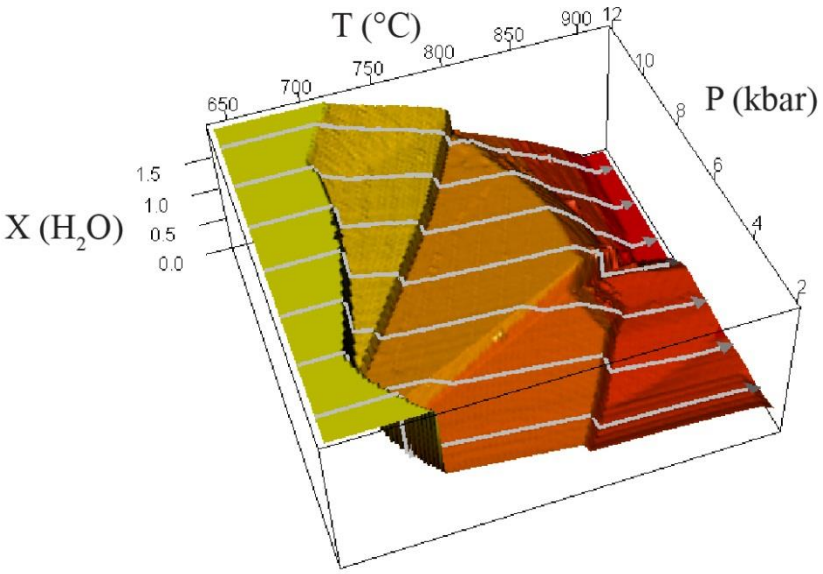
Rcrust

Phase stabilities with path-dependence

MJ Mayne,JF Moyen 14 February 2017 (for help contact mjmayne@outlook.com)



Getting started

Reference manual

GETTING STARTED

Installation

Rcrust was developed using version 3.3.0 (2016-05-03) of R. Copyright © 2016 the R Foundation for Statistical Computing. To install Rcrust perform the following steps:

1. Copy the Rcrust folder to a location of your choice (preferably a root directory for example C:\ or D:\). The result should be similar to the picture below with all the Rcrust files contained in single directory for example D:\\Rcrust\

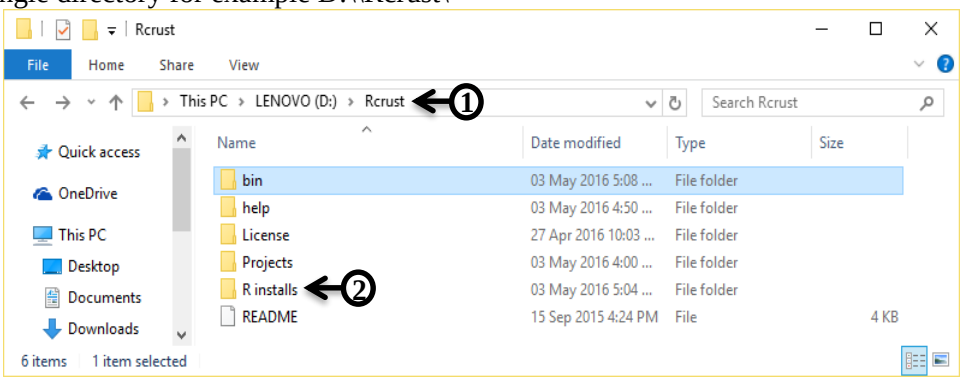


Figure 1 - Rcrust file structure located in the root directory D:\\Rcrust

2. Install a working version of R on your system (at least version 3.3.0). The latest version of R used in the development of Rcrust is located in the folder "R installs" for your convenience. **Warning:** Rcrust requires the **64 bit** version of R. When installing R please ensure "64-bit Files" is ticked.

