THERIAK-DOMINO User’s Guide

Vers. 2023.06.11

Christian de Capitani

Konstantin Petrakakis

"Si quis laetali serpentis, saucius, ictu
Cepit, et ignara, toxica saeva, manu,
Nec mora pro degustato, tactoque veneno,
Laeta propinabis pocula Theriacae,
Ista dabit vigili, Caesar, medicina soporem,
Atque medela gravis sumpta doloris erit,
Hactenus exposui vires, nunc tempora dicam,
Vipera cum certo sit capienda modo."

Galenos: Libellus de Theriaca ad Pisonem (cited from: Daniëls ,1911)

Introduction

What does the theriak/domino suite do?

The **theriak/domino suite** includes several programs for thermodynamic calculations and plotting of results in PostScript® format. These programs are:
theriak, the heart of the suite, calculates the stable mineral assemblage and phase compositions for a given rock (bulk composition) at specified P, T conditions or along a specified PT-path.
domino may calculate

- Equilibrium assemblage phase diagrams for selectable axes (P, T, components activities etc.)
- Pseudo-binary or pseudo-ternary phase diagrams.
- Pixelmaps for all thermodynamic variables.
- Isolines of several thermodynamic properties for bulk rock, non-solution and solution phases.

therbin calculates binary phase diagrams at constant P or T.
therter calculates ternary phase diagrams at constant P and T.
thalia calculates phase thermodynamic parameters as functions of T or P or (binary) composition.
thermo calculates thermodynamic parameters at T and P and tables of V and G.
guzzler uses the output graphics files produced by theriak, domino or thalia as input and cleans up labels, making graphics information more readable.
explot transforms files containing graphics information from the above programs to “ready to use” PostScript files that can be seen on screen or printed to a hard copy.
makemap uses the pixelmap information from domino and incorporates a grey pixelmap image of any variable into a PostScript® file and a portable grey map-file (pgm).
plotxy prepares plots of XY diagrams by selecting as axes among the numerous thermodynamic variables involved in a multi-step calculation done by theriak.
The theriak/domino suite is written in standard FORTRAN; it is developing and made available to the scientific community since 1987 by C. de Capitani. Executable files for the most common operating systems (Linux; Mac-OS X; Windows) have been created by using the gfortran compilers under the GNU Public Licence or the freely available ifort Intel compiler. Current and recent versions add more convenience to the user:

- On-line help, besides a reworked user guide.
- Tailoring file names as well as installation and working directories according to user preferences.
- Scheduling time-consuming, unattended runs.

- Compatibility with currently available thermodynamic databases (Standard data and activity models).

Why use the theriak/domino suite?

Results calculated by theriak/domino suite are not better or worse than results calculated by other comparable programs. The quality of results depends merely on the quality of the thermodynamic data used as well as to user personal (subjective) choices rather than on any modern software package. Based on its approach to equilibrium by Gibbs free energy minimization (c.f. de Capitani & Brown, 1987) rather, than solving complex and large equation systems, the theriak/domino software is exceptionally fast. The various types of phase diagrams are calculated and plotted without user intervention that might be a source of serious errors.
Fig. 1 demonstrates these properties for a complex equilibrium assemblage diagram including 185 stable reactions. Calculation and plot of this “ready-to-use” diagram has been carried out with a 1.4 GHz Notebook running Windows XP in less than 30 minutes!

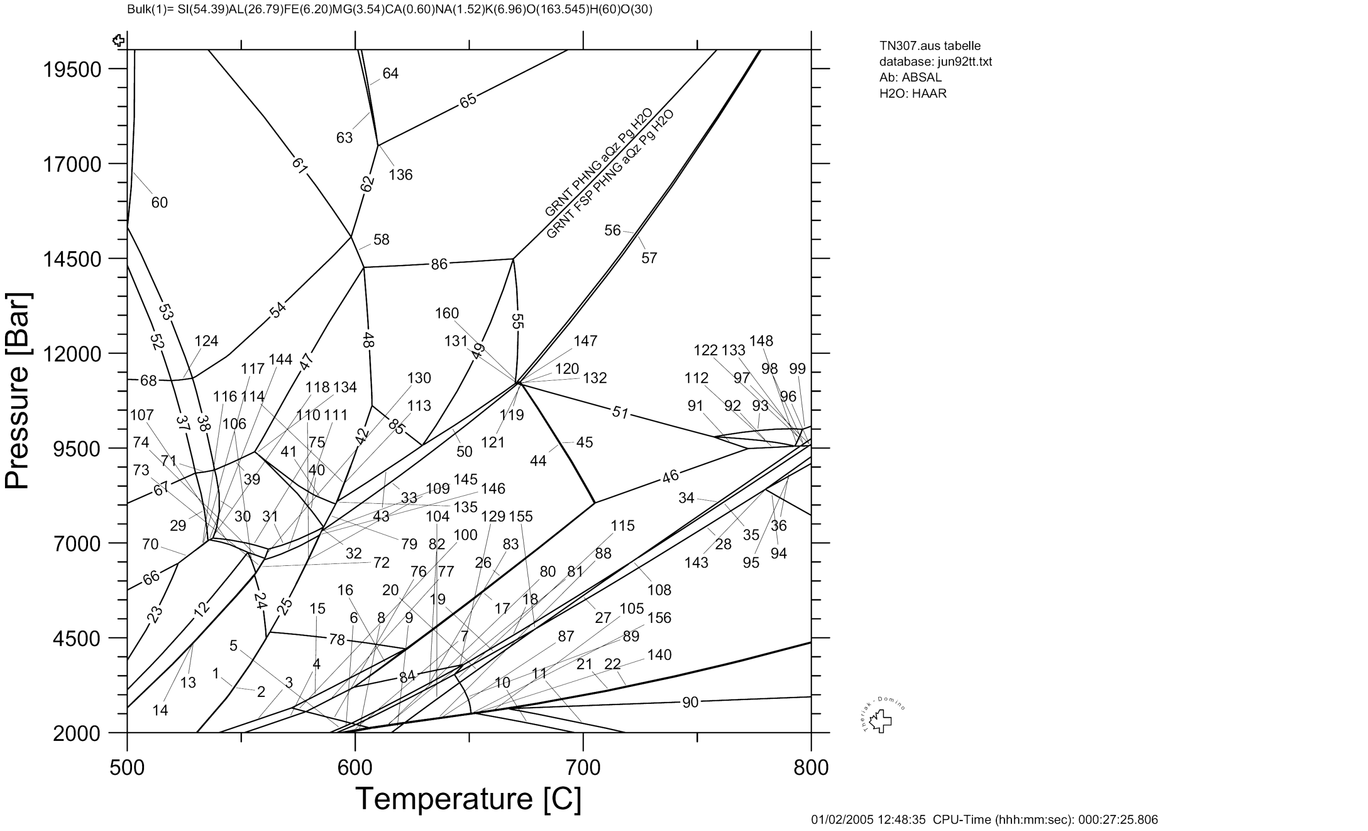

Fig. 1: Equilibrium assemblage diagram for bulk composition TN307 from Nagel (2002), Metamorphic and structural history of the southern Adula nappe (Graubünden, Switzerland), Table 2. PhD thesis Basel,

Terms of use

The executable files of the theriak/domino suite as well as the source code for programmers and developers are available for free download and use for non-commercial purposes, and as of 2022 are available on Git Hub under a GNU GPLv3 open-source license. The use and modification of the theriak/domino suite underlies solely the responsibility of the user. The source developer and distributor C. de Capitani and code contributors do not undertake any responsibility for any results produced by the user. By using or modifying the theriak/domino suite, the user implicitly agrees, that he or she has accepted these terms of use.

Download THERIAK-DOMINO software.

Releases as of 2023.03.06 are distributed and downloadable from the main Theriak-Domino Git Hub site: https://github.com/Theriak-Domino/theriak-domino

Select a release package for your specific OS. Builds with programs for macos (Intel), macos (Apple Silicon), Linux (Ubuntu) and Windows are available through the Git Hub release page. Download and unzip the file appropriate for your system. Binaries and associated files for installation are included in the Programs directory. Documentation is found in the Documentation directory. **Follow the instructions in Documentation/readme for installing the programs on your system.** For the Holland & Powell database (2011; and updates), it is recommended to obtain up-to-date database files from the associated Databases Git Hub site: https://github.com/Theriak-Domino/Databases Examples are found in the Examples Directory.

Older releases can be obtained from the Theriak-Domino site maintained by C. de Capitani: http://titan.minpet.unibas.ch/minpet/theriak/theruser.html

for Mac OS X and UNIX

TheriakDominoMAC/Programs.zip : Programs.
TheriakDominoMAC/Working.zip : Example of Working directory with databases.
TheriakDominoMAC/Documentation.zip : Documentation.
TheriakDominoMAC/Examples.zip : Examples.

for Windows

TheriakDominoWIN/Programs.zip : Programs.
TheriakDominoWIN/Working.zip : Example of Working directory with databases.
TheriakDominoWIN/Documentation.zip : Documentation.
TheriakDominoWIN/Examples.zip : Examples.

Download the zipped folders and expand them. (double-clicking should work, else download a free unzip/expander software)

Information on installing and running are in the “readme” file.

Getting started

Getting started with theriak

Consider a bulk composition of 10 moles of muscovite, 10 moles of paragonite with excess SiO_2 and water. Calculate the equilibrium assemblage at 600 °C and 4000 Bar.
Write the following two lines at the top (not counting the comment lines) of the dat-file:

```
600     4000
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

```
start theriak.

database                 JUN92.bs
type of calculation           no
```

The result will be a Feldspar coexisting with white mica, quartz and water.
Feldspar: $Ab_{0.94}An_{0.04}$
Mica: $Mu_{0.70}Pg0$
For more detailed information, see also the commented run of theriak in commented_theriak.pdf

With the same composition do multiple calculations between 650 and 800 °C at 4000 Bar.

```
start theriak

database                      JUN92.bs
type of calculation           loop
Temperature and Pressure 650    4000
Temperature and Pressure 700    4000
Temperature and Pressure 800    4000
Temperature and Pressure (CR)
```

Getting started with domino

Consider a bulk composition of 10 moles of muscovite, 10 moles of paragonite with excess SiO_2 and water.
Calculate a P-T equilibrium assemblage diagram, using the database JUN92.bs
Temperature range: 500 - 900 °C
Pressure range: 1000 - 13000 Bar

Write the following two lines at the top (not counting the comment lines) of the dat-file:

```
600     4000
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

```
start domino.

database                      JUN92.bs
```

```
X-axis                  TC   500    900
Y-axis                  P    1000    13000
Calculation type             .
labeling of reactions        1
```

For the X- and Y-axis insert several blanks between the parameters.
As calculation type enter "." (dot)
Domino produces a graphics file (plt-file, usually called coplot). In order to label the curves with the assemblages, this must be processed by the program guzzler.

```
start guzzler

graphics file name      coplot
size of labels          (CR)
option                  (CR)
```

(CR) ("Carriage return", or "enter"): Use the default values
Guzzler produces a new graphics file (cln-file, usually called clean). this can be translated to plot-files, usually called plot.ps and plot.svg

```
start explot

graphics file name      clean
```

The resulting plot-files can be displayed (and modified) e.g. with ghostview, Adobe Illustrator, Corel Draw, Inkscape etc.

Getting started with therter

The program therter calculates ternary phase diagrams. The system must be strictly ternary, i.e. all phases must be linear combinations of the three ternary endmembers.
Once the three ternary endmembers are defined, therter will automatically exclude all phases outside this ternary.
We will calculate the phase relations of the K-feldspar-Albite-Anorthite ternary.
The bulk composition must contain at least all elements needed for the ternary system. Use e.g. the same dat-file as in the previous examples, and add Calcium.

```
600     4000
0   CA(1)SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

```
start therter

database                JUN92.bs
endmember 1                  Ab
endmember 1                  An
endmember 2                  Kfs
Temperature and pressure 800   2000
number of seeds              (CR)
X-scan-density….             (CR)
```

Therter produces a graphics file (plt-file, usually called plot). **This cannot be processed by guzzler.** It can be translated to a plot-file by explot.

start explot

```
graphics file       plot
```

The resulting plot-files can be displayed (and modified) e.g. with ghostview, Adobe Illustrator, Corel Draw, Inkscape etc.

Getting started with therbin

The program therbin calculates binary phase diagrams. The system must be strictly binary, i.e. all phases must be linear combinations of the two binary endmembers.
Once the two binary endmembers are defined, therbin will automatically exclude all phases outside this binary.
We will calculate the phase relations of the K-feldspar-Albite binary.
The bulk composition must contain at least all elements needed for the binary system. Use e.g. the same dat-file as in the previous examples:

```
600    4000
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

start therbin

```
database            JUN92.bs
endmember 1              Kfs
endmember 2              Ab
Y-variable               TC   200   800
pressure            2000
X-scan-density….         (CR)
```

Therbin produces a graphics file (plt-file, usually called plot). **This cannot be processed by guzzler.** It can be translated to a plot-file file by explot.

start explot

```
graphics file       plot
```

The resulting plot-files can be displayed (and modified) e.g. with ghostview, Adobe Illustrator, Corel Draw, Inkscape etc.

Getting started with thermo

For a given p and T, THERMO calculates ΔG (apparent Gibbs free energy, = ΔH - T.S), ΔH, S, Cp and V. Also, for each thermodynamic function (Cp, volume, disorder, lambda transition) its contribution to the total ΔG, ΔH.
The bulk composition is not used by thermo. However, the dat-file must be present. Use any composition e.g.:

```
600    4000
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

```
start thermo

phases                   And   Ky   Si
phases                   v
phases                   end
```

Getting started with thalia

THALIA plots thermodynamic functions versus T or P, or ΔG (total, configurational, excess or mixing), the chemical potential, or the activity of a phase along a binary link of a solution phase. The bulk composition is not used by thermo. However, the dat-file is must be present. Use any composition.

```
start thalia

database                 JUN92.bs
X-variable …..           X
solution-name            FSP
endmember1                    1  0  0
endmember2                    0  1  0
Y-variable                    GMIX
temperature and pressure 600   2000
temperature and pressure end
```

Thalia produces a graphics file (plt-file, usually called thplot). This can be processed by guzzler and explot.

```
start guzzler

graphics file            thplot
size of labels           (CR)
option                   (CR)
```

```
start explot

graphics file            clean
```

The resulting plot-files can be displayed (and modified) e.g. with ghostview, Adobe Illustrator, Corel Draw, Inkscape etc.

Making "pixel maps" with domino

Consider a bulk composition of 10 moles of muscovite, 10 moles of paragonite with excess SiO_2 and water.

Temperature range: 500 - 900 ^oC
Pressure range: 1000 - 13000 Bar

Write the following two lines at the top (not counting the comment lines) of the dat-file:

```
600     4000
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

Start domino.

```
database                JUN92.bs
X-axis                  TC   500    900
Y-axis                  P    1000    13000
Calculation type        pix   100   100
```

For the X- and Y-axis insert several blank between the parameters.

domino will create a new folder called "_pixelmaps". If this folder already exists, you can enter a new name or overwrite the existing folder. In this folder the information for each available variable needed for pixelmaps is stored. This can be visualized with the program makemap

start makemap

```
folderNr                1
functionNr                   40
```

This example will produce a grey map (100 x 100) of "vol_solids" in folder "_pixelmaps/".
The file created is a plot-file (usually called plot.ps and plot.svg) and can be displayed e.g. with ghostview, Adobe Illustrator, Corel Draw, Inkscape etc.

