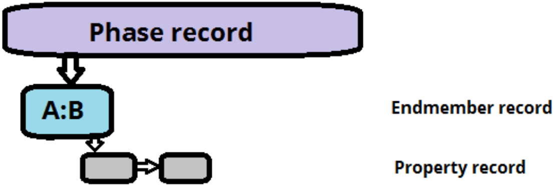
A:B

Endmember record

Property record

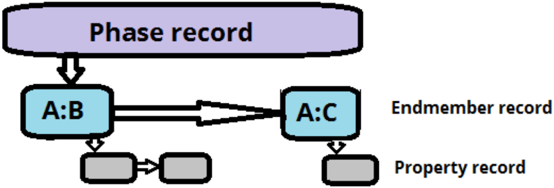
When entering an endmember parameter, $^\circ G_{\mathrm{A:B}}$, which depends on the fraction of A in the first sublattice and B in the second, a record is created linked from the phase record. The parameter expression is stored in a property record

Efficient storing of parameters for a phase $(\mathrm{A,D})_a(\mathrm{B,C})_b$



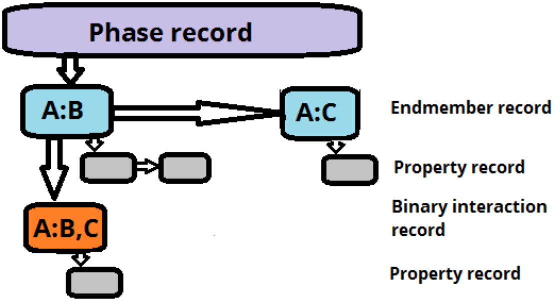
There can be several property records for the same endmember, like the Curie temperature $TC_{\mathrm{A:B}}$.

Efficient storing of parameters for a phase $(A,D)_a(B,C)_b$



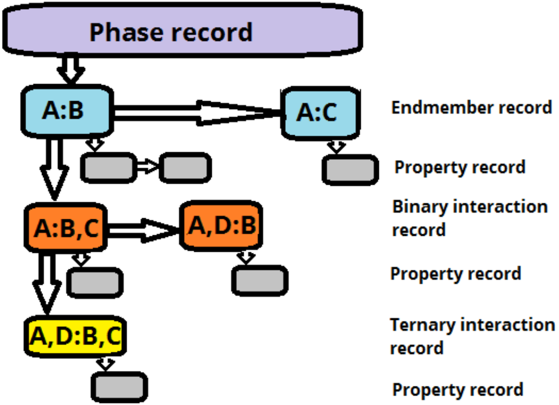
Additional endmember parameter records are stored in a linear list.

Efficient storing of parameters for a phase $(A,D)_a(B,C)_b$



An endmember record can also have a link to a record for a binary interaction parameter, $L_{\mathrm{A:B,C}}$. It “inherits” the constituents A:B from the endmember, only the interacting constituent, here C in the second sublattice, must be stored together with the property.

Efficient storing of parameters for a phase $(A,D)_a(B,C)_b$

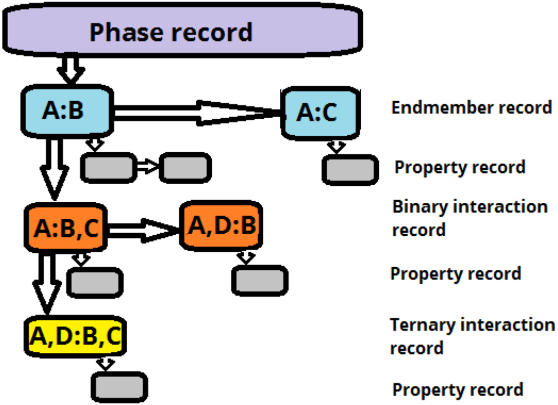


With additional model parameters the datastructure becomes a binary tree, here representing the expression

$$G = y_{\mathrm{A}} y_{\mathrm{B}}(\,^{\circ}G_{\mathrm{A:B}} + y_{\mathrm{C}}(L_{\mathrm{A:B,C}} + y_{\mathrm{D}} L_{\mathrm{A,D:B,C}}) + y_{\mathrm{D}} L_{\mathrm{A,D:B}}) + y_{\mathrm{A}} y_{\mathrm{C}} \,^{\circ}G_{\mathrm{A:C}}$$

The constituent fractions, y_i are stored in the dynamic data structure

Efficient storing of parameters for a phase $(A,D)_a(B,C)_b$



Phase record

A:B

A:C

Endmember record

Property record

A:B,C

A,D:B

Binary interaction record

Property record

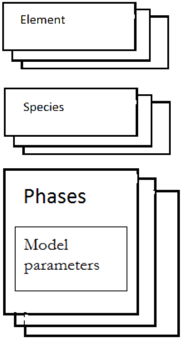
A,D:B,C

Ternary interaction record

Property record

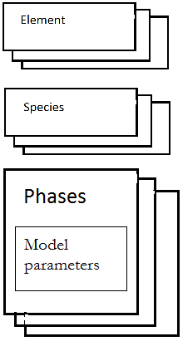
My experience is that all application software using thermodynamic data spend more than 90% of the computational time to calculate the Gibbs energies of the phases. The storage and calculation of model parameters is thus very important, more important than fast numerical routines to solve systems of equations.

Static and dynamic data

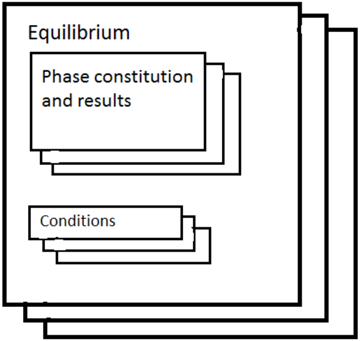


The data for elements, species and model parameters are independent on the external conditions on the system

Static and dynamic data



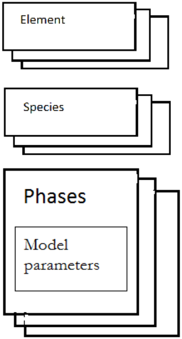
Dynamic data



The data for elements, species and model parameters are independent on the external conditions on the system

The amount and constitution of the phases vary with the external conditions like temperature and overall composition and are stored in the equilibrium record.

Static and dynamic data

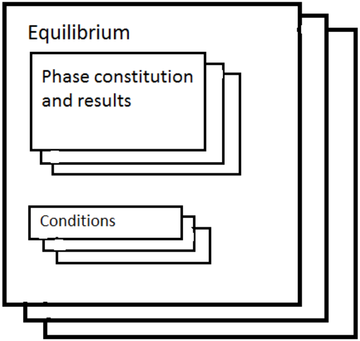

Element

Species

Phases

Model parameters

Dynamic data


Equilibrium

Phase constitution and results

Conditions

The data for elements, species and model parameters are independent on the external conditions on the system

In the dynamic data structure each equilibrium record is has a separate set of conditions and results and they can be calculated in parallel.

.

Model parameters

Calphad models depend on a lot of parameters describing the Gibbs energy outside the stable range of the phases and most important is the “lattice stability” concept.

Model parameters

Calphad models depend on a lot of parameters describing the Gibbs energy outside the stable range of the phases and most important is the “lattice stability” concept.

The Gibbs energy surface for the liquid and most solid phases like FCC, BCC, etc are modeled for all compositions between the pure elements, even if the phase is stable in a small composition or temperature range.