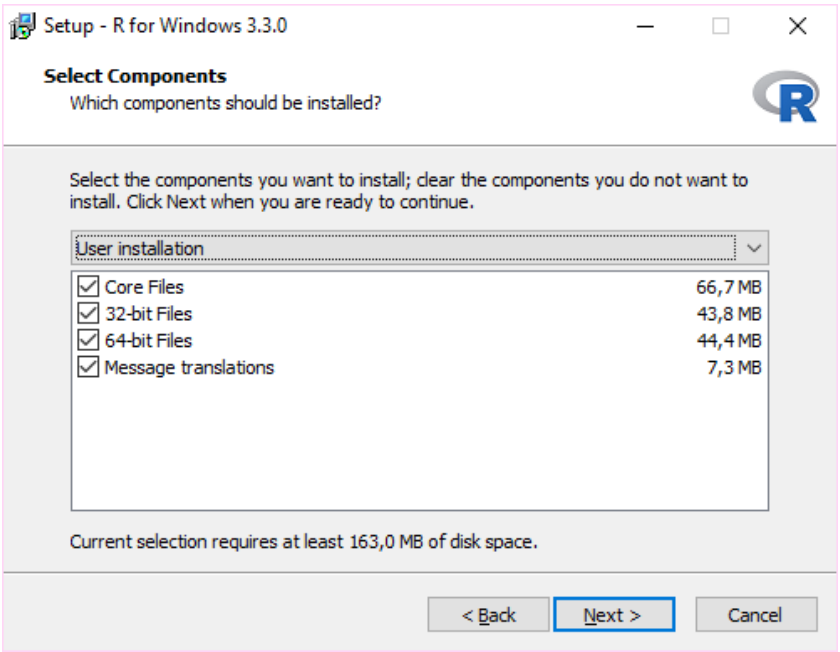


Figure 2 - R installation instruction ensuring at least "64-bit Files" is ticked

*Alternatively newer versions of R (which may not be compatible with Rcrust) can be downloaded from http://www.r-project.org/ or for windows can be found directly at http://cran.r-project.org/bin/windows/base/

3. Open the folder called bin in the Rcrust folder.

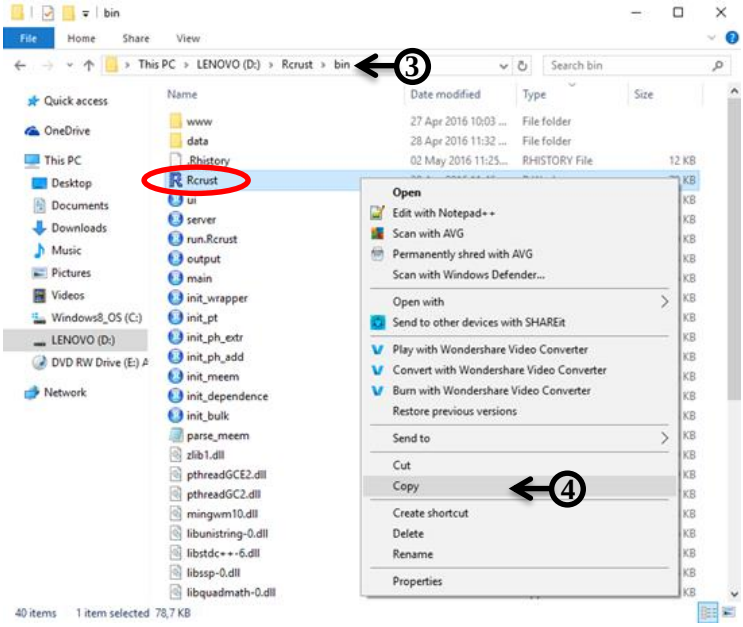


Figure 3 – Rcrust file (circled in red) within bin folder. The Rcrust file location here is D:\\Rcrust\bin\Rcrust.RData

4. Copy the Rcrust file found in the bin folder (~/Rcrust/bin/Rcrust.RData). This can be done by right clicking on the file (circled in red above) and selecting "Copy" or by selecting the file and pressing "Ctrl"+"c".
5. Paste the Rcrust file as a **shortcut** on the Desktop. This can be done by right clicking on the Desktop and selecting "**Paste shortcut**".