Calculations along a PT-path with theriak

Consider a bulk composition of 10 moles of muscovite, 10 moles of paragonite with excess SiO_2 and water.
Make a table with all amounts, compositions, activities and much more along a PT-path from 500 °C/3500 Bar to 800 °C/6000 Bar.
Use the following composition in the dat-file (usually THERIN):

```
600     4000
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

Create a text file (e.g. **ptpath**) with the following directives:

```
TP   500    3500
TP   800    6000    50
```

start theriak

```
database                JUN92.bs
type of calculations         ptpath
```

Theriak writes a (sometimes very large), comma delimited table to the tab- and tcp-files. (usually named loop.table and loop.tecplt). For simple X-Y-plots these tables can be processed by the program plotxy.

start plotxy

```
filename             loop.table
X-axis               3
Y-axis               54,55,56,57,58+59
X-min  X-max         <CR>
```

```
Y-min  Y-max        <CR>
```

plotxy produces a graphics file (plt-file, usually called xyplot). This can be processed by guzzler and explot.

start guzzler

```
graphics file             xyplot
size of labels            <CR>
option                    <CR>
```

start explot

```
graphics file             clean
```

The resulting plot-files can be displayed (and modified) e.g. with ghostview, Adobe Illustrator, Corel Draw, Inkscape etc.

Defining a specific activity for water

In some calculations, specially at higher pressures it may be desired to calculate equilibria or phase diagram with water having a reduced activity. A reduced water activity at HP conditions in magnesian metapelites has e.g. been shown by Chopin C. (1984): Coesite and pure pyrope in high-grade blueschists of the Western Alps: a first record and some consequences. Contrib Mineral Petrol 86:107–118

defining a reduced activity

In the database (dbs-file) we insert a new phase (e.g. STEAM06) after the definition of STEAM. (Note: The name refers to JUN92 database and may be different in others)

```
STEAM06     H(2)O(1)     H2Oa          na
 ST
 CP1
 COM    STEAM[1]    0.6
```

In the above example, the activity is set to 0.6. The name and the abbreviation are chosen to be different from any others in the database. The ST and CP1 lines are empty (or all zeroes) but must be included.

If sufficient H2O is in the bulk composition, then STEAM06 instead of STEAM will become stable and the activity of STEAM will be 0.6.

Don't forget to remove (or comment out) the newly defined phase before you proceed to "normal" calculations. Alternatively, you could give the phase a special code, as is usually done with buffers.

Defining a buffer

Buffers are often used in experimental petrology to control the oxygen activity. These simple buffers are probably not controlling the O2 activity in nature, but calculations with a buffer may give some insights into more complex processes.

defining a buffer phase

As an example, we will consider the hematite-magnetite buffer. This means, that we want to force an equilibrium between hematite and magnetite according to the reaction:

6 hematite = 4 magnetite + O_2

We achieve this equilibrium by defining a new phase with the composition O2, having for each pressure and temperature a Δ_aG of $6\Delta_aG$(hematite)-$4\Delta_aG$(magnetite).

In the database (dbs-file) we insert this new phase (e.g. OHM-Buffer) after the definition of HEMATITE and MAGNETITE. (Note: The names refer to JUN92 database and may be different in others)

```
OHM-BUFFER      O(2)      OHM      *OHM
  ST           0.000     000000.000     000.000000        0.000000
  CP1       0.000000        0.00000           0.00             0.0
  COM      HEMATITE[6]MAGNETITE[-4]     0    HEMATITE
```

Important: The code (4th parameter in the first line) must begin with "*" (star) and defines the name of the buffer. The ST and CP1 lines are empty (or all zeroes) but must be included.
The last line defines the thermodynamic properties of the buffer. The activity variable is set to zero (=not used) and HEMATITE will be excluded from any equilibrium.

For the case, that not sufficient O2 is present in the bulk, we add an additional negative buffer:

```
-OHM-BUFFER      O(-2)      -OHM      *OHM
  ST           0.000     000000.000     000.000000        0.000000
  CP1       0.000000        0.00000           0.00             0.0
  COM      OHM-BUFFER[-1]
```

The code (*OHM) is the same as for the positive buffer.

To use this buffer, the 'use'-variable in the dat-file must contain ",OHM" (Comma, buffer name), Example:

```
600    4000
1    FE(1)O(1)           *,OHM
```

Note: buffer phases can stay in the database. They are only used when called for through the 'use'-variable.

Description of programs

theriak

Theriak is a program that calculates the equilibrium assemblage for a given bulk composition with data from a thermodynamic database. Most programs that attempt this calculation, fail if one or more solution phases are highly non-ideal and show miscibility gaps. A more detailed discussion on why this happens and how this can be avoided, with additional references can be found in de Capitani and Brown (1987). The strategy of THERIAK is a linear programming method designed to produce the correct answer even in extremely complex, non-ideal systems. As a consequence, the program is sometimes rather slow with simple ideal systems. Because there is no absolutely infallible method to solve a nonlinear program, there is no guarantee that the result produced by THERIAK is indeed the chemical equilibrium. So far, however, no indications for any serious misbehavior of THERIAK have been reported.

Filenames associated with Theriak

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
07	dat	THERIN	User's general input file
08	dbs	Run-time defined	Thermodynamic database
09	log	theriak.last	User responses during last run
10	out	OUT	Calculation results to print or view on screen
11	bin	binout	Graphic data for "pseudo-binary system" calculations
12	drv	Run-time defined	"Special calculations" directives to run program
13	tab	loop.table	Table of variable values; input to spreadsheet programs
14	tcp	loop.tecplt	Table of variable values; input to program "TecPlot"
25	inf	loop_info	info for tables of variable values
39	ifnr		Interactive input-file, passed as argument
41,42			Used for some experiments on subduction

(*) Hard-coded file names (cannot be changed)

Theriak Input files

Theriak can be invoked with a command line argument which is the file name containing the interactive input. Used e.g. by the program “dbcheck” to plot functions (fun++), pseudo-binaries (bin++) or reactions (rea++)

Theriak.ini

This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:
`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by THERIAK.

```
$THERIAK-FILES
dat=THERIN              #User's chemical input to program
dbs=Run-time-defined    #Thermodynamic data base to be used
log=theriak.last        #User's responses during last run of program
out=OUT                 #Calculation information to be printed or viewed on screen
bin=binout              #Graphics data for "pseudo-binary system" calculations
drv=Run-time-defined    #"Special calculation" directives for running the program
tab=loop_table          #Table of variable values; input to a spreadsheet program
tcp=loop_tecplt         #Table of variable values; input to "TechPlot"
inf=loop_info           #info for Table of variable values
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt

is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

dat-file

usually called THERIN or THERIN.txt. This file must be present in the working directory and contains the user input for one T and P and the chemical compositions. E.g.:

```
600      4000
```

```
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

dbs-file
This is the thermodynamic database used for the calculations. The name is entered by the user in the interactive input. The file must be in the working directory. E.g.: `JUN92.bs`

drv-file
In the working directory. File containing information for repeated calculations. E.g. on a line between two PT points or changes in bulk composition. Enter "?" in the interactive input for more information.

log-file
In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with CR. (carriage return)

Terminal input
Is the interactive input through the keyboard

Theriak output files

Terminal output
Is the text output of the program to the screen. This is the dialog, help comments, error messages and the calculated equilibria.

out-file
In the working directory. Usually called "OUT". Most of the above "Terminal output" is also written to this file.

bin-file
In the working directory. Usually called "binout". This is a special output file produced in pseudo-binary calculations. (Keyword "bin" in interactive input). It contains the graphics information for the (ΔG^{rel} - X) diagram.
ΔG^{rel} is ΔG relative to a straight line between the first and last composition calculated. This is useful to visualize ΔG^{mix} of complex solutions, with a reduced database (containing just one solution).
The bin-file has a similar format as the DOMINO graphics output and can be converted to a PostScript file with EXPLOT. (It cannot be used as input to GUZZLER)

tab-file
In the working directory. Usually called "loop.table". This file is produced either in pseudo-binary calculations. (Keyword "bin" in interactive input), or when doing multiple calculations with a drv-file. It is a large table with one row for each calculation and columns for (almost) every variable calculated: Bulk composition, total ΔG, density of solids, wt% of H2O in solids, amounts, volumes, molar volumes of all phases, concentrations and activities of endmembers and much more.
This table (comma-delimited) can be read by most spreadsheet programs (e.g. Excel), or statistics software (e.g. StatView).

tcp-file
In the working directory. Usually called "loop.tecplt". This file is contains the same information as the tab-file, but is specially tailored as input to the program "Tecplot".

Interactive input for THERIAK

This information is also available by entering "?" in the program dialogue.

database definition

CR accept <last input> (if any) and continue.

"files" to see the default names, types and descriptions of files associated with the program. You may change these defaults by editing file theriak.ini with a common text editor.

database filename file name of database to be used (dbs-file).

database phase1 phase2 etc. file name of database to be used followed by names of phases to
include in calculations.

type of calculations

CR Accept the last run entry enclosed in < >.

"no" Do no special calculations, just calculate with T and P values defined in your dat-file.

"bin" s s1 Calculate a pseudo-binary in "s" steps (default = 100) using the 'endmember' bulk compositions defined in your dat-file. In this case, a table with all variables is written into the tab- and tcp-file (see note below). G^{rel} (G^{tot} relative to a straight line between the first and last composition calculated) is written into the bin-file
if s1 not zero (or blank): calculate only loop s1 (for testing purposes)

"loop" Prompts for new T and P values after each calculation.

file name: If the input is not recognized as one of the keywords above, it is interpreted as the name of a drv-file. This file drives equilibria calculations, e. g., along a given PT-path. The format of the drv-file is as follows:

“check” Used to test the subroutine “GETVAL”. After equilibrium calculation you can enter phase var (Phase name and variable name) repeatedly. (see program “prtcal.f90”)

format of the drv-file

Each line defines one "action" of Theriak:

"REF" t1 p2 defines the equilibrium assemblage at the reference t1, p2. This is the "reference assemblage" adding the variable "G_overstep" to the files of type "tab" and "tcp".
Example:

```
REF    650    4500
```

"COMP" formula use Defines the bulk composition for further calculations. Example:

```
COMP   SI(3)AL(1)MG(3)K(1)O(12)H(2)    *
```

"ADD" formula Adds the given formula to the current bulk composition. The amounts may be negative, but only so much will be subtracted that all components remain >= 0. Example:

```
ADD   H(20)O(10)
```

"REMOVE" phase p After each of the following calculations, the composition of stable "phase" will be (partly) removed from the bulk. "p" is the percentage of "phase" that is to be removed. Example:

```
REMOVE   GARNET   100
```

"TP" t2 p2 s — Calculates equilibria along a linear TP-path. T and P change between the current values t1,p1 and t2,p2. The current values t1,p1 are either the input from the dat-file or from a preceding TP-directive within the drv-file. "s" is the number of calculation steps (default= 1) along the path. All variable values for each step are written into the tab- and tcp-files. Example:

```
TP   800   5000   10
```

Temperature and Pressure

If the input to "type of calculation" was "loop", T and P can be entered repeatedly.

"end" Same as CR.
T(C) P(bar) start a new equilibrium calculation with new T and P values.

domino

DOMINO calculates almost any type of equilibrium assemblage phase diagram. The user defines two independent variables which may be P, T, activity of a component or a pseudo-binary system. (or, as special case, a pseudo-ternary system). The calculation may be a phase diagram, mapping the stable assemblages, isopleths of mineral compositions, isolines for amounts, densities, volumes of single phases or rock bulk parameters and much more.
the output is a graphics file (plt-file) that may be cleaned and translated to PostScript with the programs GUZZLER and EXPLOT. Optionally the output may also consist of pixmaps for all variables with a fixed grid.
Multiple diagram calculations can be defined with scripts and run as batch-jobs.

Filenames associated with Domino

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
03	spt	script	Domino default script file name
04	job	domjob	Domino default batch job file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
07	dat	THERIN	User's general input file
08	dbs	Run-time defined	Thermodynamic database
09	log	domino.last	User responses during last run
10	out	OUT	Calculation results to print or view on screen
15	plt	coplot	Graphics data; input to GUZZLER or EXPLOT
16	cln	clean	Labeling reactions in graphics files; input to EXPLOT
19	pst	plot.ps	Graphics file in PostScript format
20	fig	pixelmaps	Directory name for pixelmaps (no filename extension)
39	ifnr		Interactive input-file, passed as argument

(*) Hard-coded file names (cannot be changed)

Domino Input files

Domino can be invoked with a command line argument, which is the file name of a script-file. See below.

Theriak.ini
This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory

is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:
`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by Domino.

```
$DOMINO-FILES
dat=xxxx                 #User's chemical input to program
dbs=Run time-defined     #Thermodynamic data base to be used
log=xxxx                 #User's responses during last run of program
out=xxxx                 #Calculation information to be printed or viewed on screen
plt=xxxx                 #Graphics data (e.g. input to GUZZLER)
cln=xxxx                 #Graphics data after label arranging and cleaning (e.g. input to
                          EXPLOT)
pst=xxxx                 #Graphics file in PostScript format
fig=xxxx                 #Directory name for pixelmaps (no filename extension!)
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt

is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

dat-file

usually called THERIN or therin.txt. This file must be present in the working directory and contains the user input for one T and P and the chemical compositions. E.g.:

```
600     4000
0   SI(60)AL(60)NA(10)K(10)SI(100)H(100)O(?)     *
```

dbs-file

This is the thermodynamic database used for the calculations. The name is entered by the user in the interactive input. The file must be in the working directory. E.g.: `JUN92.bs`

script file

Script files may be produced by DOMINO (Keyword "script"). These are then used instead of the interactive input. The first line of script files begin with "script.". A single script may be run with the command:

```
domino < scriptname  (Input Redirection
domino   scriptname  (Command Line Argument)
```

log-file

In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with CR. (carriage return)

Terminal input

Is the interactive input through the keyboard

Domino output files

Terminal output

Is the text output of the program to the screen. This is the dialog, help comments, error messages and a short summary of calculation progress.

out-file

In the working directory. Usually called "OUT". Most of the above "Terminal output" is also written to this file.

plt-file
In the working directory. Usually called "coplot". This file contains the graphics information computed by Domino. (Its format is almost identical to the graphics output of TWQ.) A description of the format can be found in the Explot users guide, keyword "NPLOIG".
The graphics file is usually first run through GUZZLER, which formats the labels on the curves and writes a cln-file, which can be translated to a PostScript file by EXPLOT.

spt-file
Written to the working directory. Script files produced by DOMINO (Keyword "script"). These contain all necessary information for a single Domino run.

job-file
Written to the working directory by DOMINO (Keyword "script"). Contains the commands to run DOMINO with script files.

pixelmaps folder
This folder is created if the keyword "pix" is used as calculation type, and contains files with graymap information for all variables. Phase diagram information is written to "datetime" and " pixinfo". The folder name begins with a "_" to be recognized by the program Makemap that crates Postscript and pgm (portable graymap) images.

Interactive input for DOMINO

This information is also available by entering "?" in the program dialogue.

database definition

CR to use the thermodynamic database used in the last run and run DOMINO interactively.

"files" to see the default names, types and descriptions of files associated with the program. You may change these defaults by editing file theriak.ini with a common text editor.

database filename file name of database to be used (dbs-file).

database phase1 phase2 etc.: file name of database to be used followed by names of phases to include in calculations.

"script" to start creating script files (file type "spt") that are used to run DOMINO unattended (e.g. over-night). For more information about scripting, see below.

domino-scripting

DOMINO can be instructed to calculate many different phase diagrams non-interactively, for example overnight. For each of these phase diagrams, the input normally needed by DOMINO, e.g. name of the database, definition of X and Y axis, calculation parameters etc., must be prepared and saved in a SCRIPT file (spt-file) beforehand. The names of script files may be defined individually, or are generated by default as files named "scriptXX.spt", where XX is an integer between 01 and 99. Each of the script files substitutes for the keyboard, i.e. for your responses to questions that DOMINO normally asks you. So, you may let DOMINO do its job unattended without you being present and answering the questions. The script files may be executed by means of an executable BATCH JOB file.

Structure of the executable batch files

The default name (domjob) of this file is given in "theriak.ini". For DOS-Win it has usually the extension ".bat", which makes it executable. For Unix/Linux/Mac one has to set this attribute before execution, with.

```
"chmod u+x domjob"
```

job-file and spt-file names may be defined individually during execution. The following default names are however proposed by the program:

job name	Script names
default: **domjob**	script001.txt, script002.txt, script003.txt etc
user defined (e.g.): **myplot**	myplot001.txt, myplot002.txt, myplot003.txt etc.