Model parameters

Calphad models depend on a lot of parameters describing the Gibbs energy outside the stable range of the phases and most important is the "lattice stability" concept.
The Gibbs energy surface for the liquid and most solid phases like FCC, BCC, etc are modeled for all compositions between the pure elements, even if the phase is stable in a small composition or temperature range.
For all properties we wish to model we have to make estimations of the values of this property for metastable states, even for simple things like the molar volumes, the molar volume of FCC-Cr cannot be measured.

Model parameters

Calphad models depend on a lot of parameters describing the Gibbs energy outside the stable range of the phases and most important is the "lattice stability" concept.
The Gibbs energy surface for the liquid and most solid phases like FCC, BCC, etc are modeled for all compositions between the pure elements, even if the phase is stable in a small composition or temperature range.
For all properties we wish to model we have to make estimations of the values of this property for metastable states, even for simple things like the molar volumes, the molar volume of FCC-Cr cannot be measured.

Model parameters are inter-dependent. If we want to change the properties of a pure element, even in a meta-stable state, we must probably revise all model parameters for binary and higher order systems with this element. Such dependencies must be clear in the database.

Model parameters

Calphad models depend on a lot of parameters describing the Gibbs energy outside the stable range of the phases and most important is the “lattice stability” concept.
The Gibbs energy surface for the liquid and most solid phases like FCC, BCC, etc are modeled for all compositions between the pure elements, even if the phase is stable in a small composition or temperature range.
For all properties we wish to model we have to make estimations of the values of this property for metastable states, even for simple things like the molar volumes, the molar volume of FCC-Cr cannot be measured.

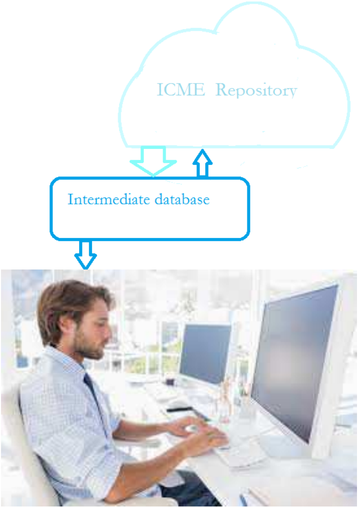
Model parameters are inter-dependent. If we want to change the properties of a pure element, even in a meta-stable state, we must probably revise all model parameters for binary and higher order systems with this element. Such dependencies must be clear in the database. Experimental data has usually not this kind of dependency.

Bottom up data

Thermodynamic databases grow from the bottom

Top down use

The materials genome and ICME initiatives will create large materials data repositories. Most likely several separate dataset will be stored for many systems and for an application a selection must be made and stored in intermediate database files.
A generalized TDB file is one kind of such databases for use in thermodynamic calculations and simulations.
The TDB format was a proposal from SGTE more than 30 years ago to provide a standard of data transfer between different software.



ICME Repository

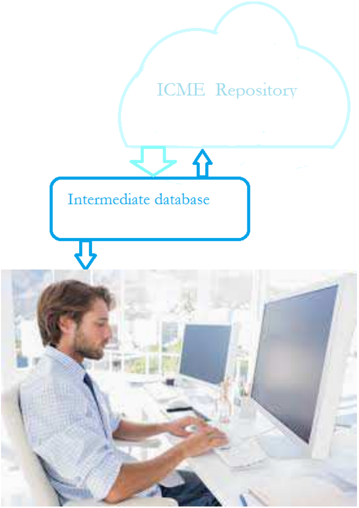
Intermediate database

Top down use

The materials genome and ICME initiatives will create large materials data repositories. Most likely several separate dataset will be stored for many systems and for an application a selection must be made and stored in intermediate database files.

A generalized TDB file is one kind of such databases for use in thermodynamic calculations and simulations.

The TDB format was a proposal from SGTE more than 30 years ago to provide a standard of data transfer between different software.



ICME Repository

Intermediate database

SGTE has decided to provide an update of the TDB format as part of its efforts to develop a new unary database and because several new software use variants of the TDB format.

Property identifier for other model parameters

An property that is of interest for materials simulation is the mobility of a component A in a phase. However, as it can vary several orders of magnitude with T and composition it is better to model its natural logarithm:

$$MQ^{\alpha}_{\mathrm{A}} = \ln\left(\nu^{\alpha}_{\mathrm{A}}\exp(\frac{Q^{\alpha}_{\mathrm{A}}}{RT})\right) \quad = \quad \frac{Q^{\alpha}_{\mathrm{A}}(T,P)}{RT}\ln(\nu^{\alpha}_{\mathrm{A}}(T,P))$$

where Q^{α}_{A} is an activation energy and $\nu^{\alpha}_{\mathrm{A}}$ a pre-exponential factor.

Property identifier for other model parameters

An property that is of interest for materials simulation is the mobility of a component A in a phase. However, as it can vary several orders of magnitude with T and composition it is better to model its natural logarithm:

$$MQ^{\alpha}_{\mathrm{A}} = \ln\left(\nu^{\alpha}_{\mathrm{A}} \exp(\frac{Q^{\alpha}_{\mathrm{A}}}{RT})\right) \quad = \quad \frac{Q^{\alpha}_{\mathrm{A}}(T,P)}{RT} \ln(\nu^{\alpha}_{\mathrm{A}}(T,P))$$

where Q^{α}_{A} is an activation energy and $\nu^{\alpha}_{\mathrm{A}}$ a pre-exponential factor.
Such a parameter does not fit into the current TDB notation because it need an additional component specifier in the phase. In OC the notation adopted is:
MQ&FE(BCC,FE) for self diffusion of Fe in BCC-Fe and
MQ&CR(BCC,CR) for self diffusion of Cr in BCC-Cr
These are endmember values for stable phases that can be measured.

Property identifier for other model parameters

An property that is of interest for materials simulation is the mobility of a component A in a phase. However, as it can vary several orders of magnitude with T and composition it is better to model its natural logarithm:

$$MQ^{\alpha}_{\mathrm{A}} = \ln\left(\nu^{\alpha}_{\mathrm{A}} \exp(\frac{Q^{\alpha}_{\mathrm{A}}}{RT})\right) \quad = \quad \frac{Q^{\alpha}_{\mathrm{A}}(T,P)}{RT} \ln(\nu^{\alpha}_{\mathrm{A}}(T,P))$$

where Q^{α}_{A} is an activation energy and $\nu^{\alpha}_{\mathrm{A}}$ a pre-exponential factor.
Such a parameter does not fit into the current TDB notation because it need an additional component specifier in the phase. In OC the notation adopted is:
MQ&FE(BCC,FE) for self diffusion of Fe in BCC-Fe and
MQ&CR(BCC,CR) for self diffusion of Cr in BCC-Cr
These are endmember values for stable phases that can be measured.
To describe how the mobility varies from pure Fe to pure Cr in the BCC phase we must have an endmember on the opposite side.
MQ&FE(BCC,CR) for diffusion of Fe in BCC-Cr at infinite dilution.
MQ&CR(BCC,FE) for diffusion of Cr in BCC-Fe at infinite dilution.