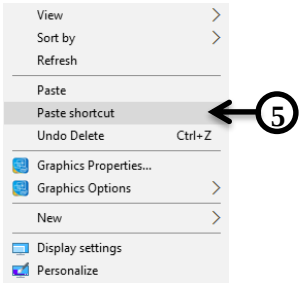


Figure 4 – Paste shortcut option selected for Rcrust file

6. Rename this file to "Rcrust". Double click on this shortcut to open Rcrust.

7. Rcrust requires the R package called "shiny", If this is not installed on your computer when you open Rcrust, Rcrust will try to install it (this requires an internet connection). Follow the prompts to complete installation of the package (it may ask you if you wish to create a personal library – choose yes). Alternatively, shiny can be downloaded here: http://shiny.rstudio.com/ .

Each new project will be automatically saved in the "Projects" folder along with its associated inputs and outputs. To load a previously saved project simply double click the "xxx.RData" file in the associated project folder or open Rcrust from the desktop shortcut and load the project via the Rcrust GUI.

Concept

Rcrust is an R program aimed at modelling with path dependence. The program functions by calculating a number of points in P-T-X space where a bulk composition is passed between points. This creates path dependence as points within the path rely on the outcomes of previous points for their calculation. The bulk composition can be altered at each point by phase manipulations consisting of phase additions and/or phase extractions.

Rcrust manages calculations by splitting the full thermodynamic system (FS) into 3 subsystems: The reactive subsystem (RS) which contains the phases in thermodynamic equilibrium; The addition subsystem (AS) where phases are waiting to be added to the reactive subsystem; and the extract subsystem (ES) where phases extracted from the reactive subsystem are stored. The reactive subsystem is in thermodynamic equilibrium with the P-T-X conditions of each point and is re-equilibrated after each P-T-X change. The addition and extract subsystems are in thermodynamic isolation from other subsystems and from the P-T-X conditions of each point.

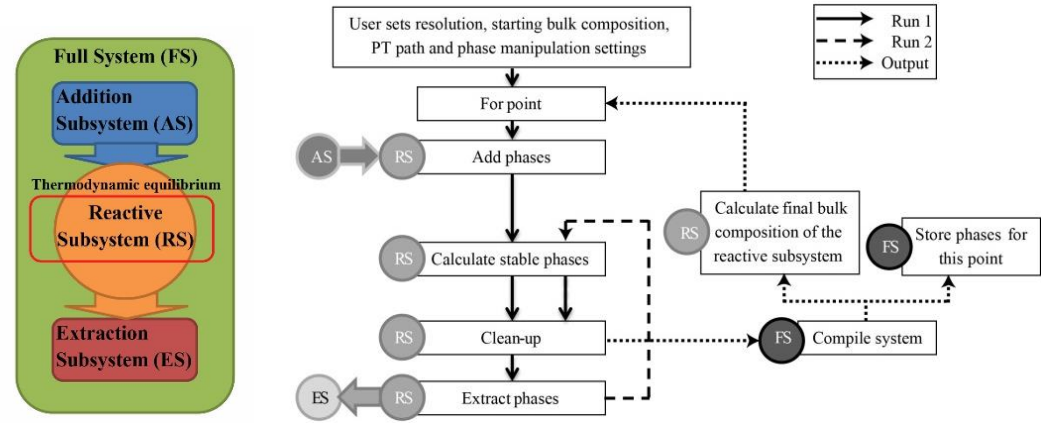


Figure 5 - Relationships between systems (left) and flow chart (right) illustrating the Rcrust program structure for a single path. The user inputs the calculation's resolution, starting bulk composition, P-T path and phase manipulation settings. Each step in a path consists of two runs and an output. The first run is shown in a solid line, the second run in a dashed line and the outputs in a dotted line. Circles show the system or subsystem involved in each step as AS (addition subsystem), ES (extract subsystem), FS (full system) or RS (reactive subsystem). Arrows show interactions between systems. From (Mayne et al. 2016)

Parameters for calculations are accessible to the user via the Rcrust Graphical User Interface (GUI). This GUI writes data to a text file which is then input to the program thus allowing the user to edit the file 'behind' the GUI as well as save inputs for re-use. The code files are extensively commented, and described in this document. The calculations routines are defined in several files, written in a modular way that should allow easy addition of features if required. For example, the Phase Extraction routine has been modified to suit the needs of magma extraction where additional capabilities allow melt extraction to leave a set melt retention amount behind.

