thermocalc scripting (for `tc350`, Nov 2020)

Scripting allows customisation of the running of THERMOCALC. This is a good idea because each time you run the program you would otherwise be met with a barrage of questions many of which you might know in advance are not relevant or that can be set ahead of time.

- scripts reside in the prefsfile or the scriptfile. Generally they involve one or more keywords[1] followed by qualifying information. For example, in "inexcess mu q H2O ", "inexcess" is the keyword, telling THERMOCALC that the calculation is to have muscovite, quartz and H_2O in excess[2]

- in general, keywords can be followed immediately by "yes", "ask" or "no", meaning—respectively— yes, use the option indicated by the keyword, ask (interactively) whether the option is to be used, or no, ignore this option. "Yes" is implied if nothing given after a keyword (readability of scripts is better if yes is omitted if that is the action required, and this is now the recommended way of writing scripts). This implicit "yes" is present in "inexcess mu q H2O ", with "inexcess yes mu q H2O " being exactly equivalent. With "inexcess no mu q H2O", there are to be no in excess phases. With "inexcess ask mu q H2O ", there is a prompt asking whether to use these in excess phases. (if there is no "inexcess" script, then you are prompted for which phase(s) to have in excess).

- "ask" is not always a sensible option, in which case THERMOCALC replaces "ask" by "yes" or "no", by context. "Ask" is generally downplayed now given the increased emphasis on calculations being script-driven (and the envisaged lesser use of interactive input).

- in the past, when much input to THERMOCALC was interactive via prompts on the screen, wrong or misunderstood scripts were ignored silently by THERMOCALC. Now that the scripts play a much more central role, THERMOCALC is fairly aggressive about handling wrong scripts. It assumes that you really want to do what the scriptfile was trying to tell THERMOCALC to do, so THERMOCALC tries to help that happen.

- a new idea is that some scripts are now mandatory, allowing for better checking that the scripting is ok. If a mandatory script is not present then THERMOCALC guides you in including it. Mandatory scripts are either "yes" or "no" ("ask" is not available).

- there now exists scripts to allow full hands-off running of THERMOCALC, for example, for driving THERMOCALC from Python. This means that THERMOCALC can be run, a

[1] In text, keywords are given in quotes; in standalone examples, they are given in a typewriter font.
[2] Many of the examples in these notes relate to simple calculations in AFM (ie KFMASH, + mu + q + H_2O.

calculation at a time, without interactive input at all, just by changing what is in the scriptfile.

- critically, the scripts are still in flux and they may not all work as intended at this stage—please notify us if things don't look right. (Send your scriptfile and logfile so that we can reproduce what THERMOCALC did or didn't do for you, and need to indicate what you were expecting).

There are two places that scripts are read by THERMOCALC :

1. in a file—the prefsfile—called `tc-prefs`: these scripts control some global aspects of running THERMOCALC
2. in a file—the scriptfile—called `tc-blah` (e.g `tc-AFM` for many of the examples below): these local scripts control the way THERMOCALC runs a particular calculation.

scripting in the prefsfile

The main scripts which can be used in the prefsfile are

dataset to specify a suffix on the THERMOCALC dataset file to use in the calculations, e.g. "dataset 62" means that the dataset to use is `tc-ds62`. This is a mandatory script.

scriptfile to specify a THERMOCALC scriptfile, e.g. "scriptfile AFM", meaning that the scriptfile to use is `tc-AFM.txt`. This will cause output files to use `AFM` in their names, e.g. `tc-AFM-ic.txt`. This is a mandatory script.

calcmode to specify a calculation mode: "calcmode 0" for thermodynamic tables; "calcmode 1" for phase equilibria calculations involving solid solutions; "calcmode 2" for avPT calculations; and "calcmode 3" for phase equilibria calculations not involving solid solutions. This is a mandatory script.

setpagewidth to specify the number of characters in a line on screen, relevant to line wrapping or line truncation (via the script, "dontwrap"). For example, for a 200 characters width, this would be "setpagewidth 200". On the Mac, you need to manually adjust the Terminal window width to match that in the "setpagewidth" script—this is not done automatically.

scripting in the scriptfile

The main scripts which can be used in the scriptfile are

axfile to specify a THERMOCALC axfile, e.g. "axfile mp50MnNCKFMASHTO", meaning that the axfile to use in the calculations is `tc-mp50MnNCKFMASHTO.txt`, as appropriate for calculations in AFM. "Axfile" is a mandatory script.

with to specify the phases to use in this calculation, or in a set of calculations. There are two forms of this: (1) for example "with bi chl g st" means that these phases (plus any specified by "inexcess") are to be used in the calculation, and (2) "with *someof* bi chl g st ctd ky" means that these phases (plus any specified by "inexcess") are available for calculations, with a prompt to ask which of the set of phases to use in the specific calculation ("someof" is obviously an additional keyword here). "With", in either form, is a mandatory script.

