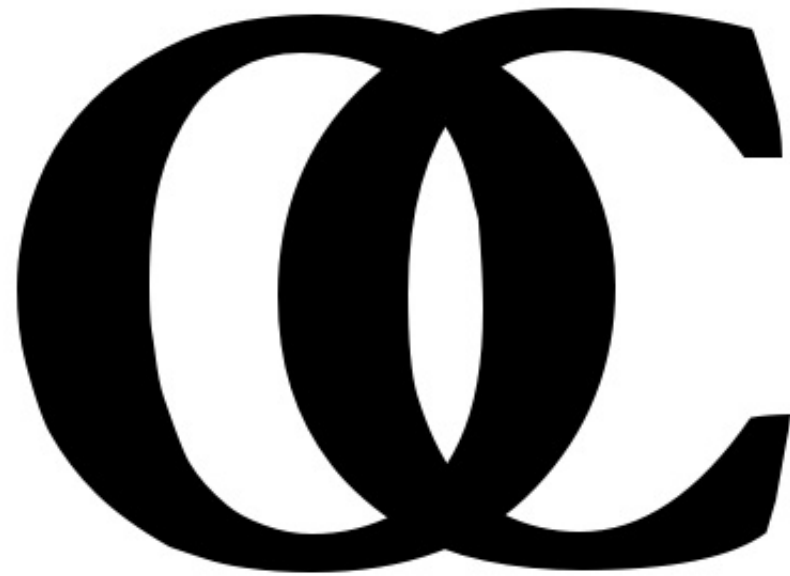


Open Calphad Software



Software layout

1. Model package (FTMP)
2. Equilibrium calculation (PMLFKT)
3. TQ interface
4. Command oriented user interface
5. STEP, MAP and POST
6. Assessment
7. Other contributions (models, phase field, GUI, alternative minimizers etc).

Collaborations

- ICAMS (Suzana) will organize the development of steps 1 to 4.
- A website will be created (OpenCalphad) with a discussion forum and downloadable official versions (first release in 2012).
- Contributions from other groups are welcome after first release. They will be integrated in official releases after testing. The contributor will keep the copyright to his contribution.
- The free software can be integrated in commercial software but the company may need to pay royalties to the copyright owner(s) of the free software. This has to be organized.
- No support nor maintenance is guaranteed for the software.

Free Thermodynamic Model Package (FTMP)

Bo Sundman

2011.06.01

Data structure and software

- Arrays with records for elements, species and phases
- The array elements are defined F90/95 structures
- The sizes of the arrays are predefined and links to the records are indices in the arrays.
- TP functions similar to Thermo-Calc have been implemented
- Data can be read from TDB files or entered interactively.
- CEF model with magnetism for multicomponent systems implemented.
- No general minimizer but one can calculate G and derivatives for a single phase or find the gridpoints for a global minimum
- The datastructure can be saved to and read from a file as "unformatted".

F90/95/03/08 datastructures

With F90 the programmer can define his own data structures called TYPE (in other languages they are called "classes" or "records"). Such a type can contain characters, integers, reals and other user defined types. Arrays can be "allocatable" meaning their sizes can be determined during execution.

Pointers in F90 are not the same as pointers in other structured languages. They are more like aliases for another variable. Allocatable arrays inside a type are declared as pointers.

One can develop modules with data-structures that are protected (private) as well as subroutines/ functions. Private data can only be accessed or set by subroutines in the module.

Parallel processing 1

I have tested the global startpoint generator in a parallel execution but some modifications are needed for general equilibrium calculations. I have understood is that one should try to keep all data belonging to one thread together and above all prevent two different threads to write in the same memory location as they will then destroy their results. But the threads can read from the same memory.

Thus data that are constant during the execution like **parameter coefficients** can be shared but data like **parameter values**, which may depend on temperature which can be different for different processes, must be separate.

Even the error code must be different in each thread as an error may not be fatal but might be handled at a higher level.

Parallel processing 2

Massive parallel processing is needed for using thermodynamic data in phase field simulations for example. In the phase field simulation G and derivatives for the same phase may be calculated on different threads for different compositions and temperatures at the same time.

The assessment procedure can benefit a lot from parallel processing as each experimental equilibrium can be calculated by separate threads. Only when coefficients are changed one has to run on a single thread.