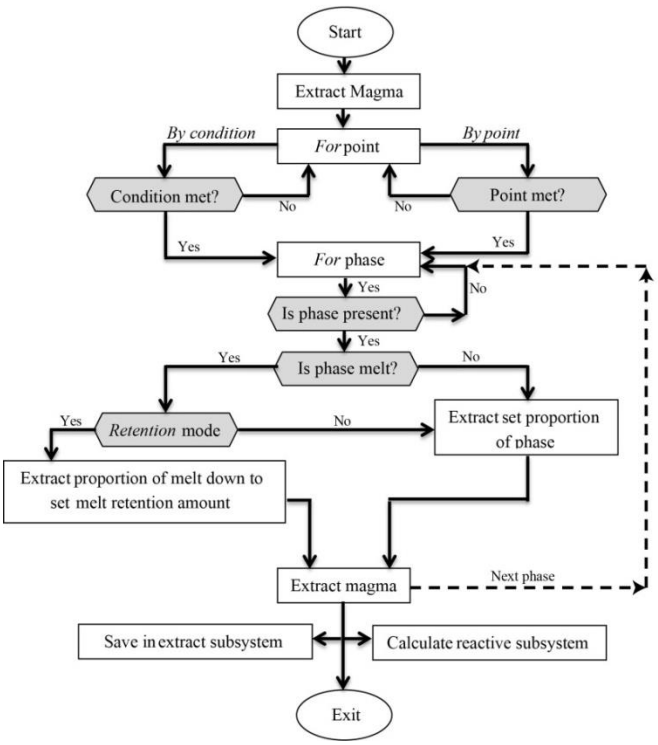


Figure 6 - Flow chart of the magma extraction routine. Grey hexagon shaped boxes are decision points. Coding variables are in italics. The for phase loop (dotted line) is repeated until each phase tagged for extraction has been considered. If Retention mode is active melt is considered last so that other phases extracted are accounted for in its calculation. From (Mayne et al. 2016)

Rcrust results should easily be loaded into GCDkit and examined from there.

It is important to remember a few things:

- Rcrust is in development. It is not mature software. It is very unstable at the best of times, and very unforgiving in terms of improperly formatted inputs, etc. When Rcrust fails, it will try to generate some human-readable error messages: read them! It may well give you hints at things you can correct in your inputs.
- Most of the errors you will see are related to incorrect input (files with incorrect number of lines etc.); or to exotic phases being produced by meemum.

Rcrust calls a set of binary files containing the thermodynamic equations thus relying on published databases (Holland and Powell typically). The output will never be better than the underlying thermodynamic model. Since we focus on melting, we are tied to the capacities (and limitations) of the melt models. For example, melt(HP) does not include Titanium therefore its use in a system containing Ti can over/under estimate melt abundance.

Examples

Below are 3 example simulations to get you started using Rcrust. All you need to do to complete the examples is to perform the actions written in bold numbered text. Explanations of what these actions achieve are given between steps.

Example1 – Simple

Follow the bold numbered steps

To begin the first example open Rcrust via the desktop shortcut.

1. **Double click the Rcrust desktop shortcut**

This will launch the R console and an empty Rcrust Graphical User Interface (GUI) in your default web browser. The "**Working File**" (circled in red) shows you which file is currently being worked on and the "**Projects Directory**" (circled in green) shows you where the projects folder is located. The Rcrust toolbar (in grey) contains buttons for file management.

