Looping the loop: an introduction to PhreePlot's looping

The principal aim of PhreePlot is to use Phreeqc to prepare data for making 1-D (line) and 2-D (colour fill) plots so the x- and y- axis loops are the two basic loops. On top of these is the z-loop (or simply 'loop'), which is designed to prepare data for a sequence of 1- or 2-D plots, and the mainspecies loop which is similar to the z-loop but loops on a list of character strings, often element names.

Phreeqc input files can contain many distinct 'simulations' (code separated by `ENDs`) which may or may not be related to each other. These are all normally run just once in sequence when run e.g. with Phreeqc Interactive. PhreePlot, which includes the iPhreeqc library, introduces a simple way to loop over these simulations. In order to preserve the Phreeqc input structure intact, specifying the looping is done remotely using various keywords and parameters. Variation is produced by the inclusion of tags which are substituted during the looping.

PhreePlot acts as a kind of 'server' for running Phreeqc code. This introduces an additional option - are the simulations all run in one block (like Phreeqc Interactive) or run 'one simulation at a time' returning to PhreePlot after each simulation has been run? In the latter case there is the additional possibility of PhreePlot acting on the intermediate results and thereby influencing later simulations. However, this interaction comes with a speed penalty.

PhreePlot makes a distinction between 'pre-loop' simulations and 'main loop' simulations.

Pre-loop simulations always precede main loop simulations in an input file. The cut-off between the two is set by the `'mainLoop'` keyword. `mainLoop` is the number of the first main loop simulation counting downwards. If there are `n` simulations, the default (=`'auto'`) is for `mainLoop` to be set to `n`, i.e. just the last simulation will be main loop and all the preceding ones will be pre-loop.

Pre-loop simulations are normally only run once for each complete set of x (and y) iterations and are executed 'one at a time' (speed is not an issue here). They are also rerun every time the z- or mainspecies loops are incremented. Main loop simulations are run on every iteration and normally in one block for maximum speed. This division between pre-loop and main loop means that various initializations, like modifying the database, calculating initial solutions etc, can be run just once in the pre-loop simulations saving time by not repeating them many times. Assuming we have

```
# Simulation 1
...
END
# Simulation 2
...
END
# Simulation 3
...
END
```

this is how the above three simulations will be executed with various settings of `mainLoop`

`mainLoop = 3` (the default for three simulations)
`1 then 2 then loop on 3`
`mainLoop = 2`
`1 then loop on 2-3`

```
mainLoop = 1
(then) loop on 1-3
```

Passing results between simulations

PhreePlot input (`*.ppi`) files are split into an upper and lower section separated by the '`Chemistry`' keyword. The upper section is the PhreePlot section that defines the looping and how the calculations are to be done, details about various input and output options, and various plotting options. The lower section is the Phreeqc section that contains normal Phreeqc code with optional PhreePlot 'tags' included. These tags (`<...>`) are placeholders and define the positions of values to be substituted before being executed. The values are set in a various ways: in the PhreePlot section of an input file (`numericTags` and `characterTags`), automatically from various input and output files, by the 'system' after each iteration, and by the four loop variables and their associated tags, namely `<x_axis>`, `<y_axis>`, `<loop>` and `<mainspecies>`. The first 3 of these are numeric; the last is character.

Tags can only be updated when control is sent back to PhreePlot (i.e. after a 'then' above, although this can be overridden with the `oneAtaTime` switch, the optional second parameter of the `mainLoop` keyword). Data are sent to PhreePlot via the `SELECTED_OUTPUT/USER_PUNCH` mechanism. On returning from Phreeqc, tags are automatically created based on the selected output and the column header. These tags and their associated values can then be re-used in subsequent simulations whenever control is returned to PhreePlot. It is therefore possible to use the results from one simulation in later ones. You can sometimes also use Phreeqc's `PUT/GET` mechanism to do much the same. Often it is only the last row of selected output that is associated with these tags, but it is possible to read in a whole block of selected output, for example, during fitting or when output is generated by one of Phreeqc's own internal looping mechanisms, e.g. REACTION.

These tags have global scope and can, with care, also be used to make substitutions in the upper or PhreePlot section of an input file. This section sets various plotting parameters and so, for example, a plot can react dynamically to the results calculated in the chemistry section.

Looping

x-axis and y-axis variables are used to drive the looping required to prepare 2D-type plots namely predominance plots and contour plots. The x-axis variable used by itself is designed for 1D plots such as species plots and custom plots in which the x_axis is the 'driving' variable and the y_axis is a variable calculated by Phreeqc. This produces curves represented by lines or sets of points.

The 'loop' variable provides a means of looping over various instances of the x- and y-axis variables, each time making a new curve or a new plot. For example, it can be used for making a series of predominance plots with a range of total concentrations for the mainspecies. This looping can be over a regular sequence of values or an irregular sequence read from a loop file. The loop file is read line by line and can contain other tagged variables that are also stepped in synchrony. Variable names are defined in a header line and these are automatically used as tag names.