During mapping the lines may be calculated in parallel by separate threads.

The global startpoint generator can calculate the gridpoints for each phase in separate threads.

New things

- Prepared for parallel processing.
- More properties can be handled, also properties like mobilities and new ones like composition dependent individual Bohr magneton numbers.
- Much more flexible "disordered" part of an ordered phase
- Prepared for multiple permutations of parameters for ordered phases. A four component 4 sublattice ordered fcc phase has 256 parameters but only 35 unique ones.
- A default reference phase for the pure elements
- A clean data structure makes it possible to implement new models and better and faster algorithms
- Written in Fortran 90/95/03/08

Aim

- An open source thermodynamic package at high scientific level for general equilibrium calculations.
- Open for all interested to implement and test new models and algorithms
- A TQ interface for application software.
- A much more flexible assessment module including step/map/post
- Interfacing with ab initio software

Element

- Character symbol*2,name*24,refstate*24
- Double mass,h298_h0,s298
- Integer splink, status, alphaindex, refstatesymbol

The element can have a "name" in addition to its symbol but the name cannot be used for anything except listings.

The elements are stored in the array `ellista` below in the order they are entered and in the array elements the alphabetical order of the elements is maintained. The electron is index -1 and the vacancy is index 0.

All references to elements from species records are to the element index in `ellista`.

Private means these arrays are not accessible outside the module.

```
TYPE(ftmp_element), private :: ellista(-1:maxel)
```

```
Integer, private :: elements(-1:maxel)
```

Species

- Character symbol*24
- Double mass, charge
- Integer noofel, status, alphaindex
- Integer, dimension(:), pointer :: ellinks
- Double dimension(:), pointer :: stoichiometry

The array "splista" contains the species in the order they were entered. The array "species" contains indices to splista in alphabetical order of the species. Alphaindex has the alphabetical order of the species. The ellinks array has indices to ellista of the elements in the species and stoichiometry has the stoichiometric factor. All references to species for phases and parameters are to the index in splista which is always the same even if new species are entered.

```
TYPE(ftmp_species), private :: splista(1:maxsp)  
Integer, private :: species(1:maxsp)
```