With "with someof . . .", and doing a series of calculations within one run (so not using "autoexit", see below), varying the assemblage for a calculation can be done as before, using "+" and "-".

Related to "with" are three scripts:

inexcess to specify[3] phases that are to be considered to be in excess in a calculation (i.e. always present). In subsolidus conditions in KFMASH, "inexcess mu q H2O" would be standard, and "inexcess ksp q liq" above the solidus. The "inexcess" phases already can be in the "with" list, or they will be added there automatically from the "inexcess" list

tozero to specify phases—in pseudosection calculations, with modes calculated for a bulk composition or compositions—whose modes are to be normalised out of the output given. For example, for subsolidus KFMASH pseudosection calculations, this is relevant for H_2O as a phase, so you would specify "tozero H2O".

acceptvar to specify that the variance of the assemblage that is given to THERMOCALC by "with", or chosen interactively via "with someof", is the one adopted. The main situation that you would want to use "acceptvar no" is in grid calculations when you give THERMOCALC a set of phases and you want THERMOCALC to calculate all the univariants, say, from all appropriate subsets of the phases.

[3] a script indented in the documentation as here, is a free-standing script on its own line, but it is related to a main script, in this case "with"

pseudosection to specify if the calculations are to relate to a pseudosection, with a bulk composition or compositions provided (for example by "bulk" or "rbi" scripts, or interactively). This script is mandatory.

zeromodeisopleth to specify in pseudosection calculations which phases are to have zero modes, defining effective univariants (lines in PT, with one mode set to zero) and effective invariants (points in PT, with two modes set to zero). With just "zeromodeisopleth", you are prompted for which modes to set to zero. With e.g. "zeromodeisopleth chl", the mode of chlorite is set to zero (assuming the assemblage involves chlorite). With e.g. "zeromodeisopleth chl st", the modes of chlorite and staurolite are set to zero (assuming the assemblage involves chlorite and staurolite).

modeisopleth to specify modeisopleth values, for example for contouring a pseudosection for modes. In a field involving garnet, say, the script for contouring with respect to the mode of garnet can be "modeisopleth g 0.3", meaning garnet mode = 0.3, or "modeisopleth g 0.1 0.3 0.02", meaning garnet mode is successively 0.1, 0.12, ..., 0.28, 0.3.

diagramPT to specify the region of PT that the calculations of interest lie in. This needs to be a rectangular area, for example "diagramPT 1 12 400 800", with the first two numbers being P in kbar, and the second two numbers T in °C. If you are calculating a PT pseudosection or grid, the PT rectangle should be a little bit bigger than the limits of the diagram. In the case of a T–**X** diagram at a fixed P, the PT rectangle needs to be a moderate range around the fixed P. The same equivalently for a P–**X** diagram. The script "diagramPT" is mandatory.

calcTatP to specify whether the calculation of an effective univariant is to be a calculation of T at fixed P ("calcTatP") or a calculation of P at fixed T ("calcT atP no"). Usefully, the script can be used with "calcTatP ask", and you are prompted for which you want. The script has no effect for effective invariants or divariants

calcP to specify the P at which calculations are to be done. There are two forms, the keyword followed by one pressure, e.g. "calcP 8", or the keyword followed by a start P, a finish P and a P increment, e.g. "calcP 5 10 0.5" (meaning successively 5, 5.5, ..., 9.5, 10). P is in kbar.

calcT to specify the T at which calculations are to be done. There are two forms, the keyword followed by one temperature, e.g. "calcT 600", or the keyword followed by

a start T, a finish T and a T increment, e.g. "calcT 500 600 20" (meaning successively 500, 520, ..., 580, 600). T is in °C.

The effect of "calcP" and "calcT" depend on the calculation being done, particularly its effective variance. One or both are ignored if they are inappropriate. The range within which a calculation is expected to lie is given by the PT in "diagramPT"

bulk to specify the bulk composition or compositions to use in pseudosection calculations. This is a multiline script, each line starting with "bulk". The first line gives the oxides involved in the bulk composition, on a molar basis, as in:

```
% ------------------------------------------------------------
  bulk           H2O     SiO2    Al2O3      MgO      FeO      K2O
% ------------------------------------------------------------
  bulk       25.4318  42.2270  11.3240   4.1906  13.7691   3.0574
% ------------------------------------------------------------
```