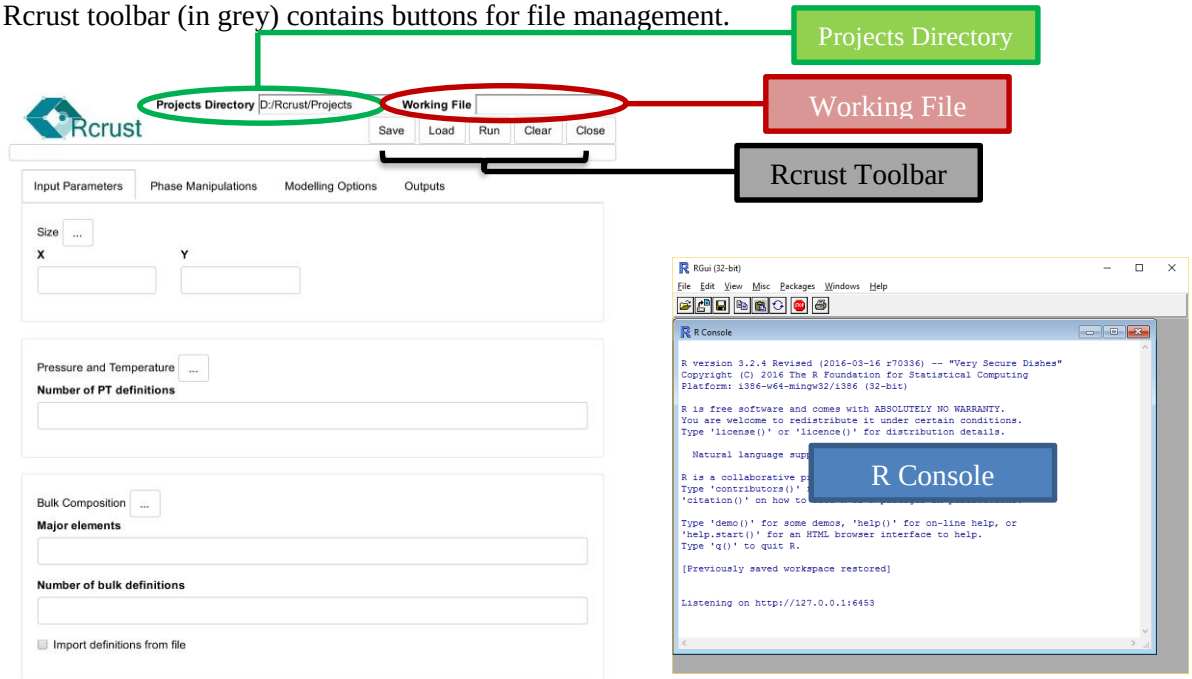


Figure 7 -Rcrust GUI and R Console (blue). Highlighted are the positions of the Projects Directory (green), Working File (red) and Rcrust toolbar (black).

2. **Type "Example1" into the text box on the right of Working File and then click the Load button from the Rcrust toolbar**

The data previously saved in the "Example1" file is now loaded into R and previously saved input parameters are loaded into the Rcrust GUI. To ensure that we do not overwrite any data lets rename the Working File.

3. **Rename Example1 by typing "Example_simple" into the Working File textbox then click the Save button from the Rcrust toolbar**

This will save the current Rcrust GUI inputs into a new file named "Example_simple". The Rcrust GUI should now look similar to Figure 8.