The name of the pixelmap folder (if needed) is always "_(script name)_pix"
If the chosen names exist, job-files are always appended and spt-files always overwritten.

The executable BATCH JOB file "myplot.bat" (DOS, Windows) or "myplot" (Unix and derivates) will contain the following records:

```
domino < myplot001.spt
guzzler  myplot001.plt   myplot001.cln  myplot001.rxn
explot   myplot001.cln   myplot001.ps
domino < myplot002.spt
guzzler  myplot002.plt   myplot002.cln  myplot002.rxn
explot   myplot002.cln   myplot002.ps
etc.
```

The first of these records calls DOMINO and instructs it that the information it needs for calculation of phase diagram myplot001 is provided by script file "myplot001.txt". The graphics data will be written into a plt-file having the name myplot001.plt. This is given as input argument to program GUZZLER in the second line. GUZZLER will create appropriately named files that, again, are input arguments to program EXPLOT in the third line. The final result will be a PostScript file named myplot001.ps.
NOTE: BATCH JOB files may be edited (carefully) with any text editor!

Structure of SCRIPT files

Each script file contains records with the following information needed by DOMINO. Text in parentheses are comments.

```
script.script001.txt                                      (keyword "script." is diagnostic, don't change!)
script001.txt                                             (graphics output file name)
jun92bs.txt                                               (name of database
1.0 350 0.01 0.1E-08 0.1E-06 0.1E+01 0.1E-04 25 15 100    (calculation parameters)
0   AL(2)SI(1)O(?)   * calculates the triple point        (1 to 6 lines of chemical information)
                                                          (2-6 are often empty)
                                                          (empty)
                                                          (empty)
                                                          (empty)
                                                          (empty)
TC  300  1000                                             (definition of X-axis variable)
P  1  10000                                               (definition of Y-axis variable)
.                                                         (type of calculation)
1                                                         (choice of labels)
0.0000000E+00   0.0000000E+00                             (T and P if constant)
_script01_pix\                                            (folder name for pixel map information)
```

NOTE: SCRIPT files may be edited carefully with any text editor!

definition of X-axis

"list" Prints an alphabetically ordered list of all considered phases and their abbreviations.

X-variable: "TC" Temperature in deg. C.
"TK" Temperature in K
"P" Pressure in bar.
"A(Abc)" Activity of phase Abc. (phase name abbreviation)
"LNA(Abc)" Natural logarithm of activity for phase Abc (phase name
abbreviation).
"LOGA(Abc)" Logarithm of activity for phase Abc (phase name abbreviation).
"BIN" Binary (compositions read from input file (dat-file)).
"TER" Ternary (compositions read from input file (dat-file)).
NOTE: "Abc" is the abbreviation of a phase referred in the thermodynamic database.
(The phase is usually an endmember of a solution, but may be any non-solution
phase)
X-min X-maxDefines the value-range of the X-variable.
Grid_Points (optional) Number of grid points along the X-axis (default=10).
width (optional) Width (cm) of diagram (default=15).

IF X-variable = "TER", no X-min, X-max and definition of Y-axis is necessary.

Examples:

```
       TC  400  800
       LNA(Ab)  0.5   1.2
```

definition of Y-axis

"list" Prints an alphabetically ordered list of all considered phases and their abbreviations.

Y-variable: "TC" Temperature in deg. C.
"TK" Temperature in K
"P" Pressure in bar.
"PT" special function: Temperature in ^{o}C. P defined along "gradient":
$P = 1062.091685 + 2.567123 \cdot TC + 0.01085 \cdot TC^2$
“THG” thgr Gradient defined as P = (TC-25.0D0)*thgr+1.0D0
"A(Abc)" Activity of phase Abc. (phase name abbreviation)
"LNA(Abc)" Natural logarithm of activity for phase Abc (phase name
abbreviation).
"LOGA(Abc)" Logarithm of activity for phase Abc (phase name abbreviation).
"BIN" Binary (compositions read from input file (dat-file)).
NOTE: "Abc" is the abbreviation of a phase referred in the thermodynamic database.
(The phase is usually an endmember of a solution, but may be any non-solution
phase)
Y-min Y-maxDefines the value-range of the Y-variable.
Grid_Points (optional) Number of grid points along the Y-axis (default=10).
height (optional) Height (cm) of diagram (default=15).

Examples:

```
       P   1000   20000
       A(H2O)   0   1
```

Some checking is made, that X-axis and Y-axis is not the same variable.

calculation type

"list" Prints an alphabetically ordered list of all considered phases and their abbreviations, and a list of solutions, endmembers and site occupancies.

"." (dot) calculates an equilibrium assemblage phase diagram

"pix" ix iy Calculates equilibria on an ix by iy grid and stores all results for graymap images.
(see below for more details)

"I" Provides a simplified input for isolines calculations.

(Phase) Key (Nr) min max step Isolines calculation (see below)

isolines calculation

(Phase) Key (Nr) min max step

1. for bulk rock properties

Enter: key min max step

key is one of the following keywords:

"volsol"	Volume of solids [ccm]
"wtsol"	Weight of solids [g]
"rhosol"	Density of solids [g/ccm]
"gtot"	Total Gibbs Free Energy [J]
"%h2o.sol"	Wt% of H2O in solids

min max step minimum, maximum and step values for the isolines.
The maximum number of isolines calculated is 50.

Example: plot Wt% of H2O in solids between 0 and 10 in steps of 0.5

```
%h2o.sol    0   10   0.5
```

1. for non-solution phases

Enter: Name key min max step

Name is the name (not abbreviation) of a phase

Key is one of the following keywords:

"mol"	Amount of phase [mol]
"vol"	Volume of phase [ccm]
"mvol"	Molar volume of phase [ccm]
"wt"	Weight of phase [g]
"mwt"	Molar weight of phase [g]
"rho"	Density of phase [g/ccm]
"vol%"	Volume% of phase (% of solids)

min max step minimum, maximum and step values for the isolines.
The maximum number of isolines calculated is 50.

Example: plot iso-density lines of quartz between 2.5 and 2.7 in steps of 0.02

```
A-QUARTZ  rho   2.5   2.7   0.02
```

1. for solution phases

Enter: Name Key Nr min max step

Name is the name of a solution phase

Nr: If more than one phase of the same solution is stable, the phase with the highest concentration of the first endmember is numbered 1, the next lower is numbered 2 and so on. This is for most cases sufficient to distinguish the different phases. Nr. must be at least = 1.

Key is one of the following keywords:

endmember	abbreviation of endmember, isopleths will be calculated.
El(site)	Site occupancies, isopleths will be calculated.
"mol"	Amount of solution phase [mol]
"vol"	Volume of solution phase [ccm]
"mvol"	Molar volume of solution phase [ccm]
"wt"	Weight of solution phase [g]
"mwt"	Molar weight of phase [g]
"rho"	Density of solution phase [g/ccm]
"vol%"	Volume% of solution phase (% of solids)
"Mg#"	Ratio Mg/(Mg+Fe) of solution phase.
min max step	minimum, maximum and step values for the isolines. The maximum number of isolines calculated is 50.

Example: Plot grossular isopleths in garnet between 0.3 and 0.6 in steps of 0.02

```
GARNET   Grs   1    0.3   0.6   0.02
```

Example: plot iso-density lines of feldspar between 2.5 and 2.7 in steps of 0.02

```
FSP   rho    1  2.5    2.7    0.02
```

Note: Valid endmembers and site occupancies can be printed with "list"

pixelmap calculations

"pix" ix iy Calculates equilibria on a ix by iy grid and stores all results for graymap images. reasonable values for ix and iy are 50-200.

The information stored is the assemblage and most variables available for isolines calculation. For each variable a file is created. The information on the diagram dimension, the X- and Y-axes and the names of all variables are written to the file "pixinfo", the date and time to "datetime" These files are later used by the program MAKEMAP to reconstruct graymaps for any variable. All files are stored in a folder with a name that begins with '_'. If no such folder exists, the program will create a folder "_pixelmaps". If that name exists already, the user is asked to either enter a new name or to overwrite the folder.

labeling of reactions

(only if calculation type is not "pix")

Label Is "1", "2" or "3".

"1" For labeling reactions on each side with the stable assemblage.
E.g.: A B C = A B D

"2" For labeling reactions only by new phases produced.
E.g.: (C) = (D)

"3" For labeling reactions with a balanced equation.
E.g. A + 2 B + C = 3 D
(If more than one reaction is involved, the coefficients are omitted.
E.g.: A B C = D)

"prec" Optional, precision in cm, (default=0.005).

"smooth" Optional, smoothness in cm, (default=0.04).

The most useful option is "1", because the labels will contain the maximum information on assemblages.

During calculation, DOMINO prints comments on progress. Some are explained in commented_domino.pdf.

Interpretation of phase diagrams produced by DOMINO

Once the calculation is finished. the plt-file should be cleaned with GUZZLER (use label-size 0.2 or smaller, no special options), translated by EXPLOT and printed.
Also print the rxn-file (table) that contains a numbered list of all curves.
Because the phase diagram may consist of many curves, it is not always easy to understand. The following suggestions may help:

1. use THERIAK to get detailed information of single points in the diagram.
2. trace important curves (e.g. the stability fields of staurolite, garnet, plagioclase etc.) with colored pencils.
3. Consider calculating a simplified diagram with less components, or some phases deleted from the database.

therbin

The core of THERBIN is the equilibrium calculation program THERIAK. This approach is chosen in order to calculate very complex and non-ideal binary phase diagrams automatically, without human interference (or almost). The strategy of the program is to scan the T-x (or P-x) surface is for two-phase regions. Once one is found, it is mapped and a search for new two-phase regions is initiated adjacent to it. When all interconnected regions are found, the scanning is resumed, but only outside the already mapped areas. For THERBIN, all considered phases have to be part of a binary sub-system.

Filenames associated with Therbin

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
07	dat	THERIN	User's general input file
08	dbs	Run-time defined	Thermodynamic database
09	log	therbin.last	User responses during last run
10	out	OUT	Calculation results to print or view on screen
15	plt	plot	Graphics data; input to EXPLOT

(*) Hard-coded file names (cannot be changed)

Therbin Input files

Therbin can be invoked with a command line argument which is the file name containing the interactive input.

Theriak.ini

This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:

`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by THERBIN.

```
$THERBIN-FILES
dat=THERIN               #User's chemical input to program
log=therbin.last         #User's responses during last run of program
dbs=Run time-defined     #Thermodynamic data base to be used
plt=plot                 #Graphics data input (e.g. from DOMINO)
out=OUT                  #Calculation information to be printed or viewed on screen
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt
is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

dat-file
usually called THERIN or therin.txt. This file must be present in the working directory and contains the user input for one T and P and the chemical compositions. e.g.:

```
400     2000
1   K(0.5)NA(0.5)AL(3)SI(3)O(12)H(2) SI(10)O(20) H(20)O(10)    *
```

dbs-file
This is the thermodynamic database used for the calculations. The name is entered by the user in the interactive input. The file must be in the working directory. E.g.: `JUN92.bs`

log-file
In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with CR. (carriage return)

Terminal input
Is the interactive input through the keyboard

Therbin Output files

Terminal output
Is the text output of the program to the screen. This is the dialog, help comments, error messages and a short summary of calculation progress.

out-file
In the working directory. Usually called "OUT". Most of the above "Terminal output" is also written to this file.

plt-file
In the working directory. Usually called "plot". This file contains the graphics information computed by Therbin. A description of the format can be found in the Explot users guide, keyword "NPLOIG".
The graphics file can be translated to a PostScript file by EXPLOT. It cannot be read by GUZZLER.

Interactive input for THERBIN

This information is also available by entering "?" in the program dialogue.

database definition

"files" to see the default names, types and descriptions of files associated with the program. You may change these defaults by editing file theriak.ini with a common text editor.

database filename file name of database to be used (dbs-file).

endmembers:

"list" Prints an alphabetically ordered list of abbreviations of all considered phases.

endmember 1 Abbreviation of phase defining the composition to the left side of the diagram.

formula 1 Optional if endmember 1 is a one of the phases considered. Else the composition may be entered in the same format as the bulk is written in the dat-file.

Exemples:

```
      aQz
      phaseX    SI(1)AL(0.5)
```

endmember 2, formula2:
same as above, but defining the composition ro the right side of the diagram.

Y-variable

Y-variable may be one or the following:

"TC" Temperature in (C)
"TK" Temperature in (K)
"P" Pressure in (Bar)

ymin, ymax minimal and maximal values for the Y-axis of the binary diagram.

If the Y-variable was Temperature you have to enter the constant pressure for the diagram.
If the Y-variable was Pressure you have to enter the constant temperature for the diagram.

seeds

Usually=0.
Seeds are points in the diagram that will be searched for two-phase regions, besides the grid points. If one suspects, that a small two-phase region might be overlooked several "seeds" may be entered manually. The program will ask the coordinates for each point

scan density and tolerance

Usually suggested values are OK

X-scan-density Number of grid points in X-direction (program always suggests 10)
Y-scan-density Number of grid points in Y-direction (program always suggests 10)
tolerance Precision of calculation in [cm] (program always suggests 0.02)

Before performing any calculation, the program excludes all phases that are not within the defined binary system. (Phases must be a linear combination of the two given compositions)
The graphics input is written to the plt-file which can be converted to PostScript with EXPLOT.

therter

The core of THERTER is the equilibrium calculation program THERIAK. This approach is chosen in order to calculate very complex and non-ideal ternary phase diagrams automatically, without human interference (or almost). The strategy of the program is to scan the ternary composition surface for two-phase regions and three-phase regions. Once one is found, it is mapped and a search for new multi-phase regions is initiated adjacent to it. When all interconnected regions are found, the scanning is resumed, but only outside the already mapped areas. For THERTER, all considered phases have to be part of a ternary sub-system.

Filenames associated with Therter

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
07	dat	THERIN	User's general input file
08	dbs	Run-time defined	Thermodynamic database
09	log	therter.last	User responses during last run
10	out	OUT	Calculation results to print or view on screen
15	plt	plot	Graphics data; input to EXPLOT

(*) Hard-coded file names (cannot be changed)

Therter Input files

Therter can be invoked with a command line argument which is the file name containing the interactive input.

Theriak.ini

This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:

`$GHELP=0` (reduces the comments at the start of the programs)

and the following list with the names of input and output files used by THERTER.

```
$THERTER-FILES
dat=THERIN              #User's chemical input to program
log=therbin.last        #User's responses during last run of program
dbs=Run time-defined    #Thermodynamic data base to be used
plt=plot                #Graphics data input (e.g. from DOMINO)
out=OUT                 #Calculation information to be printed or viewed on screen
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt

is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

dat-file

usually called THERIN or therin.txt. This file must be present in the working directory and contains the user input for one T and P and the chemical compositions. E.g.:

```
400     2000
1   K(0.5)NA(0.5)AL(3)SI(3)O(12)H(2) SI(10)O(20) H(20)O(10)    *
```

dbs-file
This is the thermodynamic database used for the calculations. The name is entered by the user in the interactive input. The file must be in the working directory. E.g.: `JUN92.bs`

log-file
In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with <CR>. (carriage return)

Terminal input
Is the interactive input through the keyboard

Therter Output files

Terminal output
Is the text output of the program to the screen. This is the dialog, help comments, error messages and a short summary of calculation progress.

out-file
In the working directory. Usually called "OUT". Most of the above "Terminal output" is also written to this file.

plt-file
In the working directory. Usually called "plot". This file contains the graphics information computed by Therter. A description of the format can be found in the Explot users guide, keyword "NPLOIG".
The graphics file can be translated to a PostScript file by EXPLOT. It cannot be read by GUZZLER.

Interactive input for THERTER

This information is also available by entering "?" in the program dialogue.

database definition

"files" to see the default names, types and descriptions of files associated with the program. You may change these defaults by editing file theriak.ini with a common text editor.
database filename file name of database to be used (dbs-file).

endmembers

"list" Prints an alphabetically ordered list of abbreviations of all considered phases.
endmember 1 Abbreviation of phase defining the composition in the left lower corner of the diagram.
formula 1 Optional if endmember 1 is a one of the phases considered. Else the composition may be entered in the same format as the bulk is written in the dat-file.