In the first "bulk" line, the oxides specified must match the oxides that are present in the phases in the calculation, and oxides must be provided in the correct order: H_2O, SiO_2, Al_2O_3, CaO, MgO, FeOt or FeO[4], K_2O, Na_2O, TiO_2, MnO, O, Cr_2O_3. If THERMOCALC doesn't find the list of oxides that it expects at the start of a calculation, it will protest. The second "bulk" line gives the bulk composition in mole%. It's also possible to specify an additional bulk composition, for example in order to calculate a T–$\mathbf{X}$ diagram, where $\mathbf{X}$ varies between the two bulk compositions. Then "bulk" looks like this:

```
% ------------------------------------------------------------
  bulk           H2O     SiO2    Al2O3      MgO      FeO      K2O
% ------------------------------------------------------------
  bulk       25.4888  42.3217  11.3494        1       17   3.0643
  bulk       25.4888  42.3217  11.3494       17        1   3.0643
% ------------------------------------------------------------
```

Optionally the last "bulk" line can have at the end of it a number of x increments between the two bulk compositions, considered as being between $x = 0$ and $x = 1$. In the absence of this part of the script, the number of increments is prompted for.

onebulk to specify one bulk composition between two given bulk compositions, in terms of a value between 0 and 1, giving the position between the two bulks. For example "onebulk 0.2".

[4] FeOt is "all iron as FeO". If the system contains ferric iron, FeOt and O should be specified in the bulk script; implicitly, moles FeO = FeOt - 2 O and moles Fe_2O_3 = O. In an ferric-free system, such as the example, FeO ≡ FeOt.

bulksubrange to specify a subrange within the range of bulk composition betwen two bulk compositions. For T–**X** calculations this allows focussing in on a part of the composition range. For example "bulksubrange 0 0.2". The script "onebulk" has precedence over "bulksubrange".

readbulkinfo (or **rbi**) to specify one bulk composition—alternative to using "bulk"—using the modes of phases and the phase compositions in an assemblage to construct a bulk composition. For "onebulk 0.2" and the above two bulk compositions, THERMOCALC generates the following (using "printbulkinfo"). As usual, the modes, in the first column of numbers, are on a one-element basis. The commented-out line at the end of the information block gives the bulk composition that should be involved:

```
% ---------------------------------------------------------------------------------------------
rbi                          H2O        SiO2       Al2O3        MgO         FeO         K2O
% ---------------------------------------------------------------------------------------------
rbi       g    0.045762          0          3          1    0.267202    2.732798          0
rbi      bi    0.348959   1.000000   2.772251   0.727749    0.764809    2.007442    0.500000
rbi      mu    0.088359          1   3.089130   1.410870    0.037317    0.051813    0.500000
rbi      st    0.157549          2   7.500000          9    0.441832    3.558168          0
rbi     chl           0   4.000000   2.589762   1.410238    1.665529    2.924234          0
rbi       q    0.180773          0          1          0           0           0          0
rbi     H2O    0.178598          1          0          0           0           0          0
% ---------------------------------------------------------------------------------------------
%  bulk                   25.4318    42.2270    11.3240     4.1906     13.7691     3.0574
% ---------------------------------------------------------------------------------------------
```

printbulkinfo to output the information from the current calculation in "rbi" form to make a bulk composition. The output from a calculation, in the form just to copy into the scriptfile, is in the logfile.

dogmin tells THERMOCALC to look at all the equilibria from subsets of the phases given in "with" from some maximum variance down to divariant, and report which has the lowest Gibbs energy[5]. This needs considerable care to be useful, but with care, it is an essential tool. More notes on "dogmin" are now available here on the website. Optionally, "dogmin" can be followed by an integer, 0, 1 or 2. If omitted, 2 is assumed, which is similar to the form of "dogmin" in recent versions of `tc347`.

If they are set, the values in "calcP" and "calcT" are used for the PT that the equilibria are calculated at. The information provided by "dogmin" is more informative if the calculations are done at just one P and T. "Dogmin 1" is the logical way to run this facility; only if everything is very well-defined and organised does it make sense to use "dogmin 0" which produces little output beyond the assemblage with the lowest Gibbs energy.

[5] "Dogmin" = DO Gibbs energy MINimisation

As explained in the dogmin notes, the absolutely critical thing in using "dogmin" is to have an effective set of starting guesses provided to THERMOCALC by the "xyzguess" script. Generally, these have to be worked for, and there are important tactics to adopt, as outlined in the dogmin notes.

maxvar to specify the maximum variance of assemblages that "dogmin" will try and calculate, for example with "maxvar 8". With two numbers provided[6], the first one is the minimum variance, for example "maxvar 4 8". By default the minimum variance is 2. In dogmin, all possible combinations of the phases from the maximum variance down to the minimum variance are examined.

xyzguess to specify starting guesses for the xyz variables (i.e. composition variables) in a calculation. This is a key script—and one that is unchanged from recent THERMOCALCs. It is made of a block of starting guesses for each phase, each block corresponding to the xyz variables in the axfile. If a block for a phase is not present in "xyzguess", the starting guesses in the axfile are used, with no guarantee that the block is relevant or useful for the current calculation.

