

Datafile construction for DRAWPD 1.0

[back](#)

DRAWPD is the new software that takes mode 1 THERMOCALC output and produces raw postscript output of the phase diagram. Consider that a pseudosection is being constructed, and an appropriate THERMOCALC datafile exists. In the datafile there needs to be the script

```
drawpd yes
```

This switches on the outputting of information to be used in making a DRAWPD datafile. The default name of the output file is TH DR (but this can be changed in the TH PREFS if desired). If the THERMOCALC datafile is called, say, TH DBLAH, then it is suggested that the DRAWPD datafile be called DR DBLAH. The raw postscript output file produced by DRAWPD will then be called DRBLAH.PS. The DRAWPD log file will be called DR LOG.

When THERMOCALC is run in the normal way, building up the pseudosection line-by-line, each time THERMOCALC is quit, the user copies the contents of the TH DR file into the DR DBLAH file. (This should be done every time THERMOCALC is quit because each time THERMOCALC is started again the TH DR file is overwritten). The block of information for each equilibrium found in the TH DR file will need a small amount of editing, as outlined below. In addition to the blocks of information for each equilibrium, instructions that control the behaviour of DRAWPD have to be included in the DRAWPD datafile.

Overall datafile structure

The overall DRAWPD datafile structure is

1. initial setup
2. the description of the points/lines (THERMOCALC output), terminated by a “*”
3. definition of areas to be shaded/coloured, terminated by a “*”
4. scripts controlling phase diagram “window” etc, terminated by a “*”
5. storage area (not read by drawpd)

The code used to read the DRAWPD datafile is based on the THERMOCALC datafile reading code, so they share various things in common. For example, in the following, note that anything after a “%” on a line is treated as a comment, annotating what is in the datafile.

Initial setup

The initial setup primarily involves defining what will be the x -axis and y -axis of the phase diagram. An example initial setup illustrates what is involved, this one being for a T - x pseudosection for a composition in NCKFMASH at 5 kbars :

```
% == initial setup ==

3      % no of variables in each line of data,
      % in this case, x, P, T

6      % effective system size: 8 (NCKFMASH) - 2 (+q+pl) = 6

1 3    % which columns to be x,y in phase diagram

x 2 5  % x keyword turns on interpolation for points,
      % involving column 2 at value 5 (kbar)

% == start of points/line ==

% -----
u1    liq ksp cd g sill  - bi

i2 i4

0.0500 5.00 727.7
0.1000 5.00 728.0
0.1500 5.00 728.3
0.2000 5.00 728.6
...
```

To understand the initial setup, look at the first lines of the THERMOCALC information for u1 (the divariant to trivariant boundary where the mode of biotite goes to zero in the assemblage bi + sill + g + cd + ksp + pl + q + liq). Each line involves 3 numbers (x , P and T). The first number in the initial setup is this number, 3. The second is the effective size of the system, 6. (This number is not currently used, but is there for forward compatibility—plans are afoot for DRAWPD to do reality checking of the phase diagram information in the datafile). The next two numbers (1 3), say that the phase diagram involves x (1st column) as the x axis, and T (3rd column) as the y axis.

The last line of the initial setup need only be there, when, as here, the diagram to be drawn is a T - x (or P - x) pseudosection. As THERMOCALC cannot (currently) calculate the T - x coordinates of points in pseudosections directly, but only the P - T coordinates at specified x , DRAWPD allows the user to interpolate such information. The input (x 2 5), communicates that this is a T - x (or P - x) pseudosection (via the keyword, “x”), that it is the 2nd column that needs to be interpolated (the 2) and the interpolation value is 5 kbars (the 5). Use of this follows below.

Points and lines

An example of THERMOCALC output for a line is

```
% -----  
u<k>   liq cd g sill  - ksp  
  
begin  end  
  
0.8200 5.00 743.0  
0.8300 5.00 740.7  
0.8400 5.00 738.4  
0.8500 5.00 736.1  
0.8600 5.00 733.8  
0.8700 5.00 731.5  
% -----
```

The first thing that must be changed is that the <k> in u<k> must be replaced by, say, 17, making u17, if this is line 17 in the sketch you are making as you construct the pseudosection. The second thing that usually has to be changed is the “begin end”, that refers to the part of the line which is to be drawn. If, for example, the line is to run from the beginning of the list through to i4 (ie point 4), then this should be replaced by “begin i4”. If, for example, the line is to run from i7 to the end of the list, then this should be replaced by “i7 end”. If, for example, the line is to run from i6 to i8, then this should be replaced by “i6 i8”.

Sometimes line segments between points are so short that it is sufficient to just connect them with a straight line. This is done by, for example

```
% -----  
u11   ksp bi cd sill H2O - liq  
  
i10 i12 connect  
% -----
```

obviously without there being a list of coordinate information defining the line. The “connect” keyword tells DRAWPD that no list follows.