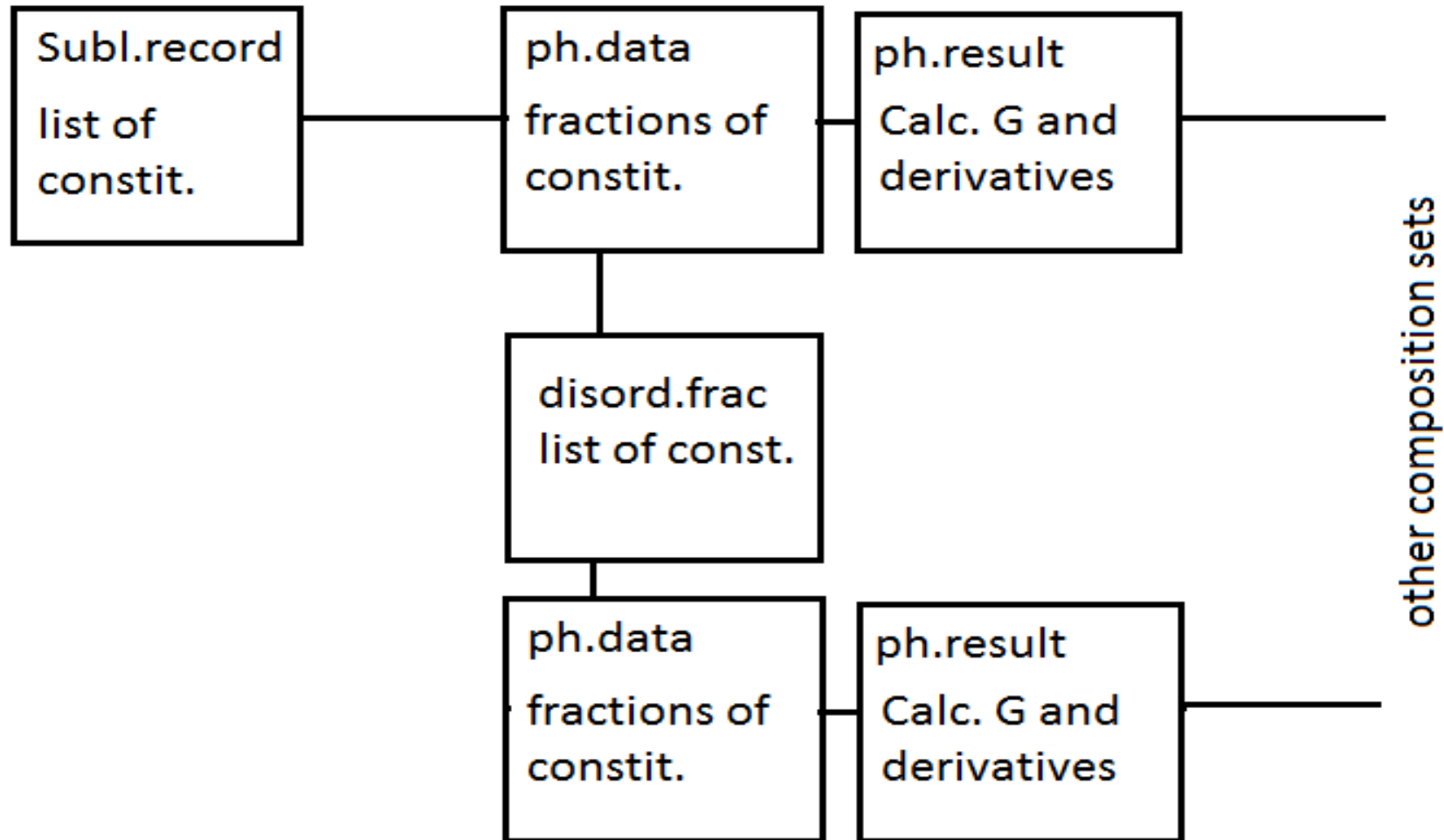
Phase

- Character name*24, models*72,phletter*1
- Integer status1, alphaindex, noofcs, nooffs
- Integer, dimension(3) :: endmember

Noofcs is the number of composition sets and nooffs is the number of fraction sets. The reason to have 3 endmember lists is that each fraction set must have its own (currently only 2 fraction sets implemented). Index 0 below is to the STABLE_ELEMENT_REFERENCE phase.

```
TYPE(ftmp_phase), private :: phlista(0:maxph)
Integer, private :: phase(0:maxph)
```

Fraction records



Sublattice

- Integer noofsubl, cslink, tnooffr
- Integer, dimension(maxsubl) :: nooffr
- Double, dimension(maxsubl) :: sites
- Integer, dimension(:), pointer :: constitlist

Noofsubl is the number of sublattices. Cslink is the index to the constituent fraction record corresponding to this sublattice record. Tnooffr is the total number of constituents.

Nooffr is the number of constituents in each sublattice. Constitlist contains indices to splista of the constituent species from the first sublattice to the last.

The arrays nooffr and sites should be allocatable and declared as pointers but when I wrote this part I did not know so much about allocatable arrays.

The index to a sublattice record is the same as to its phase record. Maybe they should be a single record.

```
TYPE(ftmp_sublattice), private :: sublista(0:maxph)
```

Constituent fractions 1

- Integer next,phlink,status2
- Character*4 prefix, suffix
- Integer, dimension(:), pointer :: constat
- Double, dimension(:), pointer :: yfr,maxmin

One record of this type for each composition set of a phase. During calculations each parallel processes can have its own copy of this if the fractions are different.

Prefix and suffix are added to the phase name for different composition sets.

Constat has a bit set if a constituent is suspended.

Yfr is an array of fractions and the dimension of yfr is the same as for constlist in the sublattice record.

```
TYPE(ftmp_constituent_fractions), private, dimension(2*maxph) ::  
phase_pardata
```


Constituent fractions 2

- Double abnorm(2)
- Double dimension(:), pointer :: mmyfr
- Double, dimension(:), pointer :: sites
- Type(ftmp_fraction_set) :: disfra

Abnorm(1) is the current number of real atoms per formula unit for the current composition. Abnorm(2) is the current mass.

Mmyfr is to store min/max of fractions (min fraction of C for carbides for example). For ionic liquids there must also be the current number of sites (and derivatives). Disfra contains information of a disordered fraction set if any.