Property identifier for other model parameters

An property that is of interest for materials simulation is the mobility of a component A in a phase. However, as it can vary several orders of magnitude with T and composition it is better to model its natural logarithm:

$$MQ^{\alpha}_{\mathrm{A}} = \ln\left(\nu^{\alpha}_{\mathrm{A}} \exp(\frac{Q^{\alpha}_{\mathrm{A}}}{RT})\right) \quad = \quad \frac{Q^{\alpha}_{\mathrm{A}}(T,P)}{RT} \ln(\nu^{\alpha}_{\mathrm{A}}(T,P))$$

where Q^{α}_{A} is an activation energy and $\nu^{\alpha}_{\mathrm{A}}$ a pre-exponential factor.
Such a parameter does not fit into the current TDB notation because it need an additional component specifier in the phase. In OC the notation adopted is:
MQ&FE(BCC,FE) for self diffusion of Fe in BCC-Fe and
MQ&CR(BCC,CR) for self diffusion of Cr in BCC-Cr
These are endmember values for stable phases that can be measured.
To describe how the mobility varies from pure Fe to pure Cr in the BCC phase we must have an endmember on the opposite side.
MQ&FE(BCC,CR) for diffusion of Fe in BCC-Cr at infinite dilution.
MQ&CR(BCC,FE) for diffusion of Cr in BCC-Fe at infinite dilution.
These endmember parameters represent a value for the mobility when we have a single constituent in each sublattice.

Mobilities and diffusion coefficients

The mobility of Fe in the BCC phase in the Cr-Fe system is calculated as:

$$MQ\&\mathrm{FE(BCC)} = x_{\mathrm{Fe}} MQ\&\mathrm{FE}_{\mathrm{Fe}}^{\mathrm{BCC}} + x_{\mathrm{Cr}} MQ\&\mathrm{FE}_{\mathrm{Cr}}^{\mathrm{BCC}} + x_{\mathrm{Cr}} x_{\mathrm{Fe}} MQ\&\mathrm{FE}_{\mathrm{Cr,Fe}}^{\mathrm{BCC}}$$

where the last term can be used to describe any deviation from linearity.

Mobilities and diffusion coefficients

The mobility of Fe in the BCC phase in the Cr-Fe system is calculated as:

$$MQ\&\mathrm{FE(BCC)} = x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{BCC}}_{\mathrm{Fe}} + x_{\mathrm{Cr}} MQ\&\mathrm{FE}^{\mathrm{BCC}}_{\mathrm{Cr}} + x_{\mathrm{Cr}} x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{BCC}}_{\mathrm{Cr,Fe}}$$

where the last term can be used to describe any deviation from linearity. The mobility of Fe in the FCC phase has a similar equation:

$$MQ\&\mathrm{FE(FCC)} = x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{FCC}}_{\mathrm{Fe}} + x_{\mathrm{Cr}} MQ\&\mathrm{FE}^{\mathrm{FCC}}_{\mathrm{Cr}} + x_{\mathrm{Cr}} x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{FCC}}_{\mathrm{Cr,Fe}}$$

and require an estimate of the value of **MQ&FE(FCC,CR)**, i.e. the mobility of Fe in an FCC phase consisting of pure Cr. Hopefully DFT calculations can help.

Mobilities and diffusion coefficients

The mobility of Fe in the BCC phase in the Cr-Fe system is calculated as:

$$MQ\&\mathrm{FE}(\mathrm{BCC}) = x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{BCC}}_{\mathrm{Fe}} + x_{\mathrm{Cr}} MQ\&\mathrm{FE}^{\mathrm{BCC}}_{\mathrm{Cr}} + x_{\mathrm{Cr}} x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{BCC}}_{\mathrm{Cr,Fe}}$$

where the last term can be used to describe any deviation from linearity.
The mobility of Fe in the FCC phase has a similar equation:

$$MQ\&\mathrm{FE}(\mathrm{FCC}) = x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{FCC}}_{\mathrm{Fe}} + x_{\mathrm{Cr}} MQ\&\mathrm{FE}^{\mathrm{FCC}}_{\mathrm{Cr}} + x_{\mathrm{Cr}} x_{\mathrm{Fe}} MQ\&\mathrm{FE}^{\mathrm{FCC}}_{\mathrm{Cr,Fe}}$$

and require an estimate of the value of **MQ&FE(FCC,CR)**, i.e. the mobility of Fe in an FCC phase consisting of pure Cr. Hopefully DFT calculations can help.
To convert mobilities to diffusion coefficients we also need to calculate:

$$\left(\frac{\partial \mu_{\mathrm{A}}}{\partial x_{\mathrm{B}}}\right) = \frac{1}{N}\left(\frac{\partial^2 G_M^\alpha}{\partial x_{\mathrm{A}} \partial x_{\mathrm{B}}} - \sum_i x_i \left(\frac{\partial^2 G_M^\alpha}{\partial x_i \partial x_{\mathrm{A}}} + \frac{\partial^2 G_M^\alpha}{\partial x_i \partial x_{\mathrm{B}}}\right) + \sum_i \sum_j x_i x_j \frac{\partial^2 G_M^\alpha}{\partial x_i \partial x_j}\right)$$

in the α phase for cases when we are in a non-equilibrium state during a diffusional process (when there are sublattices it is even more complicated).

Parameter property identifier

Inside the OC source code property identifiers can be entered as shown in this piece of the source code

```
! logarithm of individual mobility
   npid=npid+1
   propid(npid)%symbol=’MQ  ’
   propid(npid)%note=’LN mob. of const.’
   propid(npid)%status=0
! MQ is specific for a constituent
   propid(npid)%status=ibset(propid(npid)%status,IDCONSUFFIX)
```

where *propid* is an array of property records and *npid* is the internal index of this property.
Each property has a symbol and it can be a constant or depend on T and/or P and, as for MQ, have individual values for each component of the phase.

Parameter property identifier

Inside the OC source code property identifiers can be entered as shown in this piece of the source code

```
! logarithm of individual mobility
   npid=npid+1
   propid(npid)%symbol=’MQ ’
   propid(npid)%note=’LN mob. of const.’
   propid(npid)%status=0
! MQ is specific for a constituent
   propid(npid)%status=ibset(propid(npid)%status,IDCONSUFFIX)
```

where *propid* is an array of property records and *npid* is the internal index of this property.
Each property has a symbol and it can be a constant or depend on T and/or P and, as for MQ, have individual values for each component of the phase. As the OC source code is free anyone can add the properties and code he needs for his application.

Parameter property identifier

Inside the OC source code property identifiers can be entered as shown in this piece of the source code

```
! logarithm of individual mobility
   npid=npid+1
   propid(npid)%symbol='MQ  '
   propid(npid)%note='LN mob. of const.'
   propid(npid)%status=0
! MQ is specific for a constituent
   propid(npid)%status=ibset(propid(npid)%status,IDCONSUFFIX)
```

where *propid* is an array of property records and *npid* is the internal index of this property.

Each property has a symbol and it can be a constant or depend on T and/or P and, as for MQ, have individual values for each component of the phase. As the OC source code is free anyone can add the properties and code he needs for his application.

During an equilibrium calculation these property parameters can be calculated for the current T, P and composition of the phases at the same time as the Gibbs energy.

Calculating the equilibrium state

The equilibrium at fixed temperature, pressure and composition is given by a minimum in the total Gibbs energy of the system. The thermodynamic models describe the molar Gibbs energy, G_M^α and the total energy for the system is:

$$G = \sum_\alpha \aleph^\alpha G_M^\alpha \tag{1}$$

where $\aleph^\alpha$ is the amount of the stable phases.