Exemples:

```
      Gr
      phaseX    SI(1)AL(0.5)
```

endmember 2, formula 2
same as above, but defining the composition in the right lower corner of the diagram.

endmember 3, formula 3
same as above, but defining the composition in the top corner of the diagram.

Temperature and Pressure

T(C)	Temperature in deg. C
P(Bar)	Pressure in Bar

seeds

Usually=0.
Seeds are points in the diagram that will be searched for two-phase regions, besides the grid points. If one suspects, that a small two-phase region might be overlooked several "seeds" may be entered manually. The program will ask the coordinates for each point.

scan density and tolerance

Usually suggested values are OK

scan-density	Number of grid points for each side (program always suggests 11)
tolerance	Precision of calculation in [cm] (program always suggests 0.02)

Before performing any calculation, the program excludes all phases that are not within the defined ternary system. (Phases must be a linear combination of the three given compositions)
The graphics input is written to the plt-file which can be converted to PostScript with EXPLOT.

thalia

THALIA plots thermodynamic functions versus T or P, or ΔG (total, configurational, excess or mixing), the chemical potential, or the activity of a phase along a binary link of a solution phase. THALIA does not do any equilibrium calculations and is mostly useful to visualize the variation of thermodynamic functions and to check the database for typing errors.

Filenames associated with Thalia

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
07	dat	THERIN	User's general input file
08	dbs	Run-time defined	Thermodynamic database
09	log	thalia.last	User responses during last run
10	out	OUT	Calculation results to print or view on screen
15	plt	thplot	Graphics data; input to GUZZLER and EXPLOT
21	lpl	thaliaplot	Graphics data (long format); input to GUZZLER and EXPLOT

(*) Hard-coded file names (cannot be changed)

Thalia Input files

Thalia can be invoked with a command line argument which is the file name containing the interactive input.

Theriak.ini

This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:

`$GHELP=0` (reduces the comments at the start of the programs)

and the following list with the names of input and output files used by THALIA.

```
$THALIA-FILES
dat=THERIN               #User's chemical input to program
log=thalia.last          #User's responses during last run of program
dbs=Run time-defined     #Thermodynamic data base to be used
out=OUT                  #Calculation information to be printed or viewed on screen
plt=thplot               #Graphics data input
lpl=thaliaplot           #Graphics data input
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt

is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

dat-file

usually called THERIN or therin.txt. This file must be present in the working directory and contains the user input for one T and P and the chemical compositions. E.g.:

```
400    2000
1   K(0.5)NA(0.5)AL(3)SI(3)O(12)H(2) SI(10)O(20) H(20)O(10)    *
```

dbs-file

This is the thermodynamic database used for the calculations. The name is entered by the user in the interactive input. The file must be in the working directory. E.g.: `JUN92.bs`

log-file

In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with CR. (carriage return)

Terminal input

Is the interactive input through the keyboard

Thalia Output files

Terminal output

Is the text output of the program to the screen. This is the dialog, help comments, error messages and the calculation results.

out-file

In the working directory. Usually called "OUT". Most of the above "Terminal output" is also written to this file.

plt-file

In the working directory. Usually called "thplot". This file contains the graphics information computed by Thalia. (Its format is almost identical to the graphics output of TWQ.) A description of the format can be found in the Explot users guide, keyword "NPLOIG".

The graphics file is usually first run through GUZZLER, which formats the labels on the curves and writes a cln-file, which can be translated to a PostScript file by EXPLOT.

lpl-file

In the working directory. Usually called "thaliaplot". This file contains the graphics information computed by Thalia. The difference to the plt-file is, that all numbers are in the format: (0PE20.12) instead of F10.4.
This is because for some plots with large numbers (e.g. ΔG versus T) the number of digits in the plt-file is not sufficient and produces steps in the curve.

Interactive input for THALIA

This information is also available by entering "?" in the program dialogue.

database definition

"files" to see the default names, types and descriptions of files associated with the program. You may change these defaults by editing file theriak.ini with a common text editor.
database filename file name of database to be used (dbs-file).

X-variable definition

X-variable "TC" Temperature in deg. C
"TK" Temperature in K
"1000/T"(1000/T) Temperature in K
"P" Pressure in Bar
xmin, xmax minimal and maximal values for the X-axis.
"X" to plot G-X, μ-X or a-X diagrams (see below)

Y-variable definition (X-variable is TC, TK, 1000/T or P)

"B" back one step (in this case: stop the program)
"end" Stop the program
Y-var the Y-variable may be one of the following:

G	V	CP	H	S	LNK	LOGK	K	Basic thermodynamic functions
G-G0	V-V0	CP-CP0	H-H0	S-S0				Difference between T/P and 25 C/1Bar
G.CP		CP.CP	H.CP	S.CP				Contribution from the Cp-function
G.DIS	V.DIS	CP.DIS	H.DIS	S.DIS				Contribution from the disorder function
G.TR	V.TR	CP.TR	H.TR	S.TR				Contribution from lambda transitions
G.VOL	V.VOL							Contribution from the volume function
G.GAS	V.GAS							Contribution from the Gas function (Van der Waal, Redlich-Kwong)

Formula definition (X-variable is TC, TK, 1000/T or P)

"B" back to Y-variable definition: This allows to draw several thermodynamic functions on the same plot.
"end" Stop the program
"list" Prints an alphabetically ordered list of abbreviations of all considered phases.

formula — the abbreviation of a single phase
or a formula like: `n1 A + n2 B + C - n4 D - E`
A, B, C etc: abbreviations of phases
n1, n2, etc. coefficients, may be omitted if =1)
all items are separated by one blank.

Pressure or Temperature

"B" — Back to formula definition. This allows to draw thermodynamic functions for different phases (or formulas).
"end" — Stop the program
P — Pressure in Bar. If the X-variable was P, the program asks for T.

After this dialogue a Diagram is calculated and the question is repeated. Thus, you may enter a new pressure (or temperature), go back one step or stop the program.

Solution definition (X-variable is "X")

The program will always print a list of all solution phases.
"B" — back one step (in this case: stop the program)
"end" — Stop the program
solution name — Name of the solution.

Composition definition (X-variable is "X")

Define "binary endmembers" as linear combination of the endmembers of the solution

"B" — Back to solution definition. This allows to draw different solutions on the same plot.
"end" — Stop the program
binary endmember1 — Definition of the left side of the diagram in terms of endmember concentrations.
As an example with FSP as solution phase, Ab would be (1 0 0), Kfs is (0 1 0) and An is (0 0 1). One could also define a mixture e.g. (1 1 0), which would lie in the middle between Ab and Kfs. (Note: the concentrations are normalized to a sum of one by the program)

binary endmember2
same as above for the second composition, defining the right side of the diagram.

Y-variable definition (X-variable is "X")

"B" — Back to composition definition. This allows to draw different binary links on the same plot.
"end" — Stop the program
Y-variable — The Y-variable may be:
"G" Total Gibbs Free Energy ΔG
"GCON" Configurational Contribution to ΔG
"GEX" Excess contribution to ΔG
"MUE" n Chemical potential of n'th endmember
"ACT" n Activity of n'th endmember

Temperature and Pressure

"B" Back to composition definition. This allows to draw different thermodynamic functions on the same plot.
"end" Stop the program
temperature pressure Temperature in deg. C and pressure in Bar.

After this dialogue a Diagram is calculated and the question is repeated. Thus you may enter a new pressure (or temperature), go back one step or stop the program.
The graphics input is written to the plt- and lpl-file which can be cleaned with GUZZLER and converted to PostScript with EXPLOT.

thermo

For a given p and T, THERMO calculates ΔG (apparent Gibbs free energy, = ΔH - T.S), ΔH, S, Cp and V. Also, for each thermodynamic function (Cp, volume, disorder, lambda transition) its contribution to the total ΔG, ΔH. The main application of THERMO is to check the calculations for programming errors and to check the database for typing errors. Further it prints tables of ΔG and Volume for comparing different databases or equations of state.

Filenames associated with Thermo

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
07	dat	THERIN	User's general input file
08	dbs	Run-time defined	Thermodynamic database
09	log	thermo.last	User responses during last run
10	out	OUT	Calculation results to print or view on screen
13	tab	stat.table	Table of variable values; input to spreadsheet programs

(*) Hard-coded file names (cannot be changed)

Thermo Input files

Thermo can be invoked with a command line argument which is the file name containing the interactive input.

Theriak.ini
This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:
`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by THERMO.

```
$THERMO-FILES
dat=THERIN                #User's chemical input to program
log=thermo.last           #User's responses during last run of program
dbs=Run time-defined      #Thermodynamic data base to be used
out=OUT                   #Calculation information to be printed or viewed on screen
tab=stat.table            #Table of variable values; input to a spreadsheet program
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt
is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

dat-file
usually called THERIN or therin.txt. This file must be present in the working directory and contains the user input for one T and P and the chemical compositions (not used by Thermo). E.g.:

```
400     2000
1   K(0.5)NA(0.5)AL(3)SI(3)O(12)H(2) SI(10)O(20) H(20)O(10)    *
```

dbs-file
This is the thermodynamic database used for the calculations. The name is entered by the user in the interactive input. The file must be in the working directory. E.g.: `JUN92.bs`

log-file
In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with CR. (carriage return)

Terminal input
Is the interactive input through the keyboard

Thermo Output files

Terminal output
Is the text output of the program to the screen. This is the dialog, help comments, error messages and the calculation results.

out-file
In the working directory. Usually called "OUT". Most of the above "Terminal output" is also written to this file.

Interactive input for THERMO

database definition

"files"	to see the default names, types and descriptions of files associated with the program. You may change these defaults by editing file theriak.ini with a common text editor.
database filename	file name of database to be used (dbs-file).

phase definition

"list"	Prints an alphabetically ordered list of abbreviations of all considered phases. Independently on the chemical composition given in the dat-file, these are all phases in the database.
"end"	Stop the program.
"TP" t p	Set the Temperature (C) and Pressure (Bar) (Initially set to the values from the dat-file)
"g"	Calculate a table of G-values (for previously defined phases)
"v"	Calculate a table of Vol-values (for previously defined phases)

phases	may be a single phase (one abbreviation) or a list of phases, each item separated by two blanks or the keyword "all"

GUZZLER

GUZZLER is a program that distributes the labels on the curves, avoiding overlaps. The input file to GUZZLER is a graphics file (plt-file) produced by DOMINO, or THALIA. Because TWQ uses the same file format, GUZZLER may also be used to clean up the plot.dat files from TWQ.
The plt-files from DOMINO have initially a label-size of zero. Translating them with EXPLOT to PostScript and printing will show no labels. GUZZLER assigns a size and distributes the labels.
Four cases are distinguished:

1) Full label is written
2) Only the number of the label is written on the curve.
3) The number is written somewhere close, with a fine line pointing to the reaction.
4) The curve is not labeled at all

Like all software that appears to do some "thinking", GUZZLER is quite straight forward and does not do any thinking at all.

Filenames associated with Guzzler

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
12	log	guzzler.last	User responses during last run
13	cln	clean	Labeling reactions in graphics files; input to EXPLOT
14	plt	Run-time defined	Graphics data; input to GUZZLER
15	rxn	table	List of reactions and assemblages
16	grd	gitter	File used for bookkeeping.

(*) Hard-coded file names (cannot be changed)

Guzzler Input files

Guzzler can be invoked with three command line arguments which are the file name of the graphics file (e.g. plot), the name of the cleaned file (e.g. clean) and the name of the reaction list (e.g. table)

.

Theriak.ini
Only used in the "kguzzler" version. See comments inTheriakDominoGuide.doc.
This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:
`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by GUZZLER.

```
$GUZZLER-FILES
log=guzzler.last        #User's responses during last run of program
cln=clean               #Graphics data after label arranging and cleaning (e.g. input to EXPLOT)
plt=Run time-defined    #Graphics data input (e.g. from DOMINO)
rxn=table               #Reaction/assemblage list
grd=gitter              #Grid information for graphics file
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt
is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

plt-file
In the working directory. This file contains the graphics information computed e.g. by Domino. (Its format is almost identical to the graphics output of TWQ.) A description of the format can be found in the Explot users guide, keyword "NPLOIG".

log-file
In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with CR. (carriage return)

Terminal input
Is the interactive input through the keyboard

Guzzler output files

Terminal output
Is the text output of the program to the screen. This is the dialog, help comments and error messages

cln-file
In the working directory. Usually called "clean". This file contains the graphics information of the "cleaned-up" plot. This may be translated to PostScript by EXPLOT.

rxn-file
In the working directory. Usually called "table". This is a numbered list of all curves defined in the graphics file.

grd-file
In the working directory. Usually called "gitter". This file is used by Guzzler to keep track of space available for labeling. For the user this file is of no interest.

Interactive input for GUZZLER

This information is also available by entering "?" in the program dialogue.

graphics file

```
Enter [ "?" | CR | graphics file name ] <coplot>?
```

graphics file name   name of a graphics file to be used as input.

Size of labels

size of labels   label-size in cm. with no log-file the program will suggest 0.2

After this input, a numbered list of all curves is printed on the screen.

Labeling

option	choose one of the options (see below) (default=-3)
min_length	define the minimum length of a curve for labeling (default=0.02 cm)

options

-3 (default) Write full labels, if possible, else try numbers on lines or offset numbers.
3 Force full text-labels on every curve. This may be useful, if few lines are present and the overlapping labels are still readable.
-2 Use only numbers (no text-labels). The numbers may be either on the curve or offset.
2 Force the reaction numbers to be written on each curve.
-1 (use offset numbers only, option not recommended)
1 (force offset numbers, option not recommended)
0 Do not label curves. This option removes all labels and produces a clean diagram which can be labeled and colored manually.
4 Label all curves with either full text-labels or with numbers on line
5 No offset numbers. This option produces a nice readable diagram. Short lines will not be labeled.

run GUZZLER with no interactive input

This option is normally used with DOMINO scripts, doing multiple phase diagram calculations. In this case GUZZLER is run with command line arguments:

guzzler plt-file [cln-file rxn-file]

where plt-file Name of graphics input file
cln-file Name of "cleaned" output file (if not given, default name is taken)
rxn-fileName of file with table of reactions (if not given, default name is taken)

GUZZLER will use the default labeling method and 0.2 for label-size or the value given in the log-file.

EXPLOT

Explot is a simple general-purpose plot program. The input is a command language, similar to the plot commands used by many computer languages. The output is a PostScript file that may be sent to a printer or viewed at a console. A detailed description of the input options is given in ExplotGuide.doc.
One of the options for input is the format used by DOMINO and other software for the graphics output file.

Filenames associated with Explot

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
12	log	explot.last	User responses during last run
13	cln	clean	Labeling reactions in graphics files; input to EXPLOT

14	plt	Run-time defined	Graphics data; input to GUZZLER
15	rxn	table	List of reactions and assemblages
16	grd	gitter	File used for bookkeeping.
17	pst	plot.ps	Graphics in PostScript format

(*) Hard-coded file names (cannot be changed)

Explot Input files

Theriak.ini

Only used in the "kexplot" version. See comments inTheriakDominoGuide.doc.
This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:
`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by EXPLOT.

```
$EXPLOT-FILES
log=explot.last          #User's responses during last run of program
cln=Run time-defined     #Graphics data (input)
pst=plot.ps              #Graphics file in PostScript format
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt

is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

cln-file

In the working directory. This file contains the graphics information computed e.g. by DOMINO or GUZZLER or a general EXPLOT input file. A description of the format can be found in the Explot users guide, keyword "NPLOIG".

Explot output files

Explot can be invoked with two command line argument which are the file name of the graphics file (e.g. clean) and the name of the plot file (e.g. plot.ps). (Note that a second plot file (plot.svg) will also be produced.)

Terminal output

Is the text output of the program to the screen. This is the dialog, help comments and error messages

pst-file

In the working directory. Usually called "plot.ps". This is a PostScript file and may be viewed (and even modified) by several programs, e.g. Adobe Illustrator, Corel Draw, PreView or ghostview (UNIX).