Disordered fraction set record

- Integer latd, ndd, tnoofxfr, tnoofyfr, phdalink, totdis
- Character id*1
- Double, dimension(:), pointer :: sites
- Integer, dimension(:), pointer :: splink
- Integer, dimension(:), pointer :: y2x
- Double, dimension(:), pointer :: dxidyj

This record contains information of a second set of fractions for a phase. Corresponds to the "disordered" part of an ordered phase.

Id is the suffix for parameters for this fraction set (default "D")

The other variables contain information needed to calculate the disordered fractions from the constituent fractions.

Phdalink is index of the phase_pardata record for the disordered fractions.

Totdis indicates if the phase can be totally disordered (like fcc) or not (like sigma)

Disordered fractions

FCC ordering

$(\text{Al, Cr, Fe, Nb, Ni})_{0.75}(\text{Al, Cr, Fe, Nb, Ni})_{0.25}(\text{C, Va})_1$

$(\text{Al, Cr, Fe, Nb, Ni})_1(\text{C, Va})_1$

Sigma ordering

$(\text{Cr, Fe, Ni})_{10}(\text{Cr, Nb})_4(\text{Cr, Fe, Nb, Ni})_{16}$

$(\text{Cr, Fe, Nb, Ni})_{30}$

Double ordering

$(\text{Al, Fe})_{0.25}(\text{Al, Fe})_{0.25}(\text{Al, Fe})_{0.25}(\text{Al, Fe})_{0.25}(\text{C, Va})_{0.25}(\text{C, Va})_{0.25}(\text{C, Va})_{0.25}(\text{C, Va})_{0.25}$