Calculating the equilibrium state

The equilibrium at fixed temperature, pressure and composition is given by a minimum in the total Gibbs energy of the system. The thermodynamic models describe the molar Gibbs energy, G_M^α and the total energy for the system is:

$$G = \sum_\alpha \aleph^\alpha G_M^\alpha \tag{1}$$

where $\aleph^\alpha$ is the amount of the stable phases.
If one has other conditions like fixed volume, chemical potentials or fixed phases one can modify the Gibbs energy by the use of Lagrange multipliers.

Calculating the equilibrium state

The equilibrium at fixed temperature, pressure and composition is given by a minimum in the total Gibbs energy of the system. The thermodynamic models describe the molar Gibbs energy, G_M^α and the total energy for the system is:

$$G = \sum_\alpha \aleph^\alpha G_M^\alpha \tag{1}$$

where $\aleph^\alpha$ is the amount of the stable phases.
If one has other conditions like fixed volume, chemical potentials or fixed phases one can modify the Gibbs energy by the use of Lagrange multipliers.

The algorithm used in OC was published by Mats Hillert in 1981.
It is a much more flexible and faster method for equilibrium calculations than using any kind of "common tangent" construction.

Single equilibrium calculation, Lagrange multipliers

The algorithm uses Lagrange multipliers to take care of both external and interal constraints. At constant T, P and overall composition, N_{A} we have:

$$L = G + \sum_{\mathrm{A}} (N_{\mathrm{A}} - \sum_{\alpha} \aleph^{\alpha} M_{\mathrm{A}}^{\alpha}) \xi_{\mathrm{A}} + \sum_{\alpha} \sum_{s} \eta_{s}^{\alpha} (\sum_{i} y_{is}^{\alpha} - 1) \qquad (2)$$

where N_{A} is the total amount of component A, M_{A}^{α} is the amount of component A and y_{is}^{α} is the fraction of constituent i on sublattice s in phase α. ξ_{A} and η_{s}^{α} are Lagrange multipliers.

Single equilibrium calculation, Lagrange multipliers

The algorithm uses Lagrange multipliers to take care of both external and interal constraints. At constant T, P and overall composition, N_{A} we have:

$$L = G + \sum_{\mathrm{A}} (N_{\mathrm{A}} - \sum_{\alpha} \aleph^{\alpha} M^{\alpha}_{\mathrm{A}}) \xi_{\mathrm{A}} + \sum_{\alpha} \sum_{s} \eta^{\alpha}_{s} (\sum_{i} y^{\alpha}_{is} - 1) \qquad (2)$$

where N_{A} is the total amount of component A, M^{α}_{A} is the amount of component A and y^{α}_{is} is the fraction of constituent i on sublattice s in phase α. ξ_{A} and η^{α}_{s} are Lagrange multipliers.

The first summation is the external massbalance conditions.

The second constraint is that the sum of constituent fractions in each sublattice of α must be unity.

With ionic constituents there is a constraint due to the electro-neutrality

Single equilibrium calculation, Lagrange multipliers

The algorithm uses Lagrange multipliers to take care of both external and interal constraints. At constant T, P and overall composition, N_A we have:

$$L = G + \sum_\mathrm{A}(N_\mathrm{A} - \sum_\alpha \aleph^\alpha M_\mathrm{A}^\alpha)\xi_\mathrm{A} + \sum_\alpha \sum_s \eta_s^\alpha(\sum_i y_{is}^\alpha - 1) \qquad (2)$$

where N_A is the total amount of component A, M_A^α is the amount of component A and y_{is}^α is the fraction of constituent i on sublattice s in phase α. ξ_A and η_s^α are Lagrange multipliers.

At equilibrium all partial derivatives must be zero and from the derivative of the Lagrange function with respect to $\aleph^\alpha$ and using $G = \sum_\alpha \aleph^\alpha G_M^\alpha$:

$$\frac{\partial L}{\partial \aleph^\alpha} = G_M^\alpha - \sum_\mathrm{A} M_\mathrm{A}^\alpha \xi_\mathrm{A} = 0 \qquad (3)$$

which means the multiplier ξ_A is the chemical potential as $G_M^\alpha = \sum_\mathrm{A} \mu_\mathrm{A} M_\mathrm{A}^\alpha$

Single equilibrium calculation, Lagrange multipliers

The algorithm uses Lagrange multipliers to take care of both external and interal constraints. At constant T, P and overall composition, N_{A} we have:

$$L = G + \sum_{\mathrm{A}} (N_{\mathrm{A}} - \sum_{\alpha} \aleph^{\alpha} M_{\mathrm{A}}^{\alpha}) \xi_{\mathrm{A}} + \sum_{\alpha} \sum_{s} \eta_s^{\alpha} (\sum_{i} y_{is}^{\alpha} - 1) \qquad (2)$$

where N_{A} is the total amount of component A, M_{A}^{α} is the amount of component A and y_{is}^{α} is the fraction of constituent i on sublattice s in phase α. ξ_{A} and η_s^{α} are Lagrange multipliers.

The implementation of this algorithm in OC was published last year together with some related facilities.

Global grid minimizer provides start values

Hillert's algorithm is iterative which depends strongly on the initial set of stable phases and their constitutions.

Global grid minimizer provides start values

Hillert's algorithm is iterative which depends strongly on the initial set of stable phases and their constitutions.
To ensure the calculated equilibrium is a global minimum the OC software has an initial step which calculates the Gibbs energy of all phases over a grid of compositions.

Global grid minimizer provides start values

Hillert's algorithm is iterative which depends strongly on the initial set of stable phases and their constitutions.

To ensure the calculated equilibrium is a global minimum the OC software has an initial step which calculates the Gibbs energy of all phases over a grid of compositions.

These are treated as stoichiometric phases in a preliminary minimization to find a set of gridpoints representing a global minimum.

Global grid minimizer provides start values

Hillert's algorithm is iterative which depends strongly on the initial set of stable phases and their constitutions.

To ensure the calculated equilibrium is a global minimum the OC software has an initial step which calculates the Gibbs energy of all phases over a grid of compositions.

These are treated as stoichiometric phases in a preliminary minimization to find a set of gridpoints representing a global minimum.

This set of gridpoints is then used as initial guess of the set of stable phases and their constitutions are inserted as start values for the iterative algorithm.

Global grid minimizer provides start values

Hillert's algorithm is iterative which depends strongly on the initial set of stable phases and their constitutions.

To ensure the calculated equilibrium is a global minimum the OC software has an initial step which calculates the Gibbs energy of all phases over a grid of compositions.

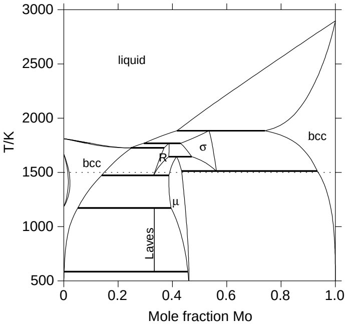
These are treated as stoichiometric phases in a preliminary minimization to find a set of gridpoints representing a global minimum.

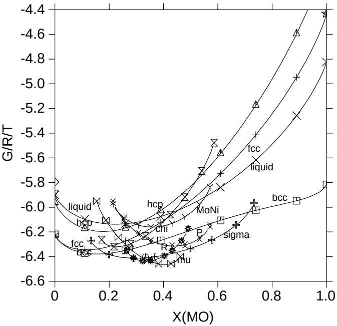
This set of gridpoints is then used as initial guess of the set of stable phases and their constitutions are inserted as start values for the iterative algorithm.