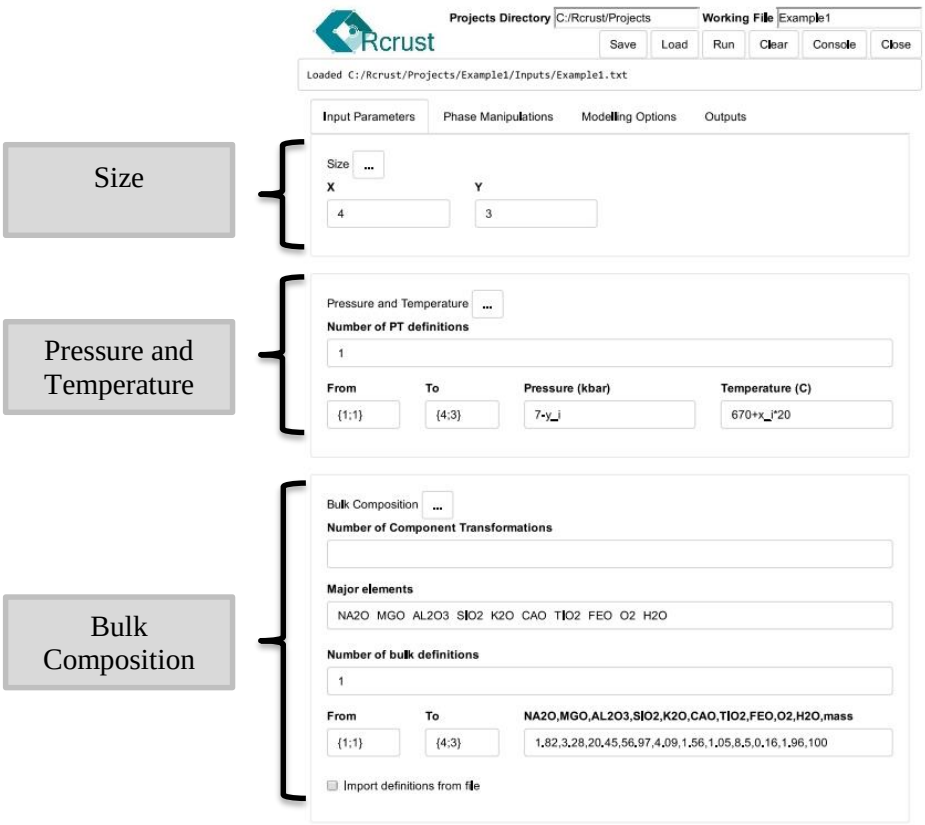


Figure 8 - Rcrust GUI with Example1 (Example_simple) parameters loaded. The GUI consists of a number of tabs. The Input Parameters tab sets the size, P-T conditions and bulk composition (X) of the simulation.

This example calculates the phases encountered at points in P-T-X space. Input parameters are grouped into collapsible panels:

- Size Panel

The **Size** panel sets the number of points in the simulation (here 4 points in the **X** direction multiplied by 3 in the **Y** direction). Points in the simulation space are identified by **tuples** written as {x_i ; y_i} where i denotes the current point.

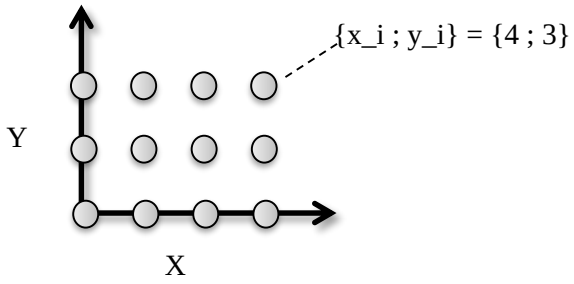


Figure 9 – Simulation space with point coordinates defined by tuples {x_i ; y_i}

- Pressure and Temperature Panel

Parameters in the simulation space are filled by a number of definitions. Each unique definition is applied over a range between the tuples **From** { x_a ; y_a } and **To** { x_b ; y_b } where a and b denote the start and end points of a rectangular range. Each definition sets its attributes as constants or as functions of the point's position. **X** and **Y** positions of points are accessed by the variables x_i and y_i respectively. In the **Pressure and Temperature Panel** we set the Pressure in kilobars and

Temperature in degrees Celsius for each point. In this example **Pressure** decreases along the y-axis by 1 kbar per point (7-y_i) and **Temperature** increases along the x-axis by 20 °C per point (670+x_i*20).

- Bulk Composition Panel

The **bulk composition** of the system is made up of a number of major elements expressed as wt.% elemental oxides. For this simple example we define the bulk composition to be constant across the full P-T space, and we give it the value of the average amphibolite-facies pelite composition considered in (Mayne et al. 2016).