Interactive input for EXPLOT

This information is also available by entering "?" in the program dialogue.

graphics file name   Name of input file.

run EXPLOT with no interactive input

This option is normally used with DOMINO scripts, doing multiple phase diagram calculations. In this case EXPLOT is run with command line arguments:

explot cln-file [pst-file]

where cln-file Name of EXPLOT input file
pst-file Name of PostScript file produced by EXPLOT. (Usually with the extension ".ps")
(if not given, default name is taken)

makemap

MAKEMAP makes graymap images from the information stored by DOMINO in the pixelmap folder. (Calculation type "pix"). MAKEMAP searches in the working directory for folder names beginning with "_". These are assumed to contain the files "pixinfo", "datetime" and one file for each variable. The files produced are a pst-file (image in PostScript format) and a pgm-file (image as 'portable gray map').

Filenames associated with Makemap

I/O unit	type	file name	Description
01	ini	theriak.ini (*)	Program initialization data file
02	hlp	thhelp.txt (*)	Help file
05	kbd	Keyboard (*)	Keyboard (standard input)
06	scr	Screen (*)	Screen (standard output)
09	log	makemap.last	User responses during last run
10	out	OUT	Error messages to print or view on screen
19	pst	pixel.ps	Image file in PostScript format
24	pgm	pixels.pgm	Pixelmap file in pgm format (Portable Gray Map)
30		pixinfo (*)	Diagram information for graymaps
32		datetime (*)	Date and time of diagram calculation
33		names read from pixinfo	graymap info for each variable

(*) Hard-coded file names (cannot be changed)

Makemap Input files

Makemap can be invoked with a command line argument which is the file name containing the interactive input.

Theriak.ini

This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:
`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by MAKEMAP.

```
$MAKEMAP-FILES
log=makemap.last         #User's responses during last run of program
pgm=pixels.pgm           # ascii-image of pixel map
pst=pixel.ps             # Pixel map (PostScript file) (old unit 20)
$END
```

use the keyword "files" in the interactive input to print all file names defined.

thhelp.txt
is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

log-file
In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the database) can be used again with CR. (carriage return)

Terminal input
Is the interactive input through the keyboard

pixelmap folder
Folder containing the information for graymaps. Folder name must begin with "_".

file "pixinfo"
Inside pixelmap folder. Contains;

- label for X-axis
- label for Y-axis
- lower and upper limits for X- and Y-axis
- height and width of diagram in cm
- number of grid points in X and Y direction and a code for xy or triangle plots.
- number of comment lines, followed by this amount of comments.
- number of composition lines followed by this amount of compositions.
- one line for each variable name.

file "datetime"
Inside pixelmap folder. Contains one line with date, time and calculation time.

files with graymap information
Inside pixelmap folder. The names of these files were created by domino. A list is included in "pixinfo".

Makemap output files

Terminal output
Is the text output of the program to the screen. This is the dialog, help comments and error messages

pgm-file
graymap image in "portable graymap' format. This may be used as raw data for some programs for coloring and enhancing. File contains only the grayscale information.

pst-file
PostSript image. Contains the axes, comments and a scale of gray values.

Interactive input for MAKEMAP

This information is also available by entering "?" in the program dialogue.

pixelmap folder

The program prints a numbered list of folders containing pixel map information.

"files"	to see the default names, types and descriptions of files associated with the program. You may change these defaults by editing file theriak.ini with a common text editor.
folderNr	Choose from the above list which folder to use. (enter the folder number)

function number

The program prints a numbered list of functions available for mapping.

"list"	to print the numbered list of all available functions for mapping.
functionNr	the number corresponding to the desired function.
key	optional key for special mapping

"x1"	map the first derivative of the chosen function f with respect to the X-variable of the diagram (df/dX)
y1"	map the first derivative of the chosen function f with respect to the Y-variable of the diagram (df/dY).
"alpha"	map (1/f)*(df/dX)
"beta"	map -(1/f)*(df/dY)

The keys "alpha" and "beta" are primarily defined having in mind the volume function (corresponding to thermal expansion coefficient and isothermal compressibility coefficient, respectively). However, they may be used for any other function too.

plotxy

This program makes simple X-Y-plots from the tables (tab- and tcp-files) produced by theriak. One axis is always restricted to one variable, the other may have up to 20 variables. Plotting the sum of variables is also possible. Note: the tables can also be analyzed with some statistical software or e.g. Excel.

Filenames associated with plotxy

I/O unit	type	file name	Description
01	*ini	theriak.ini	Program initialization data file
02	*hlp	thhelp.txt	Help file
05	*kbd	Keyboard	Keyboard (standard input)
06	*scr	Screen	Screen (standard output)
08	dbs	Run-time defined	Table of thermodynamic variables
09	log	plotxy.last	User responses during last run
10	out	OUT	Calculation results to print or view on screen
15	plt	xyplot	Graphics data; input to GUZZLER or EXPLOT

*: Hard-coded file names (cannot be changed)!

plotxy Input files

Plotxy can be invoked with a command line argument which is the file name containing the interactive input.

theriak.ini

This file is read at the beginning and contains information on file names and calculation parameters. If this file is present in the working directory, this will be used, else the file in the program directory

is taken. To make changes, copy the file to the working directory and edit its content. The most important items one might change are:
`$GHELP=0` (reduces the comments at the start of the programs)
and the following list with the names of input and output files used by PLOTXY.

```
$PLOTXY-FILES
log=plotxy.last          #User's responses during last run of
program
out=OUT                  #Calculation information to be printed or
viewed on screen
dbs=Run time-defined     #table to be used
plt=xyplot               #Graphics data (e.g. input to GUZZLER)
$END
```

thhelp.txt
is located in the program directory and contains the text printed if the user enters a "?" in the interactive input.

dbs-file
This is the table of thermodynamic variables. The name is entered by the user in the interactive input. The file must be in the working directory. E.g.: `loop.table`

log-file
In the working directory. Contains the interactive input from the last run. All input that remains the same (e.g. the name of the table) can be used again with <CR>. (carriage return)

Terminal input
Is the interactive input through the keyboard

plotxy Output files

Terminal output
Is the text output of the program to the screen. This is the dialog, help comments, error messages and a short summary of calculation progress.

out-file
In the working directory. Usually called "OUT". Most of the above "Terminal output" is also written to this file.

plt-file
In the working directory. Usually called "xyplot". This file contains the graphics information computed by plotxy. A description of the format can be found in the Explot users guide, keyword "NPLOIG".
The graphics file is usually first run through GUZZLER, which formats the labels on the curves and writes a cln-file, which can be translated to a PostScript file by EXPLOT.

Interactive input for PLOTXY

This information is also available by entering "?" in the program dialogue.

table filename

"files"   to see the files associated with the program.

filename name of a file containing a table of thermodynamic variables corresponding to an theriak output tab-file.

X-axis and Y-axis:

number(s) list of variable numbers or variable expressions separated by commas. A variable expression has the format: var1+var2+var3 ...
Each variable may be the sum of several variables.
Note: One of the axes must have only one variable number or expression.
EXAMPLE:
For X: 1
For Y: 3, 5, 6+7, 8+9

X-min X-max and Y-min Y-max

The minimum and maximum values for the axis are printed. Enter different values if desired.

At the end the program prints the labels of the X- and Y-axis, and all for all plotted lines.

Description of input files

The Initialization file `theriak.ini`

The name of this file is hard-coded and should not be changed. It contains all appropriate information need by the programs on start. The following information describes the format of `theriak.ini`; it is included also in this file.
Significant records in `theriak.ini` start with $ at position1 have the following forms:

```
$Variable=value
```

or

```
$Block-Identifier
key1=val1           #Description1
key2=val2           #Description2
.....................................
$END
```

The former defines the `numerical value` of a program `control variable (e.g. $Clear=0)`; the latter defines a `data block`. Note that in either case no space is allowed before and after the equal sign.
A data block starts with a `block-identifier` that is hard-coded and should not be changed, e.g. `$THERIAK-FILES`. Each record within a block starts at position 1 with a 3-character long `keyword` that is hard-coded and should not be changed. This keyword is followed by = and then by a `value`, e.g. `dat=THERIN`. A # character signifies an `optional comment` within the record. Within a block, those records that are empty or start at position 1 with ! are ignored. The block ends with `$END`.

Control variables

The following control variables are currently implemented.

```
$Clear=1
```

It defines if the screen will be cleared (value=1) or not (value <>1) before printing help information.

```
        $GHELP=1
```

It controls if the first help after program start will be printed or not. Valid values are 1 (print) or 0 (don't print)

```
        $BATCH=1
```

It controls if DOMINO plot files created with the help of `script files` are immediately processed with the programs GUZZLER and EXPLOT. Valid values are 0 (do not proceed) or 1 (proceed).

Data blocks

The following data blocks are currently implemented.

Calculation parameters

```
        $CALC-PARAMS
        TEST=1E-9    LO1MAX=300     GCMAX=500    DELXSTAR=1E-4  STEPSTAR=25
        EQUALX=1E-2  DELXSCAN=1.0   STEPMAX=25   DELXMIN=1E-9
        $END
```

This block contains one or more statements in the form: **parameter=value** separated by at least two blanks. For parameters that are not assigned a value in the ini-file, the default value is used. In general, these values may be changed, but if this is not done carefully, the results may be unpredictable. It is suggested not to change these parameters.

Parameter	Default	Description
LO1MAX	80	Maximum number of iterations for the calculation. If LO1MAX < 0 then the number of iterations will be exactly |LO1MAX|.
TEST	10E-8	The calculation will stop, if the absolute sum of all non-positive Gi after the linear programming step is smaller than |TEST|.If TEST < 0 then the program prints a short summary for each iteration. (For testing the program).
EQUALX	10E-5	Two phases of the same solution are considered equal if |Xi| < EQUALX.
DELXMIN	10E-7	Smallest possible stepsize = precision for calculating Xi's in a non-ideal solution.
DELXSCAN	1	Grid width for scanning non-ideal solution phases for initial guess. DELXSCAN cannot be smaller than 0.001 or bigger than 1.
DELXSTAR	0.02	Initial stepsize.
STEPSTAR	2	Maximum number of steps, if initial guess results from a scan.
STEPMAX	3	Maximum number of steps if initial guess is a previous minimum.
GCMAX	100	Maximum number of G-function calls per step (= maximum m in stepsize procedure).

File data blocks

In these blocks the user may define the `default filenames` for each program. Note that there are no default filenames, so the user must define them. There are as many blocks as programs. Each program is associated with a number of files that have certain contents. For example, the program THERIAK is associated with at least the following file types:

Type	Description
dbs	Thermodynamic data
dat	User data, chemical composition of system
out	Calculation results
log	User input from the previous run

Depending on the nature of contents, these files are assigned a **`filetype`** that is a hard-coded, 3 characters long **`keyword`**, as shown above. The default filenames defined by the user for each program are actually associated with the filetypes needed by this program. For all programs, some filenames are hard-coded and cannot be changed by the user. These include:

Type	Filename	Status	Description

ini	theriak.ini	Should always exist	Program initialization data file
hlp	thhelp.txt	Should always exist	Help file
kbd	keyboard	System	Interactive input from user (standard input)
scr	screen	System	Output to the screen (standard output)

For the Windows operating system, file `domjob` will automatically become `domjob.bat`. The names of all other files for each program may be defined by the user in **program-files blocks** (see below). The file types implemented in the current versions are the following:

Type	**Description**	**Type**	**Description**
ini	Program initialization data file	tab	Table of variable values; input to a spreadsheet program
hlp	Help file	tcp	Table of variable values; input to "TechPlot"
spt	DOMINO script file (default: scriptXXX[.txt])	plt	Graphics data (input to GUZZLER and EXPLOT)
job	DOMINO batch job file (default: domjob[.bat])	cln	Graphics data after arranging labels and cleaning (input to EXPLOT)
kbd	Keyboard, standard input	rxn	Reaction/assemblage list
scr	Screen, standard output	grd	Grid information for GUZZLER
dat	User data, chemical composition of system	pst	Graphics file in PostScript format
dbs	Thermodynamic data base	fig	Directory name for pixelmaps
log	User's responses during last run of program	lpl	Graphics data; long format.
out	Calculation results to print or view on screen	lbk	(reserved for future use)
bin	Graphic data for "pseudo-binary system" calculations	ibk	(reserved for future use)
drv	Special calculation directives to run THERIAK	pgm	Pixelmap file in pgm format (Portable Gray Map)

File name conventions and assignments

Note: Unix and derivate operating systems need case sensitive file names!
A filename can be as long as 30 characters. In case the default filenames contain period(s), the program will consider the character string immediately following the left-most period as a file extension, i.e. as `filename.extension`. In this case `filename` should not exceed 20 characters and `extension` (including the period) 10 characters. If the default filenames do not contain period(s) only the first 20 chars will be assigned to `filename`, whereas the `extension` will be empty. There are two methods to assign the default filenames for each program.
Method 1: Fixed default filenames. This method becomes obvious by the example of the `$THERIAK-FILES` block.

```
$THERIAK-FILES
dat=THERIN                    #User's chemical input to program
dbs=Run-time-defined          #Thermodynamic data base to be used
log=theriak.last              #User's responses during last run of program
out=OUT                       #Calculation information to be printed or viewed on screen
bin=binout                    #Graphics data for "pseudo-binary system" calculations
drv=Run-time-defined          #"Special calculation" directives for running the program
tab=loop_table                #Table of variable values; input to a spreadsheet program
tcp=loop_tecplt               #Table of variable values; input to "TechPlot"
inf=loop_info                 # Table of variable values; additional information
$END
```

For the Windows operating system all files are usually defined with the extension .txt, so they can be double-clicked on any File Manager window and get opened with the default text editor NOTEPAD. Note that all files of the programs are ordinary text files that can be viewed and modified by any text editor of your choice. For Unix and derivate operating systems, the file/program-to-open associations follow different rules that are not only related to file extension. Note that the number of files for each program is fixed by the number of file types. Note also that more that one programs produce output information (file type out). For some users It may be convenient to distinguish among the various out-files, e.g. ThkOut.txt could be a THERIAK out-file whereas ThbOut.ixt is a THERBIN out-file.

Method 2: Run-time-defined file names. This method allows for file naming at run time. It is active in case you set the filename of type `dat` to a single question mark. For the example above, this looks like this:

```
        $THERIAK-FILES
        dat=?                          #User's chemical input to program
        ................................................
        $END
```

In this case, the information following the `dat`-record is ignored and the program will state on start that

```
        Filename for file type dat not defined in theriak.ini
        Enter file name:
```

Now you enter the name of file type dat that contains the chemical input for your calculation, e.g.

mydata

Then, the files that will be created by the program will be named automatically to

```
        mydata-plt.txt
        mydata-cln.txt
        mydata-tab.txt
```

etc.

In this example, the contents of the associated files become evident by appending the file type to the input string "mydata". How each program names its files can be checked at program's first prompt by entering the keyword `"files"` (see on-line help).

User Input (`dat`-file)

The `dat`-file contains information on pressure, temperature as well as bulk chemical composition of the system. Although this information is not always used, this file must be present in the working directory, because it is read by all thermodynamic programs.

File format

Comments in this file start with ! at position 1. The structure of input data includes two parts of information:

1. External conditions for calculation (e.g. T and P) and
2. Bulk system composition given as element amounts

The **first significant line** should contain the external conditions as follows in the following form:

```
        temperature(deg C)    pressure(bar)    optional-[PGAS]-ratio
```

where

`[PGAS]-ratio` = (fluid pressure)/(total pressure)

This line should be followed by **at least** one bulk composition line have the following form:

```
        print-code    bulk-formula    use-code    optional-comment
```

The arguments should be **separated by at least two blanks (no tabs!)**. Their significance and format are as follows:

`print-code` should start at position 1 and may take the following values:

< -1: Print information about selected or rejected phases; no equilibrium calculation.

-1: Print composition, considered phases and solution models; no equilibrium calculation.

0: Short output of stable assemblage calculation.

1: Long output of stable assemblage calculation including information about composition, considered phases, solution models, and activities of all phases.

`bulk-formula` is the element bulk composition written as follows.