The grid minimizer will also automatically find miscibility gaps.

Grid minimizer, an example

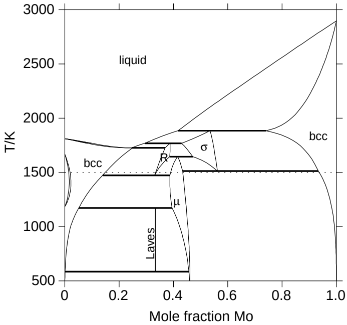
In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.

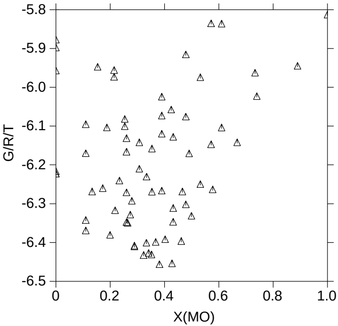




Grid minimizer, an example

In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.

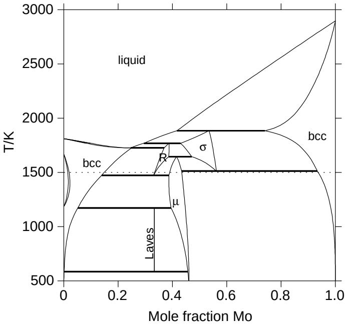


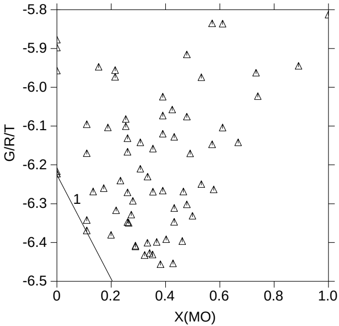


The gridpoints are treated as individual stoichiometric phases by the grid minimizer.

Grid minimizer, an example

In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.

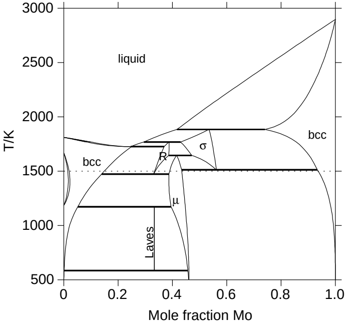


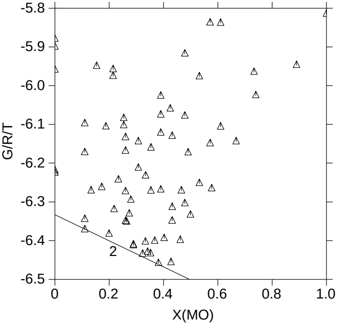


For low Mo content these two gridpoints will be selected as starting constitutions. They represent the FCC and BCC solution phases.

Grid minimizer, an example

In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.

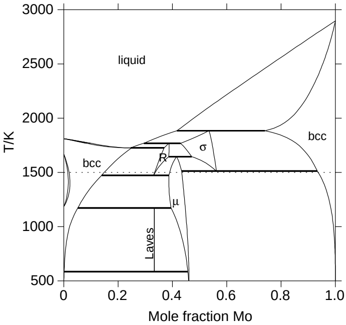


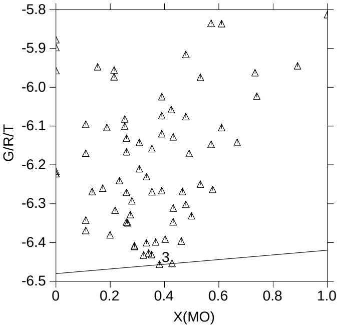


For higher Mo content these two gridpoints will be selected. They represent the BCC and μ phases. The iterative algorithm when then replace either BCC or μ with the R phase.

Grid minimizer, an example

In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.

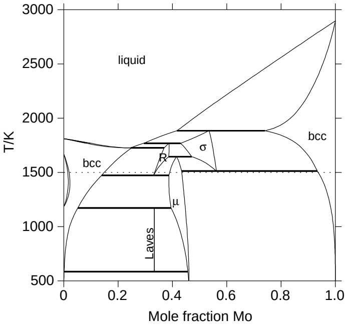


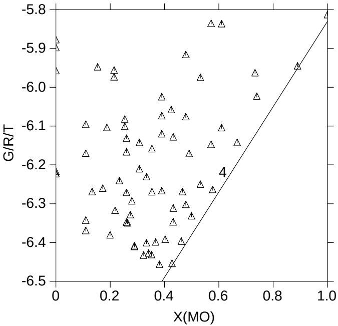


For Mo content around 40% these two gripoints will be selected. They are both in the μ phase and the iterative algorithm will merge them into a single composition.

Grid minimizer, an example

In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.

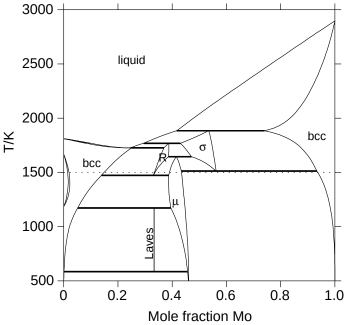


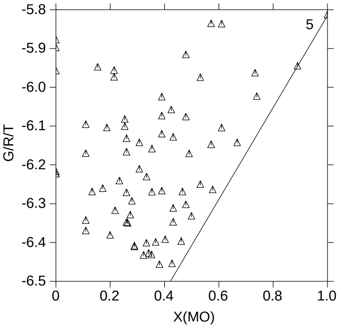


At higher Mo content one gridpoint in the μ and another in the Mo-rich BCC phase will be selected.

Grid minimizer, an example

In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.

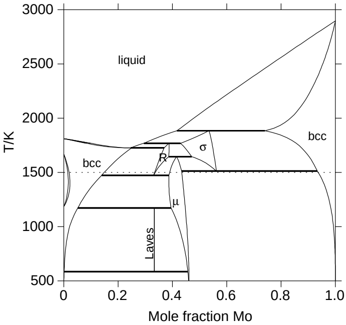


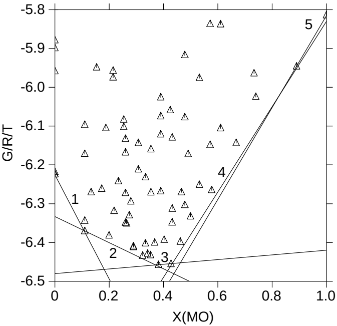


For very high Mo content these two gridpoints in the BCC phase will be used as start points. They will merge to a single composition during the iterative minimization.

Grid minimizer, an example

In the Fe-Mo system at 1500 K the grid minimizer calculates a number of points along the Gibbs energy curves of the different phases.





Here is the convex hull represented by the gridpoints.
The density of the grid can be selected but more gridpoints require longer computational times. When calculating diagrams OC uses the previous calculated equilibrium as startpoint but occationally global tests can be made.