$(\text{Al, Fe})_1(\text{C, Va})_1$

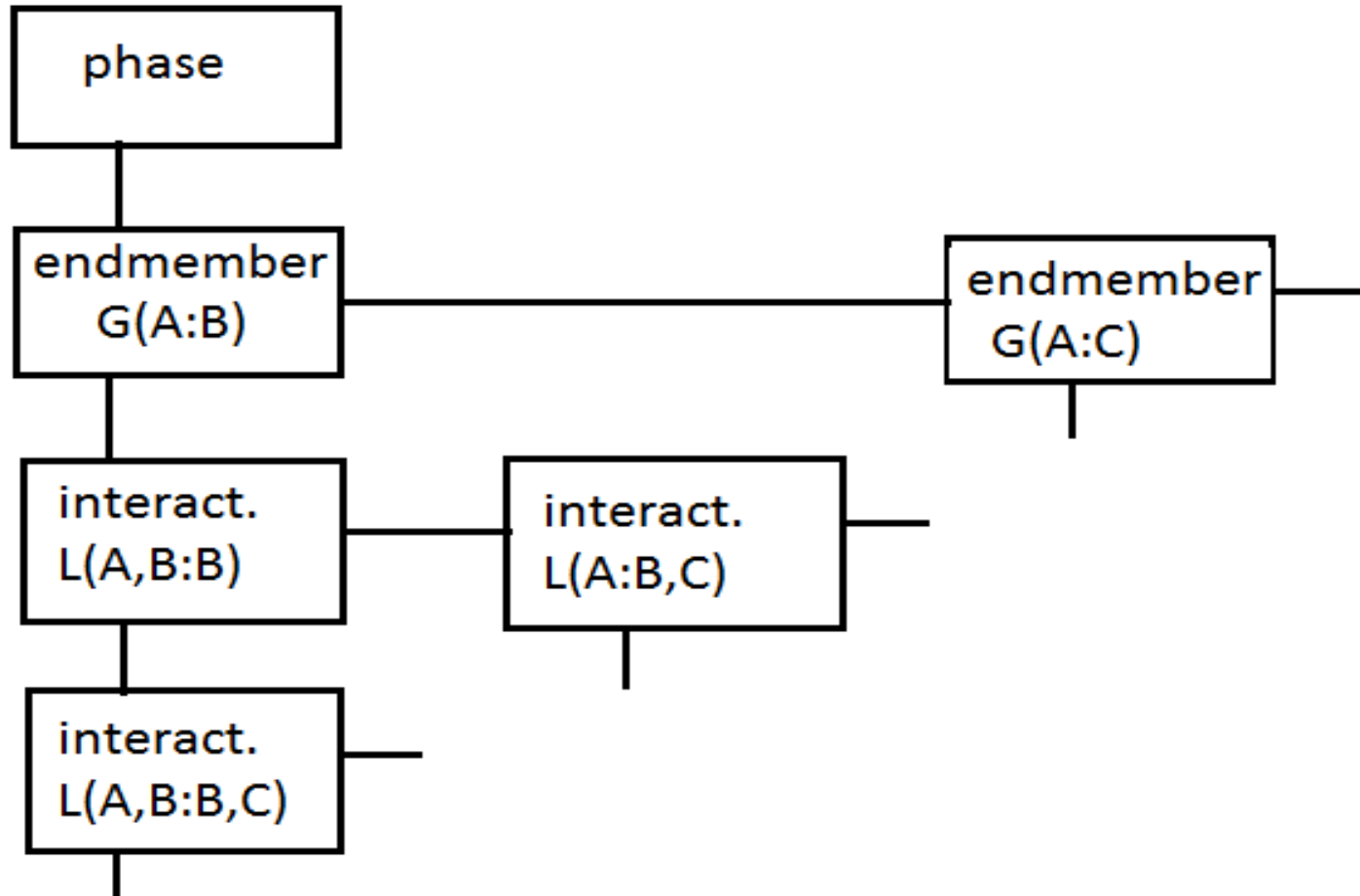
Result record

- Integer nprop
- Integer, dimension(:), pointer :: listprop
- Double amount(2),dgm
- Double dimension(:,:), pointer gval G, G.T, G.P etc
- Double dimension(:,:::), pointer dgval G.Yi, G.T.Yi, G.P.Yi
- Double dimension(:,:), pointer d2gval G.Yi.Yj (symmetrical)

This record is used during calculations and keeps the final values after a calculation. Nprop is the number of properties, property 1 is G. listprop is the type of property 2 etc. The result record has always the same index as the constituent fraction record. It is not private in order to make it easier for application software to get the values.

TYPE(ftmp_results), dimension(2*maxph) :: phase_parres

Parameter tree



Parameter syntax

The datastructure can handle any type of property **F** that are phase and composition dependent

$$F=F(T,P,x)$$

F can be G for Gibbs energy, M for mobilities etc.

This property can have parameters like

$$F\&element(phase, T, constituents)$$

Element is a possible additional specifier.

Phase and *constituents* specifies the fractions which should be multiplied with this parameter.

Surface energies cannot be handled with this syntax but maybe one can divide them between phases?

Endmember record

- Integer next, inter, proplink, noofpermut, phaselink
- Integer, dimension(:, :), pointer :: fraclinks

These records store endmember parameters and are the root of the interaction parameter tree. Next is index of next endmember for this phase, inter is index of an interaction record, proplink is index to the first property record, noofpermut is the number of permutations of this record (when ordered phases like L12), phaselink is index of phase record.

In fraclinks(s,k) the constituent index of the fraction in sublattice s is stored for k:th permutation. The permutations are very important to handle parameters in complex ordered phases like L12, L10 etc

Note that all endmembers for all phases are stored in the same array. The next link is necessary to know which endmembers that belong together for the same phase.

```
TYPE(ftmp_endmember), private, dimension(maxendm) :: emlista
```

Interaction record

- Integer next, higher, proplink, noofpermut
- Integer, dimension(:), pointer :: sublattice
- Integer, dimension(:), pointer :: fraclink

The interaction records form a binary tree. Its root is an endmember record which specify one constituent in each sublattice. The interaction records specify one additional interacting constituent in one sublattice. The next variable is the index of the next interaction on the same level, i.e. replacing the interaction in this record. The higher variable is a link to an interaction record on a higher level, i.e. including the interaction in this record.

Proplink is index to the property parameters for this interaction.

Note that one can have permutations also of interactions and the number of permutations can be different for each

```
TYPE(ftmp_interaction), private, dimension(maxintm) :: intlista
```


Permutations for ordered phases

End member A:A:A:A, 1 permutation

1st Interaction B, L(A,B:A:A:A) has 4 permutations

A,B:A:A:A; A:A,B:A:A; A:A:A,B:A; A:A:A:A,B

2nd interaction B, L(A,B:A,B:A:A) varying number of interaction depending on which sublattice the first interaction:

If A,B:A:A:A then 3: A,B:A,B:A:A; A,B:A:A,B:A; A,B:A:A:A,B

If A:A,B:A:A then 2: A:A,B:A,B:A; A:A,B:A:A,B

If A:A:A,B:A then 1: A:A:A,B:A,B

If A:A:A:A,B then none

If 2nd interaction C, L(A,B:A,C:A:A) there are 12 permutations etc.