`COMP1(A1)COMP2(A2).....`

where **`COMP1, COMP2`** ... are the components that are defined in the database file (see dbs-file, Components) and **`A1, A2`**, ... their amounts enclosed in parentheses. Usually, the components are the **elements**. For the **total oxygen** of the bulk-formula, the input **`O(?)`** instructs the program to assign the exact stoichiometric oxygen amount to the other input elements. The amounts may be any numbers, but should be positive to avoid unpredictable results. The maximum length of **`bulk-formula`** is 170 characters; it may contain single blanks and must be separated from **`print-code`** and **`use-code`** by at least two blanks.

`use-code` controls if some or all phases included in the database and matching the given bulk composition should be taken in consideration. In the normal case, an asterisk **`*`** includes all matching phases. How to work with the **`use-code`** is further explained in paragraph 0.

`optional-comment` may describe shortly the bulk composition. This information will be printed into the final diagram.

For the most typical case, only one chemical input line is needed. Any subsequent lines will be ignored. If a pseudo-binary (or pseudo-ternary) diagram is to be calculated (in THERIAK: keyword `"bin";` in DOMINO: keywords **`"BIN"`** or **`"TER"`**) two (or three) **additional** composition lines are needed that define the endmember compositions of the binary (or ternary) sub-system. Accordingly, if both axes in a DOMINO calculation should be pseudo-binary systems, two pairs of composition lines (a total of four) are required. Note that the second and further composition lines should not contain elements that are not already included in the first one.

Examples of dat-files

Example 1

```
700    3000
0    NA(0.5)K(0.5)AL(1.0)SI(3.0)O(8.0) SI(10)O(20) H(20)O(10)
*
```

This is a typical input for a simple equilibrium calculation. The temperature is 700 C, the Pressure 3000 bar. The bulk composition corresponds to 0.5 K-Feldspar + 0.5 Albite, with "excess" (10 mole) SiO_2 and H_2O. The Zero at the beginning of the second line produces a short output (only the basic results) and the star "*" at the end means, that all matching phases in the database are to be considered.

Exemple 2

```
700       3000
0   NA(0.5)K(0.5)AL(1.0)SI(3.0)O(8.0) SI(10)O(20) H(20)O(10)
*
0   K(1)AL(1)SI(3)O(8) SI(10)O(20) H(20)O(10)    *
0   NA(1)AL(1)SI(3)O(8) SI(10)O(20) H(20)O(10)    *
```

DOMINO may use this input to calculate, for example, a binary T-X phase diagram along the K-Feldspar - Albite joint. Note that the first composition line defines the whole system completely whereas the following lines define the endmembers of the binary.

Exemple 3

```
0   A(1)B(1)X(1)Y(1)      *
0   A(1)     *
0   B(1)     *
0   X(1)     *
0   Y(1)     *
```

This example demonstrates how DOMINO can handle a hypothetical reciprocal solution system by defining both diagram axes (x and y) as binary sub-systems by means of the keyword "BIN" entered at run-time. Provided that the database contains thermodynamic properties for AX, AY, BX, BY as well as an appropriate solution model, isopleths for the total Gibbs free energy may be calculated and plotted.

Directives for theriak (drv-file)

The drv-file drives equilibria calculations, e. g., along a given PT-path. The drv-file is a text file created with any editor and saved as text only if necessary. It contains commands for theriak which are executed in order.

File format

Each line contains a command and some parameters.
Initially the temperature, pressure and composition are read from the dat-file (usually THERIN). These values can be re-defined in the drv-file.

Command	Parameters	Description
REF	t1 t2	defines the equilibrium assemblage at the reference t1, p2. This is the "reference assemblage" adding the variable "G_overstep" to the tab- and tcp-files.

COMP	formula use	Defines new bulk composition for further calculations. Initially the composition is read from the dat-file. It can be changed several times during one run.
		Example:
		`COMP`
		`SI(3)AL(1)MG(3)K(1)O(12)H(2)    *`

ADD	formula	Adds the given formula to the current bulk composition. The amounts may be negative, but only so much will be subtracted that all components remain >= 0 Example: `ADD   H(20)O(10)`
REMOVE	phase p	After each of the following calculations, the composition of stable phase will be (partly) removed from the bulk. p is the percentage of phase that is to be removed. Example: `REMOVE  GARNET  100`
TP	t2 p2 s	Calculates equilibria along a linear PT-path from t1 and p1 to t2 and p2 The current values t1, p1 are input 1. either from the dat-file 2. or from a preceding "TP" or "REF" directive "s" is the number of calculation steps (default= 1) along the path. All variable values for each step are written into the tab- and tcp-file. Example: `TP   800   5000   10`

The database (dbs-file)

General structure

The format of the database follows closely the one used by E.H. Perkins, R.G. Berman and T.H. Brown for the calculation of phase diagrams. The first few lines of the database define the components to be used. The rest of the database is organized in sections. Each section begins with a line having the string *** as the first three non-blank characters. The rest of that line is scanned for a string identifying the section. The sections recognized by THERIAK are:

```
      *** ... MINERAL DATA ...
      *** ... GAS DATA ...
```

These two sections are treated identically. They contain the phase definitions and the information to calculate G for variable pressures and temperatures.

```
      *** ... SOLUTION DATA ...
```

Definitions of solutions and solution models.

```
      *** ... MARGULES ...
```

This section contains Margules parameters for non-ideal solution phases.

```
      *** ... FIXED PHASES ...
```

Section containing solution phases with fixed compositions

Other sections (e.g. *** ... COMMENTS ...) will be skipped by the program. Principally the sections can be in any order and may appear more than once. It is obvious that if we define a solution phase, its endmembers must have been read-in previously. Similarly the Margules parameters for a solution can only be assigned properly after the solution phase is defined.

Any line that is empty or begins with a '!' is considered a comment line and is skipped.

Components information

The database always begins with the definition of all components (usually elements), their atomic weight and the number of oxygens associated with them:

```
      NC       [R]
      COMPN(1) COMPN(2)  ...  ...  CMPN(7)   ...
      COMPN(X) COMPN(Y)  ...  ...  CMPN(NC)
      MOLWT(1) MOLWT(2)  ...  ...  MOLWT(7)  ...
      MOLWT(X) MOLWT(Y)  ...  ...  MOLWT(NC)
      OXYNR(1) OXYNR (2)  ...  ... OXYNR(7)  ...
      OXYNR(X) OXYNR(Y)  ...  ...  OXYNR(NC)
```

where

NC: Number of components used in the database

R: Optional input of gas constant [J/mol] (default: 8.3143).

COMPN(i): Components (Format = 7A10)

MOLWT(i): Atomic weights. (Format = 7F10.2)

OXYNR(i): Standard number of oxygens for the corresponding component. (Format = 7F10.2)

Example

```
25
    O         AL        BA        C         CA        CL        CO
    CU        F         FE        H         K         MN        MG
    NA        NI        P         S         SI        SR        TI
    ZN        ZR        B         E
  15.99940  26.98154 137.32700  12.01100  40.07800  35.45270  58.93320
  63.54600  18.99840  55.84700   1.00794  39.09830  54.93085  24.30500
  22.98977  58.69000  30.97362  32.06600  28.08550  87.62000  47.88000
  65.39000  91.22400  10.81000   1.00000
   0.0       1.5       1.0       2.0       1.0       0.0       1.0
   1.0       0.0       1.0       0.5       0.5       1.0       1.0
   0.5       1.0       1.0       0.0       2.0       1.0       2.0
   1.0       2.0       0.0       0.0
```

Section *** ... MINERAL DATA ...

Each phase is defined with one phase definition line and a variable number of data lines. The phase definition line contains the name and the chemical composition; the data lines contain the parameters used to calculate the thermodynamic properties of the phase. Different equations of state are implemented in the code. Unusual or very complex equations (e.g. the properties of H2O according to Haar et al., 1983) are programmed as external subroutines. (See paragraph 0)

Phase definition line

```
NAME     FORMULA     ABBREV     [CODE]
```

The four strings are separated by at least two blanks and may contain single blanks.

NAME is the name of the phase (maximum 16 characters).

Special case: If the name begins with a '$' (dollar sign), the phase will not be considered for an equilibrium. However, if a table or pixelmap is produced its chemical potential is recorded as G_name and Gneg_name.

FORMULA is the chemical formula of the phase in the form described earlier, i.e.

COMP1(A1)COMP2(A2).....

where **COMP1, COMP2 ...** are the components that are defined in the database file and A1, A2, ... their amounts enclosed in parentheses. Note that here **A1, A2, ...** may be negative.

ABBREV is a suitable abbreviation for the phase name (maximum 8 characters)

CODE is optional and may be composed of **less than 16 characters**, each of them defining a **group** that the phase belongs to. Phases belonging to the same group, i.e. having the same character in their code may be addressed by the use-code discussed earlier. Examples of using code and use-code are discussed in the chapter " USE and CODE (selecting phases)". If the first character in code is an asterisk * then the phase is considered as "**special**", e.g. a buffer.

A line within this section is recognized to be a phase definition line if it contains at least one '(' (left parenthesis) and is not a comment line.

Example

```
HIGH ALBITE   NA(1)AL(1)SI(3)O(8)   halb   n
```

Data lines

The general equation to calculate the ***apparent Gibbs Free Energy*** of a phase is:

$$\Delta_a G^{T\,P} = \Delta_f H^{T_0 P_0} + \int_{T_o}^{T} Cp\,dT - T\cdot S^{T_0 P_0} - T\cdot \int_{T_o}^{T} \frac{Cp}{T}\,dT + \int_{P_o}^{P} V\,dP$$

where

T = the temperature in Kelvin

P = the pressure in bar

T_o and P_o are 25 °C and 1 bar, respectively.

$\Delta_a G^{T,P}$ [J/mol] = the molar "***apparent" Gibbs Free Energy***

$\Delta_f H^{To,Po}$ [J/mol] = the Enthalpy of formation from the elements at T_0 and P_0.

$S^{To,Po}$ [J/(K mol)] = the molar entropy at T_0 and P_0..

C_p [J/(K mol)] = the heat capacity function $C_p(T)$ at P_0.

V [J/(Bar mol)] is the molar volume function $V(T,P)$

The thermodynamic data lines for each phase follow the phase definition line in **any order**. They have the general format

```
Cod         |Number-field01|Number-field02|Number-field03|Number-field04|Number-field05|
```

corresponding to the Fortran format (A5,5F15.4). The three consecutive non-blank characters **cod** identify the content of the data line. They must lie **within the first five** characters of the data line. The rest of the data line (character 6 to 80) may contain at most five 15-characters-wide numerical fields. The currently implemented data lines are discussed below.

Standard state (25°C, 1 bar) information, **cod = ST**

```
ST      G0      H0      S0      V0
```

G0 = $\Delta_a G^o$ [J/mol] is the molar "apparent" Gibbs Free Energy. It is commonly not used and this field may me empty.
H0 = $\Delta_f H^o$ [J/mol] is the Enthalpy of formation from the elements.
S0 = S^o [J/(K mol)] is the molar entropy.
V0 = V^o [J/(Bar mol)] is the molar volume.

Example (Albite)

```
ST      -3701276.00    -3921618.20       224.4120          10.083         1.00000
```

Heat capacity information, cod = Cn

```
C1    k1    k4    k3    k8
C2    k6    k2    k5    k7    k9
C3    k1    k2    k3    k4
```

C1 corresponds to the C_p-function according to Berman & Brown, 1984
C3 corresponds to the Maier-Kelly equation

For all nine parameters, the implemented C_p-function at 1 bar has the following general form. In most cases only three or four parameters are used.

$$Cp = k_1 + k_2 \cdot T + \frac{k_3}{T^2} + \frac{k_4}{\sqrt{T}} + k_5 \cdot T^2 + \frac{k_6}{T} + k_7 \cdot \sqrt{T} + \frac{k_8}{T^3} + k_9 \cdot T^3 \quad [J/mol]$$

The two integrals needed for the Gibbs Free Energy function become:

$$\int_{T_0}^{T} Cp\,dT = k_1 \cdot (T - T_0) + \frac{k_2}{2} \cdot (T^2 - T_0^2) - k_3 \cdot \left(\frac{1}{T} - \frac{1}{T_0}\right) + 2 \cdot k_4\left(\sqrt{T} - \sqrt{T_0}\right) + \frac{k_5}{3} \cdot (T^3 - T_0^3)$$

$$+ k_6 \cdot (\ln(T) - \ln(T_0)) + \frac{2 \cdot k_7}{3} \cdot \left(\sqrt{T^3} - \sqrt{T_0^3}\right) - \frac{k_8}{2} \cdot \left(\frac{1}{T^2} - \frac{1}{T_0^2}\right) + \frac{k_9}{4}(T^4 - T_0^4)$$

$$\int_{T_0}^{T} \frac{Cp}{T} dT = k_1 \cdot (\ln(T) - \ln(T_0)) + k_2 \cdot (T - T_0) - \frac{k_3}{2} \cdot \left(\frac{1}{T^2} - \frac{1}{T_0^2}\right) + 2 \cdot k_4\left(\frac{1}{\sqrt{T}} - \frac{1}{\sqrt{T_0}}\right) + \frac{k_5}{2} \cdot (T^2 - T_0^2)$$

$$- k_6 \cdot \left(\frac{1}{T} - \frac{1}{T_0}\right) + 2 \cdot k_7\left(\sqrt{T} - \sqrt{T_0}\right) - \frac{k_8}{3} \cdot \left(\frac{1}{T^3} - \frac{1}{T_0^3}\right) + \frac{k_9}{3}(T^3 - T_0^3)$$

Example (Albite)

```
C1         393.63574       -2415.498   -7892826.000    1070636032.          0.00000
```

Van der Waals Equation of state for real and ideal gases, **cod = VDW**

```
VDW    a0    a1    b0    b1
```

The Van der Waals equation is $P = \frac{R \cdot T}{V - b} + \frac{a}{V^2}$

Implemented is a linear temperature dependency of the parameters a and b:
a = $a_0 + a_1 \cdot T$ [J/mol]
b = $b_0 + b_1 \cdot T$ [cm^3]

$$V^3 - V^2 \cdot \left(b + \frac{10 \cdot R \cdot T}{P}\right) - V \cdot \frac{a}{P} + \frac{a \cdot b}{P} = 0$$

This equation is solved for V° and V (at P_o and P), then

$$\int_{P_0}^{P} V\,dP = \left[V\cdot P - V^0\cdot P_0 - 10\cdot R\cdot T\cdot \ln\left(\frac{V-b}{V^0-b}\right) - a\cdot\left(\frac{1}{V}-\frac{1}{V^0}\right)\right]\cdot\frac{1}{10}$$

Exempel: (ideal gas)

```
VDW 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
```

In this example, an ideal gas is defined by setting all parameters zero.