User interface

OC command line interface consists of single words:

```
OC3:?
 ABOUT            ENTER            LIST             QUIT
 AMEND            EXIT             MACRO            READ
 BACK             FIN              MAP              SAVE
 CALCULATE        HELP             NEW              SELECT
 DEBUG            HPCALC           OPTIMIZE         SET
 DELETE           INFORMATION      PLOT             STEP
```

User interface

OC command line interface consists of single words:

```
OC3:?
 ABOUT            ENTER            LIST             QUIT
 AMEND            EXIT             MACRO            READ
 BACK             FIN              MAP              SAVE
 CALCULATE        HELP             NEW              SELECT
 DEBUG            HPCALC           OPTIMIZE         SET
 DELETE           INFORMATION      PLOT             STEP
```

Each command can operate on several objects like:

```
OC3:enter ?
 Possible answers are:
 BIBLIOGRAPHY     ELEMENT          OPTIMIZE_COEFF   SPECIES
 COMMENT          EQUILIBRIUM      PARAMETER        SYMBOL
 CONSTITUTION     EXPERIMENT       PHASE            TPFUN_SYMBOL
 COPY_OF_EQUILIB  MANY_EQUILIBRIA  QUIT
ENTER what? /SYMBOL/:
```

User interface

Most state variables like T, P, N(CR), X(CR), W%(LIQUID,CR), H etc can be used as conditions or listed after a calculations.

User interface

Most state variables like T, P, N(CR), X(CR), W%(LIQUID,CR), H etc can be used as conditions or listed after a calculations.

Setting a phase as fixed is treated as a condition that this phase must be stable in the calculated equilibrium. This can be used to calculate a phase transition point and is used to follow zero-phase fraction (ZPF) lines when calculating phase diagrams.

User interface

Most state variables like T, P, N(CR), X(CR), W%(LIQUID,CR), H etc can be used as conditions or listed after a calculations.

Setting a phase as fixed is treated as a condition that this phase must be stable in the calculated equilibrium. This can be used to calculate a phase transition point and is used to follow zero-phase fraction (ZPF) lines when calculating phase diagrams.

Anyone interested in a graphical user interface is welcome to develop one as OC is free.

Note that OC has no databases and that OC cannot read encrypted commercial databases.

STEP, MAP and PLOT procedures

The command STEP will calculate equilibria at fixed values along a single axis and additionally the exact axis value when the set of stable phases changes. Many different state variables can be plotted as a function of this axis variable from one step calculation.

STEP, MAP and PLOT procedures

The command STEP will calculate equilibria at fixed values along a single axis and additionally the exact axis value when the set of stable phases changes. Many different state variables can be plotted as a function of this axis variable from one step calculation.

The MAP command requires two or more axis variables and will calculate lines separating regions with different sets of stable phases within the axis limits, i.e. a phase diagram. The map procedure replaces all but one of the axis variables with a fix phase and then follow the axis variable that varies most rapidly.

STEP, MAP and PLOT procedures

The command STEP will calculate equilibria at fixed values along a single axis and additionally the exact axis value when the set of stable phases changes. Many different state variables can be plotted as a function of this axis variable from one step calculation.

The MAP command requires two or more axis variables and will calculate lines separating regions with different sets of stable phases within the axis limits, i.e. a phase diagram. The map procedure replaces all but one of the axis variables with a fix phase and then follow the axis variable that varies most rapidly.
When the set of stable phases changes one has a “node point” where several lines meet and start points for these are generated.
Phase diagrams with “tie-lines in the plane”, will automatically plot the line “on the other side” of the tie-line.

STEP, MAP and PLOT procedures

The command STEP will calculate equilibria at fixed values along a single axis and additionally the exact axis value when the set of stable phases changes. Many different state variables can be plotted as a function of this axis variable from one step calculation.

The MAP command requires two or more axis variables and will calculate lines separating regions with different sets of stable phases within the axis limits, i.e. a phase diagram. The map procedure replaces all but one of the axis variables with a fix phase and then follow the axis variable that varies most rapidly.

When the set of stable phases changes one has a “node point” where several lines meet and start points for these are generated.

Phase diagrams with “tie-lines in the plane”, will automatically plot the line “on the other side” of the tie-line.

The basic MAP procedure works but there are problems with miscibility gaps and exiting from node points in multicomponent systems.

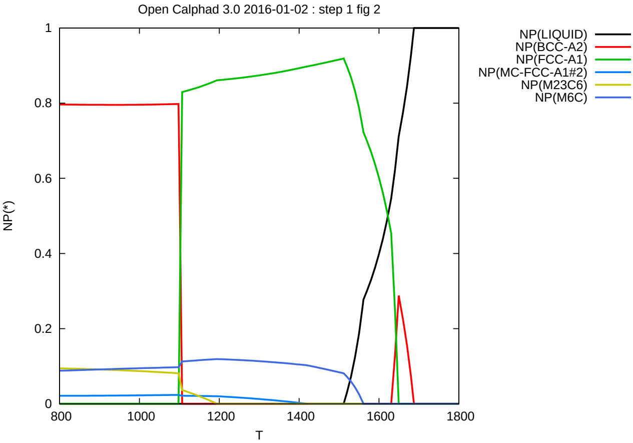
STEP, MAP and PLOT routines

The PLOT command creates a file for the GNUPLOT software with two axis variables specified by the user. This file can be further manipulated within the GNUPLOT software.

STEP, MAP and PLOT routines

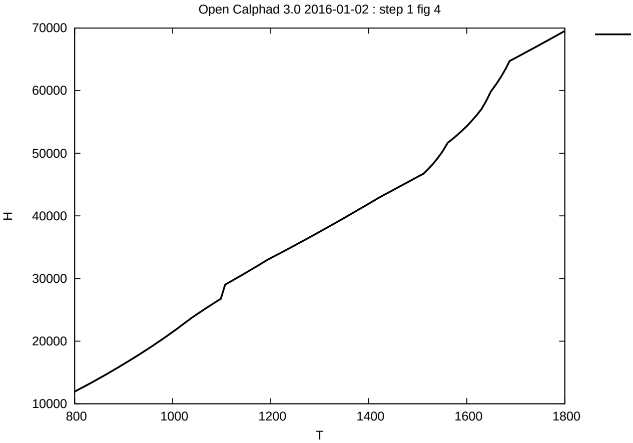
Some examples of diagrams generated by OC. Macro files to generate these are provided with the software.

A phase fraction plot of a High Speed Steel (HSS)



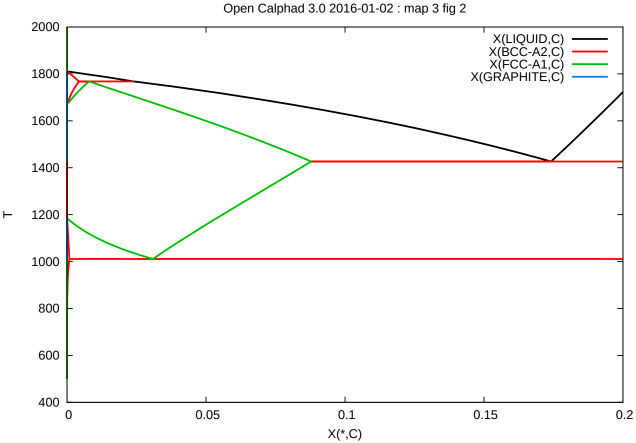
STEP, MAP and PLOT routines

Some examples of diagrams generated by OC. Macro files to generate these are provided with the software.
The enthalpy curve for the same HSS