The fourth loop is the mainspecies loop which simply loops through a list of characters strings defined by the mainspecies tag. This tag is often associated with a list of chemical elements, hence the name, but it can be any list of character strings that need substituting one after the

other. Each iteration of the mainspecies loop will normally make a new file and a new plot.

An important point is this: if a simulation contains a tag that must change during the x- and y-looping, it must be in the main loop section (not pre-loop) otherwise it will not change.

The overall structure of a PhreePlot input file is:

```
pre-loop simulations
| Chemistry
main loop simulations
```

and these will normally iterate as below

```
------→ pre-loop iterations, one simulation at a time
------→ main loop iterations, all simulations in one block
mainspecies          <mainspecies> set by mainSpecies list
|  loop              <loop> set using loopmin, loopmax and loopint, or by a loop file
   ------→  repeat pre-loop simulations here
|  |  y_axis         <y_axis> set by ymin, ymax and res (not used for 1D plots)
|  |  |  x_axis      <x_axis> set by xmin, xmax and res
|  |  |  |
|  |  |  | Phreeqc code here
|  |  |  |
|  |  |  end x_axis
|  |  end y_axis
|  end loop
end mainspecies
```

With '`species`' distribution plots, the mainspecies and loop positions in the above structure are switched to minimize the number of calculations.

Each run of PhreePlot is designed to produce a single plot, or series of related plots. If two or more unrelated plots are required, or data has to be pre-calculated for inclusion in a later plot, then Windows batch file processing should be used to sequence the execution of the various input files.

Preparing data for plotting

It is assumed that there are no extraneous simulations so the last line (by default), or lines, of selected output sent are assumed to be the one(s) needed. For other cases, there is a mechanism for telling PhreePlot how many lines are actually being sent from each simulation. The inner, x-loop writes data to an 'out' file but typically each time an outer-loop increments, either a blank line is written or a new file is created. Blank lines are interpreted as a break in the data and will normally cause a new line to be plotted. For predominance and contour plots, looping will always produce a new plot.

Predominance Plots

The '`grid`' and '`ht1`' methods are used to make predominance plots, the former by sampling on a regular grid, the latter by a method that interactively tracks the field boundaries (assuming that they can be accessed from the domain boundaries). Both methods send requests to the Phreeqc

server to return the speciation for given x- and y- values. PhreePlot then ranks the species using user-supplied code such as `ht1.inc`. This can be easily edited to modify the ranking criterion. The result is returned to PhreePlot via the selected output and used to generate a diagram.

The mainspecies tag is used to define the 'main species' (or element) on which the plot is based.

These diagrams are somewhat different from classical predominance plots in that they are based on 'real' chemistry (per Phreeqc) and, for example, make no assumptions about the activities of any of the species being fixed. They can also include more complex species such as adsorbed species and solid solutions. However, they are more challenging to make, slower to calculate and sometimes require a fair degree of familiarity with Phreeqc to complete successfully. The 'ht1' tracking can fail at certain resolutions where the intersections are complex. The 'grid' method should always produce a plot, even where Phreeqc has failed.

It is often convenient to split predominance calculations into two simulations - a pre-loop simulation to do the initial solution calculations and then the main loop simulation to do the variable calculations driven by the `<x_axis>` and `<y_axis>` tags, i.e. two simulations with `mainLoop = 2`. This is the default for `mainLoop` set by '`auto`' as read from `pp.set`. The initial solution is only calculated once and is repeatedly re-used as the starting point for the reaction calculations. The challenge is usually to engineer Phreeqc code to get from this starting point to all points in the domain following a feasible reaction path based on a fixed set of main loop simulations.

Contour plots

It is possible to make 'contour' or level plots of any parameter that Phreeqc can calculate. The calculations for this are driven by the x- and y-axis tags. The z-variable, defined by the contourzVariable, is then output to the selected output for contouring.

Fitting is special

Fitting is somewhat different as each data point potentially has its own block of simulations associated with it, each with its own set of pre- and main loop simulations. Observations are read from a 'fit data file', and compared with the output calculated by Phreeqc for each data point. The adjustable parameters are then automatically adjusted until a best fit has been found.

Simulation mode is similar except that no adjustment of parameters is made – it just uses the parameter values as given and is therefore run just once.

Fitting does not utilize the 4-level looping outlined above since it has its own special type of looping. This takes place over the data given in the fit data file. In addition to this, it is also possible to loop over the entire fitting procedure by fitting the same set of data with the same set of parameters but using different optimizing routines (currently five). This is done by specifying a list of the optimizing procedures to use with the '`fitMethod`' keyword.

David Kinniburgh

February 2016