Redlich-Kwong Equation of state for gases, `cod = R-K`

```
R-K   a0   a1   b0   b1
```

The Redlich-Kwong equation is:

Redlich-Kwong: $P = \dfrac{R\cdot T}{V-b} - \dfrac{a}{\sqrt{T}\cdot V\cdot\left(V+b\right)}$

Implemented is a linear temperature dependency of the parameters a and b:
a = a0 + a1·T [J/mol]
b = b0 + b1·T [cm^3]

$$V^3 - V^2\cdot\left(\frac{10\cdot R\cdot T}{P}\right) + V\cdot\left(\frac{a}{P\cdot\sqrt{T}} - b^2 - \frac{b\cdot 10\cdot R\cdot T}{P}\right) + \frac{a\cdot b}{P\cdot\sqrt{T}} = 0$$

This equation is solved for V^o and V (at P_o and P), then

$$\int_{P_0}^{P} V\,dP = \left[V\cdot P - V^0\cdot P_0 - 10\cdot R\cdot T\cdot \ln\left(\frac{V-b}{V^0-b}\right) - \frac{a}{b\cdot\sqrt{T}}\cdot\left(\frac{V\cdot\left(V^0+b\right)}{\mathrm{V}^0\cdot\left(V+b\right)}\right)\right]\cdot\frac{1}{10}$$

Exempel (Oxygen):

```
R-K  1.73888400E+07 0.00000000E+00 2.20990000E+01 0.00000000E+00
```

Simple volume function for solids. (as in TWQ), cod = V1

```
V1        v1*1E5     v2*1E5     v3*1E5     v4*1E8
```

Note that for the equation below, **v1, v2** and **v3** must be divided by 10^5, and **v4** by 10^8

The simple volume function as used in TWQ is:

$$V(P,T) = V^0\cdot\left[1 + v_1\cdot\left(T-T_0\right) + v_2\cdot\left(T-T_0\right)^2 + v_3\cdot\left(P-P_0\right) + v_4\cdot\left(P-P_0\right)^2\right]$$

$$\int_{P_0}^{P} V\,dP = \left(V^0 + v_{Ta}\cdot\left(T-T_0\right) + v_{Tb}\cdot\left(T-T_0\right)^2\right)\cdot\left(P-P_0\right)$$

$$+\,v_{Pa}\cdot\left(\frac{P^2}{2} - P\cdot P_0 + \frac{P_0^2}{2}\right) + v_{Pb}\cdot\left(\frac{P^3}{3} - P^2\cdot P_0 + P\cdot P_0^2 - \frac{P_0^3}{3}\right)$$

where $V_{Ta} = v_1\cdot V^0$ $V_{Tb} = v_2\cdot V^0$ $V_{Pa} = v_3\cdot V^0$ $V_{Pb} = v_4\cdot V^0$

Example (Albite)

```
V1        2.63072032      0.00032407     -0.19446932      0.00048611      0.00000000
```

Another Simple Volume function for solids. (with alpha (=f(T) and beta (=const.)), cod = V2

```
V2        aa      ab      b
```

$$\frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_{P=1} = aa + ab\cdot T \qquad \text{and} \qquad -\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_{T} = b$$

The volume function used is:

$$V = V^0 \cdot e^{aa\cdot(T-T_0)+\frac{ab}{2}\cdot\left(T^2-T_0^2\right)-b\cdot(P-P_0)}$$

$$\int_{P_0}^{P} V\,dP = \frac{V^0 \cdot e^{aa\cdot(T-T_0)+\frac{ab}{2}\cdot\left(T^2-T_0^2\right)}}{b}\cdot\left(1-e^{-b\cdot(P-P_0)}\right)$$

Volume function for solids as used by Holland and Powell (1998), cod = VHP and VH2

```
VHP     a₀     K     Tcr⁰     Smax     Vmax
VH2     a₂     Ks     dkdt
```

Note: here Q is Q^2 of Holland and Powell (1998)
P^K = P/1000 = pressure in Kilobar
According to Holland and Powell (1998):

a_2 = 10 (set constant in formula below)
K_s = 4 (set constant in formula below)
dkdt = $- 1.5\cdot10^{-4}$

$$K_t = K\cdot\left(1+dkdt\cdot(T-T_0)\right)$$

$$V = V^0\cdot\left(1+a_0\cdot(T-T_0)-20\cdot a_0\cdot\left(\sqrt{T}-\sqrt{T_0}\right)\right)\cdot\left(\frac{K_t}{4\cdot\left(P^K-P_0^K\right)+K_t}\right)^{\frac{1}{4}}$$

$$\int_{P_0}^{P} V\,dP = 10^3\cdot V^0\cdot\left(1+a_0\cdot(T-T_0)-20\cdot a_0\cdot\left(\sqrt{T}-\sqrt{T_0}\right)\right)\cdot\frac{K_t}{3}\cdot\left(\left(1+\frac{4\cdot\left(P^K-P_0^K\right)}{K_t}\right)^{\frac{3}{4}}-1\right)$$

$$T_{cr} = T_{cr}^0 + \left(\frac{V_{\max}}{S_{\max}}\right)\cdot P \qquad T_{eff} = \min(T_{cr},T) \qquad Q = \sqrt{1-\frac{T_{eff}}{T_{cr}}} \qquad Q_0 = \sqrt{1-\frac{T_0}{T_{cr}}}$$

The additional contribution of the transition to the Gibbs Free Energy becomes:

$$\Delta_{Tr}G = S_{\max}\cdot\left(\left(T_{eff}-T_{cr}\right)\cdot\left(Q-Q_0\right)+\frac{T_{cr}}{3}\cdot\left(Q^3-Q_0^3\right)\right)$$

$$+10^3\cdot V_{\max}\cdot Q\cdot\left(1+a_0\cdot\left(T_{eff}-T_0\right)-20\cdot a_0\cdot\left(\sqrt{T_{eff}}-\sqrt{T_0}\right)\right)\cdot\frac{K_t}{3}\cdot\left(\left(1+\frac{4\cdot\left(P^K-P_0^K\right)}{K_t}\right)^{\frac{3}{4}}-1\right)$$

Example (Gehlenite)

```
VHP      0.000041700         1080.0      700.0000          11.00          0.09700
VH2          10.0000         4.0000   -0.000162000
```

Volume function as used by Holland and Powell (2011), cod = V11

```
V11     a₀     k₀     k₀'     k₀''     L     Θ
```

α_0 : thermal expansion at T_0P_0
k_0 : bulk modulus at T_0P_0
k_0' and k_0'' : first and second derivative of k_0
L : a flag with one of five values:

0: no additional transition
1: phase transition described via Landau theory
2: phase transition described via Bragg-Williams theory
3: phase is an aqueous species
4: phase is a melt end-member

(For following equations, P is in Kbar)

(Θ may be input, just in case it becomes another adjustable parameter)

if Θ = 0 then

$$\theta_i = \frac{10636}{{S_i}/{n_i} + 6.44}$$ Einstein temperature of phase i

where S_i = molar entropy of phase i (at T_0P_0 ??)
n_i = number of atoms in phase i

$$\zeta = \frac{u^2 \cdot e^u}{(e^u - 1)^2}$$ function defined for α·k = n·ζ

$$\zeta_0 = \frac{{u_0}^2 \cdot e^{u_0}}{(e^{u_0} - 1)^2}$$

where u = Θ/T
u_0 = Θ/T_0

$$P_{th} = \alpha_0 \cdot k_o \cdot \frac{\theta}{\zeta_0}\left(\frac{1}{e^u - 1} - \frac{1}{e^{u_0} - 1}\right)$$ thermal pressure term

$$V = V_0\left(1 - a\left(1 + b(P - P_{th})\right)^{-c}\right)$$ volume pre-2023.01.02beta release

$$V = \frac{V_0(1 - a\ (1 - (1 + b\ (P - P_{th})))^{-c})}{(1 - a\ (1 - (1 + b\ P_0))^{-c})}$$ volume as of 2023.01.02beta release

where
$$a = \frac{1 + k_0'}{1 + k_0' + k_o \cdot k_0''}$$
$$b = \frac{k_0'}{k_0} - \frac{k_0''}{1 + k_0'}$$
$$c = \frac{1 + k_0' + k_0 \cdot k_0''}{(k_0')^2 + k_0' - k_0 \cdot k_0''}$$
$$P_0 = 0.001\ kbar$$

Gibbs Free Energy:

$$\Delta G_{P,T} = \Delta G_{P_0,T} + P \cdot V_0\left(1 - a + \frac{a\left((1 - b \cdot P_{th})^{1-c} - \left(1 + b(P - P_{th})\right)^{1-c}\right)}{b(c-1)P}\right)$$ pre-2023.01.02beta release

$$\Delta G_{P,T} = \Delta G_{P_0,T} + V_0\left(\frac{(1-a)(P - P_0) + \frac{a\ ((b\ (P_0 - P_{th}) + 1)^{1-c} - (b\ (P - P_{th}) + 1)^{1-c})}{b\ (c-1)}}{(1 - a + a\ (1 + b\ P_0)^{-c})}\right)$$ as of 2023.01.02beta release

note: $(P-P_0)V_0$ .. etc. etc. used where required

Example (Forsterite)

```
V11     0.000028500          1285.00           3.8400       -0.003000             0.00
```

Landau order function as used by Holland and Powell (2011), cod = LA1 (must be used with above volume function, cod = V11)

LA1 T_{cr}^{0} S_{max} V_{max}

(For following equations, P is in KBar)
Q defined as in Holland and Powell (1998)
T_{cr}^{0} : Critical temperature at 1 Bar
S_{max} : entropy of ordering at T_{cr}
V_{max} : volume of ordering at T_{cr}
k_0 : bulk modulus at T_0P_0 (from volume function, cod = V11)
α_0 : thermal expansion at T_0P_0 (from volume function, cod = V11)

$$T_{cr} = T_{cr}^{0} + \frac{V_{max}}{S_{max}} \cdot P$$
$$T_{ef} = min(T_{cr}\,, T)$$
$$Q_0^4 = \left(1 - \frac{T_0}{T_{cr}^0}\right) \qquad Q^4 = \left(1 - \frac{T_{ef}}{T_{cr}}\right)$$
$$k = k_0\big(1 - 1.5 \cdot 10^{-4}(T - T_0)\big)$$
$$\Delta G_{Land} = S_{max}\left(\left(T_{ef} - T_{cr}\right)Q^2 + \frac{1}{3}T_{cr}Q^6\right)$$
$$v_T' = V_{max}Q_0^2\left(1 + a_0\left(T_{ef} - T_0\right) - 20 \cdot a_0\left(\sqrt{T_{ef}} - \sqrt{T_0}\right)\right)$$
$$\int v_T' dP = \frac{1}{3} v_T' \cdot k\left(\left(1 + \frac{4P}{k}\right)^{3/4} - 1\right)$$
$$h_0' = S_{max}T_{cr}^0\left(Q_0^2 - \frac{1}{3}Q_0^6\right)$$
$$s_0' = S_{max}Q_0^2$$

$$G^{excess} = h_0' - T \cdot s_0' + \int v_T'\, dP + \Delta G_{Land}$$

Example (Sphene)

```
LA1            485.00          0.4000         0.00500
```

Bragg-Williams order function as used by Holland and Powell (2011), cod = BW1

BW1 ΔH_{BW} ΔV_{BW} W_H W_v n fac

Note: all units in [J] not [KJ]

ΔH_{BW} : Enthalpy of disordering

ΔV_{BW} : Volume of disordering
W_H and W_V : Interaction terms used in $W = W_H + P \cdot W_V$
n : number of Si disordering with each Al
fac : mystical factor

$W = W_H + P \cdot W_V$
$A = \Delta H_{BW} - W + P \cdot \Delta V_{BW}$
$B = 2 \cdot W$
(W, A and B are divided by fac)
Q = ordering parameter

For versions older than 2023.01.02beta, G^{excess} for ordering is calculated as follows:

At equilibrium: $A + B\cdot Q + \frac{n}{n+1}RT\cdot ln\left(\frac{(n-nQ)(1-Q)}{(1+nQ)(n+Q)}\right) = 0$

Solve iteratively for Q then:

$$a^{id} = \frac{1}{(n+1)^{n+1}}\cdot(1+nQ)(n+Q)^n$$
$$RTln(a) = RTln\left(a^{id}\right) + W(1-Q)^2$$
$$G^{excess} = RTln(a)\cdot fac$$

Starting with version 2023.01.02beta, the calculation is modified to also accommodate cases where fac is given as a negative value by the database; G^{excess} for ordering is calculated as follows:

If fac >0, fac1=fac2=fac; If fac<0, fac1=1 and fac2=-fac

At equilibrium: $A + B\cdot Q + \frac{n}{n+1}RT\cdot\left(fac1\cdot ln\left(\frac{(n-nQ)}{(1+nQ)}\right) + fac2\cdot ln\left(\frac{(1-Q)}{(n+Q)}\right)\right) = 0$

Solve iteratively for Q and calculate G^{excess} appropriately.

Exemple (Albite)

```
BW1      14000.00      0.04200     13000.00      0.04200          3.0        0.9000
```

Lambda transitions 1.

```
T1    T°tr    T°ref    l1    l2    ΔHtr
T2    Tq       ΔVtr    (dV/dT)    (dV/dP)
```

According to Berman and Brown, 1985, Contributions to Mineralogy and Petrology, 89, 168-183
Between the limits T_{ref} and T_{tr} a Cp-like function is added.

$$Cp_{tr} = (T-\Delta T)\cdot\left(l_1 + l_2\cdot(T-\Delta T)\right)^2$$

It is assumed, that T_{tr} (transition) is a linear function of P with slope Tq.
$\Delta T = (P-P^0)\cdot Tq$ $\quad T_{ref} = T^0_{ref} + \Delta T,$ $\quad T_{tr} = T^0_{tr} + \Delta T,$
V and ΔG is adjusted for ΔH_{tr}, ΔV_{tr}, (dV/dT) and (dV/dP) for low T polymorph.

For $T<T_{tr}$:

$$\Delta_{tr}G(T) = \int_{T_0}^{T} Cp_{tr}\,dT - T\cdot\int_{T_0}^{T}\frac{Cp_{tr}}{T}dT + (dV/dT)\cdot(P-P_0)\cdot(T-T_0) + \frac{(dV/dP)}{2}\cdot\left(P^2-P_0^2\right) - (dV/dP)\cdot(P-P_0)$$
$$\Delta_{tr}V = (dV/dT)\cdot(T-T_0) + (dV/dP)\cdot(P-P_0)$$

For $T>T_{tr}$:

$$\Delta_{tr}G(T) = \Delta_{tr}G(T_{tr}) - \frac{\Delta H_{tr}}{T^0_{tr}}\cdot(T-T_{tr})$$

One phase may have more than one transition.

Exemple (Quartz)

```
T1           848.00          373.00    -0.09186959     0.00024607            0.00
T2           0.023743        0.000000   0.0000000000   0.0000000000          0.00000
```

Cp and Lambda transitions 2.

```
CSK     a    b    c    TTr0    ΔHtr    ΔVtr    Tslop
```

According to Helgesson et al., 1978, Amer. Jour. Sci., v. 278A, 229 p. (as used in SUPCRT)
The Cp-function is divided into n (up to four) temperature ranges:

$T_0 < T < T_{tr}^0(1)$: $$Cp = a(1) + b(1)\cdot T + \frac{c(1)}{T^2} \quad [J/mol]$$

$T_{tr}^0(1) < T < T_{tr}^0(2)$: $$Cp = a(2) + b(2)\cdot T + \frac{c(2)}{T^2} \quad [J/mol]$$

...

$T_{tr}^0(n\text{-}1) < T$: $$Cp = a(n) + b(n)\cdot T + \frac{c(n)}{T^2} \quad [J/mol]$$

Exemple (Quartz)

```
CSK       46.94448     0.0343088    -1129680.0       848.000       1213.36        0.3720       38.5000
CSK       60.29144     0.0081170           0.0         0.000          0.00        0.0000        0.0000
```

Disorder contributions

```
D1    d1    d4    d3    d8    d6
D2    d2    d5    To    Td    Vad
```

Disorder contributions to G can be approximated by adding a cp-like function between the limits To and Td.

$$Cp_{dis} = d_1 + d_2\cdot T + \frac{d_3}{T^2} + \frac{d_4}{\sqrt{T}} + d_5\cdot T^2 + \frac{d_6}{T} + d_7\cdot\sqrt{T} + \frac{d_8}{T^3} + d_9\cdot T^3 \quad [J/mol]$$

The volume is taken from the shape of the enthalpy function and is scaled to the data by the variable V_{ad}.

$$\Delta_{dis}V = \frac{\int_{To(>T)}^{\min(T\,Td)} Cp_{dis}\,dT}{10\cdot V_{ad}}$$

The total contribution to the Gibbs Free Energy becomes:

$$\Delta_{dis}G = \int_{To(>T)}^{\min(T\,Td)} Cp_{dis}\,dT - T\cdot\int_{To(>T)}^{\min(T\,Td)} \frac{Cp_{dis}}{T}\,dT + \left(P - P_0\right)\cdot\frac{\int_{To(>T)}^{\min(T\,Td)} Cp_{dis}\,dT}{10\cdot V_{ad}}$$

Example (Gehlenite)

```
D1       -221.74000         0.00000 17291280.00000         0.00000         0.00000
D2          0.36950        -0.00015       698.15000      1600.00000         0.00000
```

Special phases:

```
SPC     keyword
```

If the equation of state cannot be expressed with the above keywords, G at P and T may calculated in an external subroutine (fsol.f). All other data lines will be ignored. Used e.g. to calculate G for water and steam according to Haar et al. (1982). The maximum length of the keyword is 10 characters.