Property record

- Character reference*16
- Integer next, proptype, degree, extra, refix
- Integer, dimension(:), pointer :: degreelink

The property records are linked from endmembers and interaction records. Each property record has data for one property but as there can be several there is a next link. Reference is a textual reference for the value of the property. Proptype is the property type, 1 is G, 2 is TC etc. In some cases a property depends on an element like the mobility, then extra is used to store that information. Refix is a reference index.

Degree link have indices to TP functions for different degrees (for interaction parameters). Inside the TP functions one has the temperature ranges and also the currently calculated values which must be separate in each parallel process.

```
TYPE(ftmp_property), private, dimension(maxpty) :: ptylista
```

And so onmany more data
structures

FTMP:

Gibbs energy J/mol -8.24949454E+04

Helmholtz energy J/mol -8.24949454E+04

Enthalpy J/mol -1.72397209E+04

Internal energy J/mol -1.72397209E+04

Entropy J/mol/K 6.52552245E+01

Volume m3 0.00000000E+00

Heat capacity J/mol/K 3.21476087E+01

Thermal expansion 1/K 0.00000000E+00

Bulk modulus 1/Pa 0.00000000E+00

First partial derivative with respect to AL in sublattice 1 of

G -8.33167320E+04

H -3.71321131E+04

G.T -4.61846189E+01

G.P 0.00000000E+00

Second partial derivative of Gibbs energy with respect to also

AL in sublattice 1..... 8.26000449E+04

TAB:t-d

Gibbs energy: -8.24949453E+04

Helmholz energy: -8.24949453E+04

Enthalpy: -1.72397208E+04

Internal energy: -1.72397208E+04

Entropy: 6.52552246E+01

Molar volume: 0.00000000E+00

Thermal expansivity: 1.00000000E+20

Isothermal compressibility: -1.00000000E+20

Heat capacity at constant pressure: 3.21476082E+01

First partial derivative with respect to AL in sublattice 1

of Gibbs energy: -8.33167318E+04

of enthalpy: -3.71321121E+04

of entropy: 4.61846197E+01

of volume: 0.00000000E+00

Second partial derivative of Gibbs energy with respect to also

AL in 1: 1.03310268E+05

FTMP: l da f-nod

Phase: FCC_NOD 3 0.750 0.250 1.000

Constituents: AL, CR, NI; AL, CR, NI; VA;

Magnetic model according to Inden, antiferromagnetic factor: -3

Magnetic function below the Curie temperature:

$1 - 0.860338755 * TAO^{(-1)} - 0.17449124 * TAO^{(-3)} - 0.00775516624 * TAO^{(-9)}$

$- 0.0017449124 * TAO^{(-15)}$

Magnetic function above the Curie temperature:

$- 0.0426902268 * TAO^{(-5)} - 0.0013552453 * TAO^{(-15)} - 2.84601512E-04 * TAO^{(-25)}$

$G(\text{FCC_NOX}, \text{AL}:\text{CR}:\text{VA}; 0) = 298.15 + \text{AL}3\text{CR}; 6000 \text{ N REF9}$

$G(\text{FCC_NOX}, \text{AL}:\text{NI}:\text{VA}; 0) = 298.15 + \text{AL}3\text{NI}; 6000 \text{ N REF9}$

$G(\text{FCC_NOX}, \text{CR}:\text{AL}:\text{VA}; 0) = 298.15 + \text{ALCR}3; 6000 \text{ N REF9}$

.....