STEP, MAP and PLOT routines

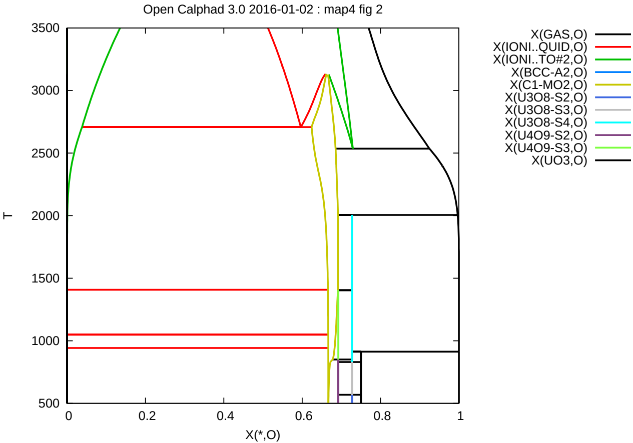
Some examples of diagrams generated by OC. Macro files to generate these are provided with the software.
The Fe-rich side of the C-Fe phase diagram



STEP, MAP and PLOT routines

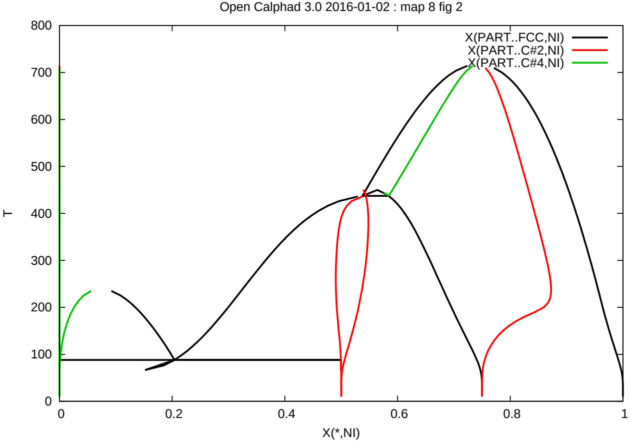
Some examples of diagrams generated by OC. Macro files to generate these are provided with the software.

The O-U phase diagram



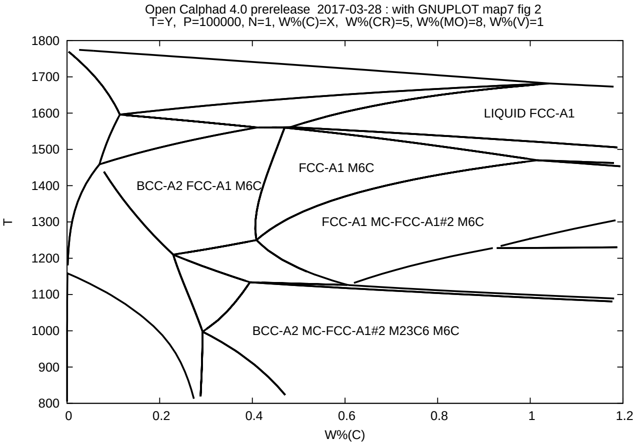
STEP, MAP and PLOT routines

Some examples of diagrams generated by OC. Macro files to generate these are provided with the software.
Stable and metastable fcc ordering in Fe-Ni



STEP, MAP and PLOT routines

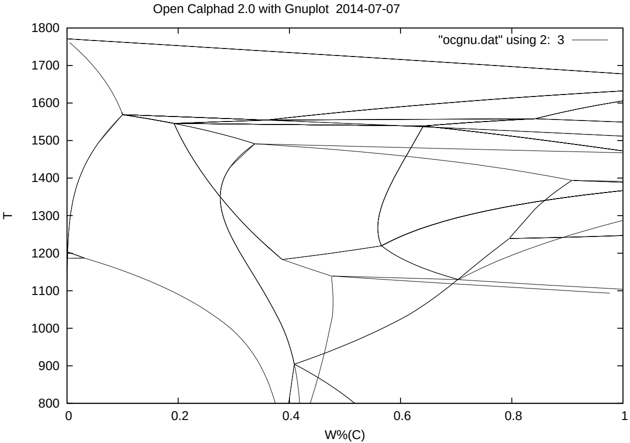
Some examples of diagrams generated by OC. Macro files to generate these are provided with the software.
A failed mapping of the HSS ... many things still to fix



STEP, MAP and PLOT routines

Some examples of diagrams generated by OC. Macro files to generate these are provided with the software.

In an earlier version of OC this diagram looked like this ...



Parallelization

Inside OC one can enter several equilibria for the same system with different sets of conditions. Care has been taken that the data structure for each equilibrium is completely independent so they can be calculated in parallel.

Parallelization

Inside OC one can enter several equilibria for the same system with different sets of conditions. Care has been taken that the data structure for each equilibrium is completely independent so they can be calculated in parallel.
This has been tested with the OpenMP library and works very nicely.
On a PC with 4 kernels and 8 threads the gain in speed is almost a factor of 4, i.e. there is no overhead.

Parallelization

Inside OC one can enter several equilibria for the same system with different sets of conditions. Care has been taken that the data structure for each equilibrium is completely independent so they can be calculated in parallel.
This has been tested with the OpenMP library and works very nicely.
On a PC with 4 kernels and 8 threads the gain in speed is almost a factor of 4, i.e. there is no overhead.
OC will use this for assessments, because each experiment is an independent equilibrium.

Parallelization is also available in the application software interface (OCASI) and has already been used with considerably success.

OC Application Software Interface, OCASI

OCASI is based on the TQ standard for thermodynamic software interfaces, used by Thermo-Calc and ChemApp, and will provide similar facilities.

OC Application Software Interface, OCASI

OCASI is based on the TQ standard for thermodynamic software interfaces, used by Thermo-Calc and ChemApp, and will provide similar facilities.

An advantage can be that OCASI is available as source code so it can be modified by the application programmer and compiled on different hardwares, OS or compilers.

Example, solidification and homogenization

An aluminum company in France has converted a software they used for simulating solidification and homogenization to use OCASI.

They used an alloy with the composition

Si	Fe	Cu	Mn	Mg	Zn	Ti	Zr	Al
0.06	0.06	1.8	0.1	2.3	8.1	0.06	0.06	rest

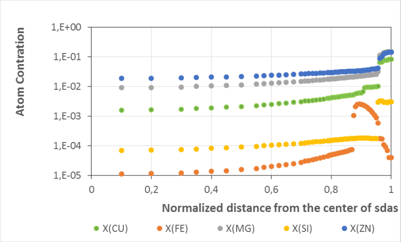
Example, solidification and homogenization

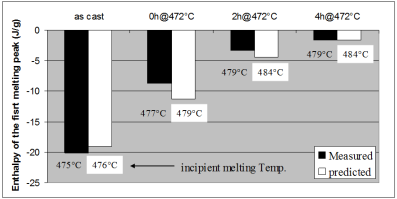
An aluminum company in France has converted a software they used for simulating solidification and homogenization to use OCASI.

They used an alloy with the composition

Si	Fe	Cu	Mn	Mg	Zn	Ti	Zr	Al
0.06	0.06	1.8	0.1	2.3	8.1	0.06	0.06	rest

Their own software took about 3.5 days to simulate the solidification and several subsequent homogenization steps.





The left hand figure show the concentration profiles across a dendrite after solidification, the right the heat evolved during homogenization.