The header information for each equilibrium—in this case, the phases with non-zero modes involved at this boundary (ksp bi cd sill H2O), and which phases have zero mode (liq)—is optional. In the u11 case it would have to be user-provided, given that the information for this equilibrium is not provided by THERMOCALC. In a future version, DRAWPD will have the facility to do a reality check of the information provided, and this header information will be critical for that. Otherwise it is included for completeness (and debugging). Also reality check-related, a legal keyword to end the header information is “univariant”, and the header may just be the keyword “line” (for example if some construction lines are to be included in the diagram)

Points are handled in much the same way as lines, except their names start with i. In the absence of the “x” keyword in the initial setup, a point will look like

```
% -----
  i3  liq bi g sill  - ksp cd

  0.7400 5.00 733.2
% -----
```

with just the coordinates of the point (in this case, at the exact P of the diagram being drawn: 5 kbars).

In the presence of the “x” keyword in the initial setup, a point may look like

```
% -----
  i2  ksp cd g sill  - liq bi

  0.0200 5.55 743.8
  0.0300 5.30 736.4
  0.0400 5.05 729.0
  0.0500 4.79 721.8
  0.0600 4.53 715.1
  0.0700 4.29 709.1
  0.0800 4.06 703.9
% -----
```

and, on running, DRAWPD will interpolate the T - x at $P = 5\text{kbars}$ from these numbers. Even if the “x” keyword is present, information like that for i3 above, is fine, as long as the given P is correct.

Sometimes THERMOCALC struggles to calculate particular equilibria (for algorithmic reasons). If this occurs for points, and two of the equilibria involved at the point cross there, there is a keyword “crossover” that allows the point to be calculated as where the lines cross. An example shows the syntax

```
% -----
  i2  ksp cd g sill  - liq bi

  crossover u8 u9
% -----
```

This tells DRAWPD to find where i2 is by finding where the lines u8 and u9 cross. It is worth noting that at an effectively-invariant point in a pseudosection, where an n , two $n + 1$, and an $n + 2$ variant fields meet, that the boundaries between the $n + 1$ and $n + 2$ variant fields will cross (and can be used by crossover), whereas the boundaries between the n and $n + 1$ variant fields cannot cross (and cannot be used by crossover) as they die at the point.

Logic regarding the header information for points is the same as for lines, with the corresponding possible keywords, 'invariant' and 'point'.

Areas

Shaded or coloured areas on phase diagrams, particularly pseudosections make them much easier to read, at least if it is done properly. My view is that, on pseudosections, divariants should be white, and fields should be darker with increasing variance.

This section of the datafile allows this to be achieved. However, if there are no areas to do, then this section of the datafile can be left blank (so it will involve two consecutive “*”). Areas are just set up by the lines that define them entered in order around the area. Matters relating to edges and corners *should* be handled transparently, not requiring user attention. (Tell me if this is not the case!)

The areas section might look like

```
*           % end of points/lines; start of areas
% -----

0.8 u5 u1 u12 u6    % trivariant
0.8 u7 u2 u8
0.8 u9 u3 u10
0.8 u11 u4 u17

0.6 u6 u19         % quadrivariant
0.6 u12 u13 u14
0.6 u17 u15 u18
0.6 u10 u11

0.4 u19 u13 u18    % quinivariant
0.4 u20

*           % end of areas; start of scripts
```

Each line starts with a number between 0 and 1: nearer 0 means darker; nearer 1 means lighter. By default these numbers produce grayscale, but a script allows colour. Usually all the fields of a particular variance will be given the same grayscale/colour. After the number between 0 and 1 is simply a list of the lines, in order around the area. The “closing” of the area is done DRAWPD (so you do not need to repeat the starting line in the list of lines)

Scripts

An example of scripts to draw a phase diagram (in fact a blow-up of part of the scripts for a T - x diagram)

```
% -----  
window 0.83 0.89 732 736 % x,T window  
  
bigticks 0.02 0.83 1 732 % main x ticks at 0.02 intervals, starting at 0.83  
% main T ticks at 1 intervals, starting at 732  
  
smallticks 0.005 0.2 % minor x,T ticks  
  
darkcolour 0 0 255 % red green blue in 0<->255 => Illustrator RGB  
% this choice gives the diagram in blues  
  
showlineinfo yes % puts line info into log file  
doareas yes % uses the area info  
numbering no % numbering the lines on the diagram  
% -----
```

There are defaults for all these scripts *except* “window”: this defines the dimensions of the phase diagram to be drawn (x_{low} , x_{high} , y_{low} , then y_{high}).

“bigticks” and “smallticks” control the ticks along the axes of the diagram, and the axes annotations are based on “bigticks”.

“darkcolour” controls what 0 would mean in the 0-to-1 scheme used in areas. If the three integers given after the keyword are the same (for example “0 0 0”), then the colours will actually be grayscale. Experiment to produce something that is aesthetic (for you).