$G(\text{FCC_NOD}, \text{CR}, \text{NI}:\text{VA}; 0) = 298.15 + 3 * \text{L04CRNI}; 6000.00 \text{ N REF9}$

$G(\text{FCC_NOD}, \text{CR}, \text{NI}:\text{VA}; 1) = 298.15 + 3 * \text{L14CRNI}; 6000.00 \text{ N REF9}$

$G(\text{FCC_NOD}, \text{AL}, \text{CR}:\text{VA}; 0) = 298.15 + \text{L04ALCR}; 6000.00 \text{ N REF9}$

$G(\text{FCC_NOD}, \text{AL}, \text{CR}:\text{VA}; 1) = 298.15 + \text{L14ALCR}; 6000.00 \text{ N REF9}$

FTMP: Please give overall composition:

Give mole fraction of C /.1/: Give mole fraction of CR /.99/: Give mole fraction
of FE /.79/: Give mole fraction of MN /.14/: Give mole fraction of MO /.13/: Give
mole fraction of N /.12/: Give mole fraction of NI /.11/: Give mole fraction of
SI /.04/: Give mole fraction of TI /.03/: Temperature /1000/: Pressure /100000/:

Calculated 41312 gridpoints in 3.276020E-01 seconds

Merging 1 6 -9.8223E+00 -9.8215E+00 -9.5184E+00 -9.8366E+00

Merging 2 10 -5.4006E+00 -5.3924E+00 -5.3878E+00 -6.5203E+00

Merging 4 7 -5.9829E+00 -5.8516E+00 -7.6261E+00 -5.6920E+00

Total CPU time for global minimum is seconds: 3.432020E-01

Global minimum with 7 gridpoints:

HCP_A3 , amount 3.1500E-02, with constitution:

1.000E-30 1.000E-30 1.000E-30 5.000E-02 1.000E-30 1.000E-30 1.000E-30

9.500E-01 9.524E-01 4.762E-02 1.000E-30

SIGMA , amount 3.3248E-02, with constitution:

5.021E-01 1.000E-30 4.979E-01 9.979E-01 1.000E-30 2.051E-03 4.979E-01

5.000E-01 1.000E-30 1.000E-30 1.000E-30 2.051E-03

POLY_3:c-e

Calculated 34695 grid points in 0 s

Found the set of lowest grid points in 0 s

Calculated POLY solution 1 s, total time 1 s

POLY_3:l-st ph

PHASE STATUS DRIVING FORCE MOLES

V3C2 ENTERED 0.00000000E+00 2.30128809E-03

M23C6 ENTERED 0.00000000E+00 4.13806953E-02

FCC_A1#3 ENTERED 0.00000000E+00 2.08456152E-02

FCC_A1#1 ENTERED 0.00000000E+00 5.51309875E-01

BCC_A2 ENTERED 0.00000000E+00 3.84162531E-01

HCP_A3 ENTERED -1.36202105E-02 0.00000000E+00

SIGMA ENTERED -1.72294479E-02 0.00000000E+00

M6C ENTERED -7.84501091E-02 0.00000000E+00

CHI_A12 ENTERED -1.25284348E-01 0.00000000E+00

FCC_A1#1 Status ENTERED Driving force 0.0000E+00

In the pipeline and later

- Implementing FTMP in PMLFKT (Leo Lukas version of POLY) for general equilibrium calculations
- Heat capacity models down to 0 K
- A new magnetic model with individual Bohr magneton numbers
- Implement the ionic liquid model
- Parameter permutations for ordered phases like L12 and D03
- A TQ interface
- Implement a disordered part for ionic crystalline phases
- A step and map procedure
- Interface with gnuplot for postprocessing
- Assessment module

Speed

On my portable Thermo-Calc took less than a second and BMP took 0.33 seconds for the global minimization of the test 10 component system. When I tested BMP and the global minimization on linux, using the gfortran and ifort compilers (on a slower computer than my portable PC) the BMP global minimization took 0.7 seconds and Thermo-Calc took more than a second.

Including parallel processing the BMP code would be still faster and BMP is developed with parallel processing in mind.

The aim of BMP is not to be as fast as possible but to be as maintainable as possible.

Developments

- Commercial companies can use FTMP including a subroutine to read encrypted databases.
- New structure of the result file (POLY files). Possibly written as a text files and thus independent of hardware and OS and easier to interface to gnuplot or other free graphical software.
- Non-equilibrium calculations like T-zero, para-equilibria, solidification etc.
- Interfacing with DICTRA, phase field applications, adding new types of data in the model package like surface energies, viscosities etc.
- New database for unaries
- Free database for binaries
- Graphical User Interface

And so onmany more facilities