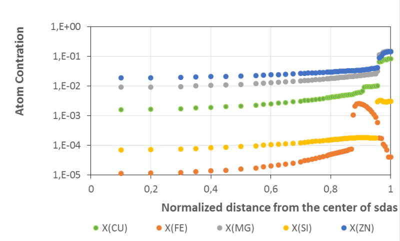
Example, solidification and homogenization

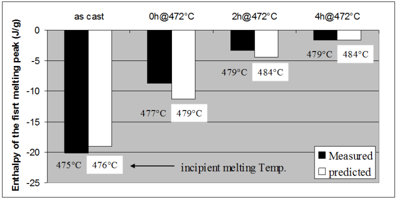
An aluminum company in France has converted a software they used for simulating solidification and homogenization to use OCASI.

They used an alloy with the composition

Si	Fe	Cu	Mn	Mg	Zn	Ti	Zr	Al
0.06	0.06	1.8	0.1	2.3	8.1	0.06	0.06	rest

Their own software took about 3.5 days to simulate the solidification and several subsequent homogenization steps.





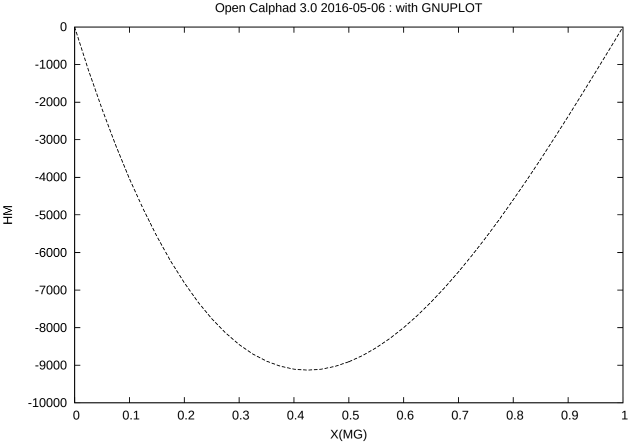
After implementing OCASI the same simulation took 3.5 hours using 12 CPUs in parallel, a gain of time with a factor of 24.

Assessment

A routine developed in Argonne National Lab for least square fitting has been included to minimize differences between experimental data and the same data calculated from the models by varying selected model parameters.

Assessment

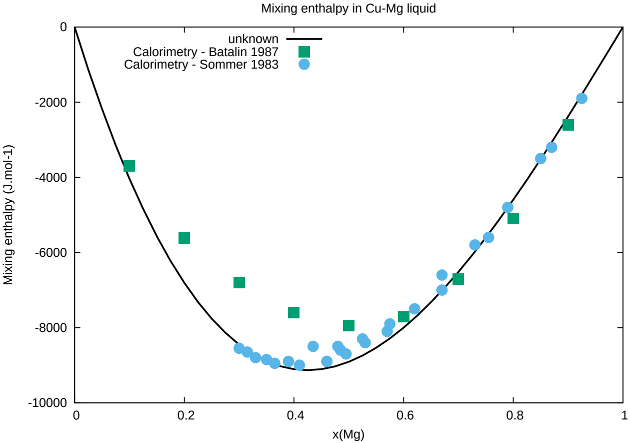
A routine developed in Argonne National Lab for least square fitting has been included to minimize differences between experimental data and the same data calculated from the models by varying selected model parameters.



The calculated enthalpy of mixing in the liquid Cu-Mg system modeled with 3 interaction parameters.

Assessment

A routine developed in Argonne National Lab for least square fitting has been included to minimize differences between experimental data and the same data calculated from the models by varying selected model parameters.



The calculated enthalpy of mixing in the liquid Cu-Mg system overlayed with experimental data.

Current development status

What is working:

- CEF and ionic liquid model
- Reading (unencrypted) TDB files and entering data interactively
- Command oriented user interface with macro facilities
- Setting many different types of conditions
- Calculating multicomponent equilibria
- STEP calculation of property diagram
- MAP of phase diagram is fragile, binary usually OK
- PLOT using GNUPLOT software
- The software interface, OCASI, has the basic routines, the rest has to be found in the OC source code
- Parallel calculation of many multicomponent equilibria
- Assessment of model parameters

Current development status

What is still in the pipeline:

- More models are waiting to be implemented, in particular for unaries.
- Phases with order/disorder transformations have some problems
- A few types of conditions not implemented like expressions
- There is not yet any SAVE/READ of a current session.
- Phase diagrams are not yet well handled
- Assessment works but there are many utilities missing
- OCASI works but needs to be restructured.
- Improving convergence and stability of calculations

Current development status

The best of all: Everyone interested can participate.

Conclusion: increase the market

The development of OpenCalphad has been critized because OC may destroy the market for commercial software with high quality databases.

Conclusion: increase the market

The development of OpenCalphad has been critized because OC may destroy the market for commercial software with high quality databases. In my opinion it is more important that Calphad groups collaborate to extend the market for thermodynamic calculations with better software and databases.

Not until all experimental work on materials is fully integrated with computational thermodynamics, we can be satisfied. Many materials scientists are still sceptical or even unaware of Calphad.

Conclusion: increase the market

The development of OpenCalphad has been critized because OC may destroy the market for commercial software with high quality databases. In my opinion it is more important that Calphad groups collaborate to extend the market for thermodynamic calculations with better software and databases.

Not until all experimental work on materials is fully integrated with computational thermodynamics, we can be satisfied. Many materials scientists are still sceptical or even unaware of Calphad.

Developing high quality thermodynamic databases require skill and extensive funding which has to be paid by the users. But keeping the thermodynamic software secret prevents necessary developments of new models. And it is anyway not certain that the income from sales of thermodynamic software is used to develop models and databases.

Conclusion: increase the market

The development of OpenCalphad has been critized because OC may destroy the market for commercial software with high quality databases. In my opinion it is more important that Calphad groups collaborate to extend the market for thermodynamic calculations with better software and databases.
Not until all experimental work on materials is fully integrated with computational thermodynamics, we can be satisfied. Many materials scientists are still sceptical or even unaware of Calphad.

A free thermodynamic software makes it possible for more researchers to contribute to the development of models and algorithms.
With better models, faster software and well documented high quality databases the number of users of Calphad methods will increase significantly.

Download and documentation

OC can be downloaded from http://www.opencalphad.org
or http://www.github.org/sundmanbo/opencalphad
The installation requires that you compile the program using GNU Fortran 4.8 or similar. There are some advice and a few 100 pages of documentation but if you are not familiar with Fortran please ask a local guru for help. You may also have to install GNUPLOT which is free.

Download and documentation

OC can be downloaded from http://www.opencalphad.org
or http://www.github.org/sundmanbo/opencalphad
The installation requires that you compile the program using GNU Fortran 4.8 or similar. There are some advice and a few 100 pages of documentation but if you are not familiar with Fortran please ask a local guru for help. You may also have to install GNUPLOT which is free.

- Sundman B, Kattner U R, Palumbo M and Fries S G, *Open Calphad - a free thermodynamic software*, Integrating Materials and Manufacturing Innovation, **4**:1 (2015)
- Bo Sundman, Xiao-Gang Lu and Hiroshi Ohtani, *The implementation of an algorithm to calculate thermodynamic equilibria for multi-component systems with non-ideal phases in a free software*, in Computational Materials Science, **101** (2015) 127-137

Thanks for listening