The usual way of getting "xyzguess" information is from the results of calculations via the "printguessform" script. The "xyzguess" information is in the logfile, ready to be copied across to the scriptfile. Whereas, in principle, calculations can be undertaken without using "xyzguess", it is strongly recommended to copy "xyzguess" blocks from the website into your scriptfile before starting calculations with a new scriptfile.

In fact, two circumstances require manual assembly, even prior to doing calculations in a particular system. The first is when the calculations are to be done in a system smaller than that of the axfile. for example, the axfile might involve Mn end-members of the phases, but the calculations are to be done in the Mn-free system. Then the axfile coding of each of the phases must be simplified (referred to as boiled down) to remove Mn. This is done using "xyzguess", as in the following, allowing THERMOCALC in this case to convert the axfile coding (which is for Mn-Fe^{3+}-Ti-bearing biotite) to that for calculations in KFMASH

```
xyzguess x(bi)        0.5834
xyzguess m(bi)             0  % boiler
xyzguess y(bi)        0.2274
xyzguess f(bi)             0  % boiler
xyzguess t(bi)             0  % boiler
xyzguess Q(bi)        0.1228
```

[6] THERMOCALC alse understands the script keyword, "varrange", a more accurate name for a script with this functionality: "varrange 4 8" has the same affect as "maxvar 4 8"

The second is when the "samecoding" script is used. If several phases have the same structure and therefore have the same axfile coding, only one axfile coding needs to be present in the axfile, the others being made the same using the "samecoding" script. In this case the 'dependent' axfile codings do not have starting guesses in the axfile; these must be provided by "xyzguess".

One way to source "xyzguess" information is to use "dogmin 1" running at one PT, then appraising the contents of the logfile judiciously to collect a set of "xyzguess" blocks for the phases of interest.

printbulkinfo to specify that "xyzguess" information from the calculations undertaken be included in the logfile

samecoding to specify that a set of phases have the same axfile coding. For example "samecoding mu pa" means that paragonite is not in the axfile, its axfile coding being identical to that of muscovite. As noted above, the "xyzguess" information must be provided for paragonite.

projection to specify that the results of calculated equilibria are to be used to calculate the position of the phases in a compatibility diagram, the coordinates of the projection plane being given using the "projplane" script.

projplane to specify the projection plane, for example, for AFM

```
% ------------------------------------------------------------
  projplane              H2O  SiO2 Al2O3   MgO   FeO   K2O
% ------------------------------------------------------------
  projplane     a          0     0     1     0     0     0
  projplane     f          0     0     0     0     1     0
  projplane     m          0     0     0     1     0     0
% ------------------------------------------------------------
```

compatibility to specify that Mathematica-form output for the projection is given in the drfile, and comma-delimited output is given in the csvfile.

isopleth to specify that xyz isopleths are to be calculated. Just "isopleth" means that you are prompted for what to isopleth and the value(s). Specifying these in the script, it can be one value, or start, finish and increment. So an example might be "isopleth x(g) 0.8 x(chl) 0.5"

exbuff to specify that external buffering is being invoked. This means that the calculated phase relations are back-projected onto the H_2O - CO_2 composition line, in the context of $Tx(CO_2)$ diagrams, primarily used to consider carbonate-bearing equilibria. H_2O and CO_2 are not given in "bulk". Some aspects of external buffering calculations

calcx(CO2) specifies the x(CO_2) values for calculations. They can be provided as one or more values, or as start, finish and increment (the latter as for "isopleth"). So for example, the script could be "calcx(CO2) 0.001 0.002 0.005 0.01 0.02 0.05 0.1 0.2 0.5" or "calcx(CO2) 0.1 0.9 0.05", meaning successively 0.1, 0.15, ... 0.85, 0.9

modebox to specify that modebox information is to be calculated and given in the drfile and csvfile

drawpd to specify that calculation results are given in the drfile in a form to copy across into a drawpd datafile for plotting (see the drawpd notes).

autoexit to specify that THERMOCALC terminates the run when the current calculation is done, not prompting to ask if more calculations are wanted. This allows fully hands-off running of THERMOCALC.

also other scripts that did work, may still work, but haven't been checked out yet. This is a work in progress.

rp/ecrg 5·12·20