In the file fsol.f, the following equations of state are present and reasonably tested:

Keyword	description
HAAR	equation of state for water and steam, according to Haar et al. (1982)
ABSAL	compute thermodynamic properties of albite ordering relative to low albite standard state. Function of Salje (1985), used with albite from database of Berman et al. (1992)
CK&J	CO2 according to Kerrick and Jacobs (1981)
WK&J	H2O according to Kerrick and Jacobs (1981)

CHP98	CO2 according to Holland and Powell (1998)
HHP98	H2O according to Holland and Powell (1998)

Example (H2O)

```
SPC    HAAR
```

Combination of phases:

```
COM    name[value] name[value]...     activity     Rname
```

To the ΔG defined by the standard state, Cp's etc. we can add any combination of previously calculated ΔG's. This is also useful e.g. for polymorphs or buffers. The thermodynamic parameters calculated are $\Delta_a G$ and Volume.

name   name of phase. (must be in the database before the com-line)

value   amount of phase used in the combination (the value may be a fraction, like 1/3)

activity (optional) activity of phase. Note that S^0 is set to -R·ln(a). (overwrites existing S^0) A value of zero means no activity correction (Useful if Rname is used)

Rname (optional) name of phase that should not be included in calculation of stable assemblage.(This may be important for buffers where ΔrG (buffer = phases) is zero, which results in an indifferent, non-unique equilibrium)

Example (Buffer)

```
COM      MAGNETITE[2] STEAM[1] HEMATITE[-3]     0     HEMATITE
```

USE and CODE (selecting phases)

The phases in the database may be divided into different groups. Each group is coded as one character (e.g. A, B, C, D). The variable "CODE" in the database (optional) defines for each phase to which groups it belongs. When the program is run we can specify in the variable "USE" all groups that should be considered. If USE = '*' then all groups are considered, including those with no CODE defined.

special phases:

If the first character of CODE is a '*' then the phase is "special" (e.g. a buffer) and does not belong to a specific group. It will only be considered if mentioned in USE after a comma. The first comma in USE is a delimiter between the group names and the special phases.

If the first character of CODE is a '+' then G of this phase will be calculated even if its composition is outside the compositional space. In that case this phase can still be used to define G of other phases, but will itself not be considered for the equilibrium assemblage.

If the first character of CODE is a '-' then the phase will be excluded from becoming stable.

In general a phase will be considered if:

- The group code is matching
- No phase with identical name is already considered.
- Its components are a subsystem of the bulk components.
- Enough data to calculate G is provided. (Standard state and Cp line)

Examples for USE and CODE

	CODE	USE	Considered phases
Phase1	A	A	1,4
Phase2	BC	B	2
Phase3	C	C	2,3,4
Phase4	AC	AB	1,2,4
Phase5		AC	1,2,3,4
Phase6	*QFM	BC	2,3,4
Phase7	*CB	ABC	1,2,3,4
		*	1,2,3,4,5
		AB,QFM	1,2,4,6
		AB,QFM,CB	1,2,4,6,7

```
                    ABC ,CB              1,2,3,4,5,7
                       A,CB                    1,4,7
```

Section *** ... SOLUTION DATA ...

A solution phase is defined by one definition line followed by any number of endmember lines.

Solution definition line:

```
SOLNAME    (modell)[mul]    s1(m1):e11,e12... - s2(m2):e21,... - ...
```

A line within this section is recognized to be a solution definition line if it contains the string '..(' (blank,blank,left parenthesis) and is not a comment line.
SOLNAME: Name of solution phase (maximum 16 characters)
modell: This string is scanned for certain keywords:

> **SKIP**: the solution phase will not be considered.
> **EXCL**: the solution phase will not be considered for an equilibrium, however the chemical potential of the solution with the smallest Gibbs Free Energy will be calculated. If a table or pixelmap is produced this chemical potential is recorded as G_solname and Gneg_solname.
> **IDEAL**: An ideal solution model is assumed. ($ai=xi^{mul}$). This is also the default. si, mi: and eij are not used to calculate the activities, however the output will include the site occupancies.
> **SITE**: An ideal site mixing is used, where:
>
> > si: name of site i.
> > mi: multiplicity of site i
> > eij: possible elements for site i.
>
> **EXT**: The activities of the endmembers are defined in an external subroutine (e.g. in file fsol.f). Other keywords (except SKIP and MARGULES) are ignored. si, mi: and eij have no influence
> **MARGULES**: This keyword states, that there may be Margules parameters defined for this phase. They will only be added if this keyword appears in 'modell'.

mul: If a number follows immediately the (modell), then this number is used to calculate multiplicities. The number may be a fraction, like "1/3"
The element names (max. 8 characters) do not have to correspond to a compositional element. They may represent element groups or coupled elements.
NOTE: A site mixing model which is not conform with THERIAK's philosophy will produce nasty warning messages.

For ideal site mixing the following general formula is used to define the activities:
e.g. MÄDER, U.K., PERCIVAL, J.A. & BERMAN, R.G. (1994): Thermobarometry of garnet – clinopyroxene – hornblende granulites from the Kapuskasing structural zone. Can. J. Earth Sci. 31, 1134-1145.

$$(a_{ideal})_j = \prod_s \prod_m \left(\frac{n_s}{r_m} x_m \right)^{r_m}$$

n_s = multiplicity of site (s)
r_m = number of ions (m) on site (s) for phase (j)
x_m = mole fraction of ions (m) on site (s)

Endmember lines:

```
 EMNAME    e11,e12,.. - e21,... -     α0   αT   αP
```

EMNAME: Name of the endmember. (maximum 16 characters). The endmember will be added to the solution phase if a considered phase with the same name exists.
eij: the mi elements occupying site i of the endmember.
α0, αT, αP: coefficients for the asymmetric van Laar model, according to Holland and Powell (2002).

> α_i is the van Laar parameter for endmember i.
> $\alpha_i = \alpha0 + \alpha T*T + \alpha P*P$ (T=Temperature, P=Pressure [Bar])

The contribution ΔG_{ij}^{ex} to ΔG from a single binary interaction parameter W_{ij} becomes:

$$\Delta G_{ij}^{ex} = \frac{2\cdot x_i\alpha_i \cdot x_j\alpha_j}{\sum_{k=1}^{ne} x_k\alpha_k \cdot \left(\alpha_i + \alpha_j\right)} \cdot W_{ij}$$

where ne is the number of endmembers in the solution

Examples:

```
GAS PHASE     (IDEAL)
  CARBON DIOXIDE
  STEAM
  OXYGEN
  CARBON MONOXYDE
H-FELDSPAR    (IDEAL,MARGULES)
  H-ALBITE
  H-SANIDINE
GARNETS (MARGULES,IDEAL)3     M(3):Ca,Mg,Fe
  GROSSULAR   Ca,Ca,Ca
  PYROPE      Mg,Mg,Mg
  ALMANDINE   Fe,Fe,Fe
FSP    (MARGULES,IDEAL)  M(1):Na,K,Ca
   sanidine       K      1.0     0    0
   high-albite    Na     0.643   0    0
   anorthiteC1    Ca     1.0     0    0
```

Section *** ... MARGULES ...

The Margules parameters are grouped in subsystems (binary, ternary etc.) Each subsystem is defined by a definition line followed by any number of parameter lines.

Margules definition line

```
EMNANE1 - EMNAME2 - ...
```

A line within this section is recognized to be a definition line if it contains the string ' - ' (blank, minus, blank).
EMNAMEn: Name of nth endmember in subsystem.

parameter lines:

```
k1k2k3...          WH    WS    WV    WCP    K
```

The number of k's defines the degree (p) of the polynomial in the Margules equation. This does not have to be the same for all parameters. The generalized equation used is:

$$\Delta G^{ex} = \sum W^{G} \cdot \frac{x_{k_1} \cdot x_{k_2} \cdot ... \cdot x_{k_p}}{\left(s\right)^{K}}$$

S = sum of different components x_j involved in polynomial $x_{k1}.x_{k2}....x_{kp}$

K = Parameter to extrapolate binary data to higher order systems. This follows the ideas of Kohler (1960) and produces results comparable to the better known Wohl (1964) equation. (See e.g. de Capitani and Kirschen, 1994)

$W^G = W^H + W^{CP}\cdot (T-T_o)-(W^S + \ln(T/T_o)\cdot W^{CP})\cdot T + W^V\cdot P$

Example

```
***** SOLUTION DATA  *****
H-FELDSPAR    (IDEAL,MARGULES)
  H-ALBITE
  H-SANIDINE
!
***** MARGULES PARAMETERS *****
H-ALBITE - H-SANIDINE
112    32098.8    16.1356    0.46903   0   0
122    26470.9    19.3807    0.38702   0   0
```

Section *** ... SITMARG ...

Margules parameters defined for site-mixing

Site-Margules definition line

```
* SOLNAME
```

A line within this section is recognized to be a definition line if it contains the string '* ' (star,blank).
SOLNAME: Name of solution phase to which the site Margules should be applied.

parameter lines:

```
s – el1el2el3...    WH   WS   WV
```

s : name of the site as defined in the solution definition line.
eli: names of elements for mixing as defined in the solution definition line.
The number of el's defines the degree (p) of the polynomial in the Margules equation. This does not have to be the same for all parameters. The generalized equation used is:

$$\Delta G^{ex} = \sum W^{G} \cdot x_{el_1}^{s} \cdot x_{el_2}^{s} \cdot \ldots \cdot x_{el_n}^{s}$$

x_{eli}^{s} = concentration of element (i) on site (s)
$W^G = W^H - W^S \cdot (T - T_o) + W^V \cdot P$

Example

```
***** SOLUTION DATA  *****
BIO     (-SITE)   M2(2):Mg,Fe - M1(1):Mg,Fe,Al - T2(2):Al,Si
  PHLOGOPITE         Mg,Mg - Mg - Al,Si
  ANNITE             Fe,Fe - Fe - Al,Si
  SIDEROPHYLLITE     Fe,Fe - Al - Al,Al
  EASTONITE          Mg,Mg - Al - Al,Al
!
***** SITEMARG *****
* BIO
M2 - Mg,Fe        12769.0     10.200     -.060
M1 - Mg,Fe         6384.6      5.100     -.030
M1 - Mg,Al        -6520.0    -18.977     0.000
M1 - Fe,Al        27270.1     29.983     0.000
```

Section *** ... SEEDS ...

Initial guesses for the minimizations
Each line defines one initial guess with the following format:

```
SOLNAME  Em1  x1  Em2  x2  Em3  x3   etc.
```

SOLNAME: Name of solution phase
Em1: Name of first endmember
X1: Concentration of first endmember
etc.

<u>Each initial guess must be on one single line.</u>
When the print-code in the dat-file (first character in THERIN) is “1” (long output) the result printed at the end of the theriak run can be used as seeds.

Exempel

```
PHNG  Muscovite   6.34806424E-01  Celadonite   2.19890500E-01  Fe-Celadonite
8.95907168E-02  Paragonite   5.39715500E-02  Margarite2   1.74080966E-03  FeMusc
8.01769453E-11
```

```
ClAMg  tremolite   8.17905247E-02  tschermakite2   1.12218622E-01  pargasite2
4.29616696E-01  glaucophane2   3.11389590E-01  cummingtonite   4.98053328E-03  grunerite2
3.00056441E-01  acam  -2.43150519E-01  bcam  -3.82003419E-02  mgriebekite
0.00000000E+00  kpargasite   2.14781407E-02  ttschermakite   1.98203137E-02

CPXo  jadeite   1.00401610E-01  diopside   1.72170370E-03  omphacite   6.32092292E-01
clinoFM   2.65784395E-01  jadacm   0.00000000E+00

CHTD  Mg-Chloritoid  -4.82160477E-01  Fe-Chloritoid   5.41981319E-01  F3-Chloritoid
9.40179158E-01
```

Section *** ... FIXED PHASES ...

For some specific applications it is useful to include solution phases with fixed compositions to the calculations. These may be included for comparison (calculated equilibria versus measured compositions) or replacing a solution (metastable equilibria).

Fixed phase definition line

fcod   fname   SOLNAM $EM_1(X_1)EM_2(X_2)...EM_{ne}(X_{ne})$

each line in this section defines one fixed solution phase

fcod=0 the fixed phase is added for comparison. Equilibrium calculated including solution SOLNAM.

fcod=1 The solution phase SOLNAM is not included in the equilibrium calculation. Only the fixed phases are considered.

fname name of the fixed composition phase (Max. 16 characters)

SOLNAM Name of the solution used to define the fixed composition phase

$EM_i(X_i)$ Names of the endmembers and their concentrations. One blank optional between items.

Example

```
***** FIXED PHASES *****
0    PL1     FSP          ALBITE(0.83) ANORTHITE(0.15) K-FELDSPAR(0.02)
0    PL2     FSP          ALBITE(0.70) ANORTHITE(0.30) K-FELDSPAR(0.00)
```

Calculation of G in an external subroutine

If ΔG is to be calculated in an external subroutine, (data line: SCP keyword), then the program calls a subroutine named GSPEC. The parameters transmitted are: NAME (name of phase), P (pressure), PGAS (pressure of fluid phases), T (temperature), CASE (=keyword), G (ΔG) and V (volume). The subroutine calculates G and V, then returns control to the main program.

Example of GSPEC:

```
SUBROUTINE GSPEC(NAME,P,PGAS,T,FALL,G,V)
CHARACTER*16 NAME
CHARACTER*10 CASE
REAL*8 P,PGAS,T,G
G=0.0D0
IF (CASE.EQ.'HAAR') CALL WHAAR2(NAME,PGAS,T,G,V)
IF (CASE.EQ.'CK&J') CALL CO2KJ(PGAS,T,G,V)
IF (CASE.EQ.'HOPO') CALL HOPOH2O(PGAS,T,G,V)
RETURN
END
```

Calculation of activities in an external subroutine.

If the model of a solution phase contains the string 'EXT' then three subroutines will be called:

SOLINI. In this subroutine we define for a given solution phase the number and the names of the endmembers.
SOLMOD. This subroutine returns a string to be printed, and which describes the activity according to solution name and endmember number. (optional).
SOLCAL. For each endmember the activity is calculated.

The solution has to be defined in the database, but may have fewer endmembers and in a different order than the same solution in the external subroutine.
The calling parameters are relatively complex, but to add a new solution phase to the already existing subroutines, only few lines have to be inserted.

Examples for using external subroutines:

```
in SOLINI:
...
IF (SOLNAM.EQ.'FELDSPAR') THEN
N=3
NAME(1)='ALBITE'
NAME(2)='POTASSIUM FELDSP'
NAME(3)='ANORTHITE'
END IF
IF (SOLNAM.EQ.'H2O-CO2') THEN
N=2
NAME(1)='STEAM'
NAME(2)='CARBON DIOXIDE'
END IF

in SOLMOD:
MODELL='NOT EXPLICITLY DEFINED'
...
IF (SOLNAM.EQ.'FELDSPAR') THEN
IF (K.EQ.1) MODELL='X(Ab)*(1-X(An)**2)'
IF (K.EQ.2) MODELL='X(Or)*(1-X(An)**2)'
IF (K.EQ.3) MODELL='(X(An)*(1+X(An))**2)/4'
END IF
IF (SOLNAM.EQ.'H2O-CO2') THEN
MODELL='See: Kerrick & Jacobs(1981)'
END IF
```

in SOLCAL:

```
...
IF (SOLNAM.EQ.'FELDSPAR') THEN
A(1)=X(1)*(1.0D0-(X(3)**2))
A(2)=X(2)*(1.0D0-(X(3)**2))
A(3)=(X(3)*(1.0D0+X(3))**2)/4.0D0
RETURN
END IF
IF (SOLNAM.EQ.'H2O-CO2') THEN
XCO2=X(2)
CALL MIXKJ(T,P,XCO2,F1,F2,A1,A2)
A(1)=A2
A(2)=A1
RETURN
END IF
```