	Na_2O	MgO	Al_2O_3	SiO_2	K_2O	CaO	TiO_2	FeO	O_2	H_2O
wt.%	1.82	3.28	20.45	56.97	4.09	1.56	1.05	8.5	0.16	1.96

Table 1 – Average amphibolite facies pelite composition considered in (Mayne et al. 2016).

To get started let's run a reconnaissance simulation:

4. **Click the Run button from the Rcrust toolbar**

This will save inputs in the Rcrust GUI and launch the calculation procedure into the R console. Navigate to the R console now. The R console should now have a few lines of text in it (like the figure below): If your simulation successfully initialized like the one below then we are ready to start the calculation. If your console failed to initialize the program try reloading the original "Example1" file by closing Rcrust then starting from step 1 again, if problems persist try reinstalling Rcrust or report the problem to the developers (mjmayne@outlook.com).

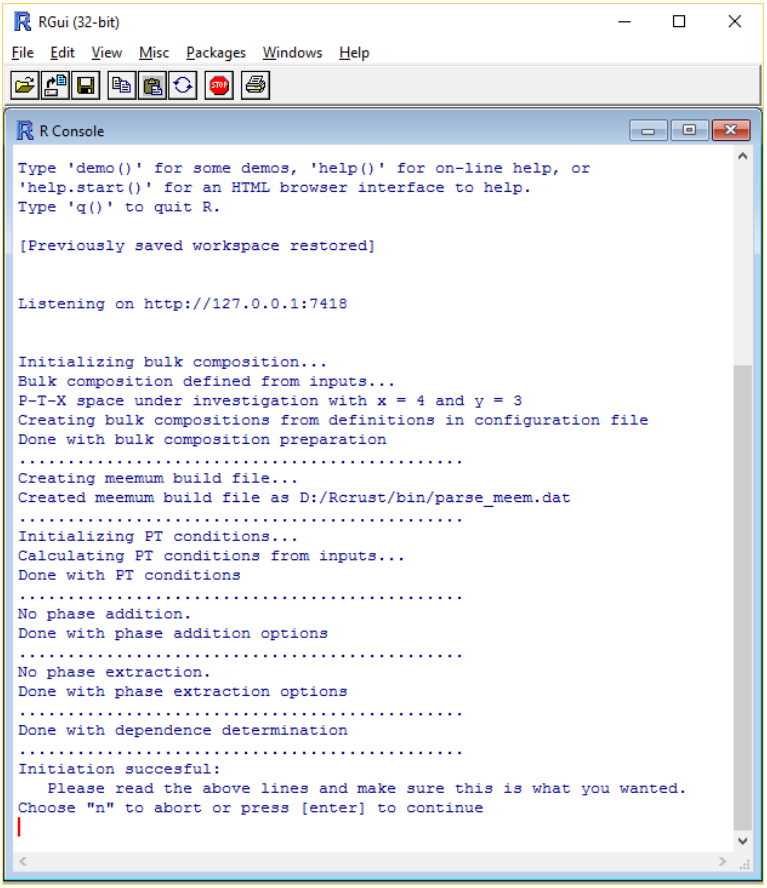


Figure 10 - The Rcrust calculation is launched into the R Console which tracks the calculation progress and is currently waiting for a response to continue or to abort.

5. **Click anywhere in the R console to activate it then press [enter] to continue**

The calculation will run for 12 points :4 in the X direction by 3 in the Y direction. The results will automatically be saved to file and you will be prompted to select outputs.

6. **Once the calculation is complete, navigate back to the Rcrust GUI and Select the Outputs tab**

Here we see a compilation data file for the points in our simulation. To compare points, click the "Select Output" drop down and choose "Grid".

7. **Choose "Grid" under "Select Output"**

Grid allows easy comparisons between points for example choose:

- Select Output = Grid
  - Variable = wt% and Melt_rs

8. **Choose Variable="wt%" and "Melt_rs"**

The output should now match Figure 11. This shows us the amount of melt in the Reactive Subsystem (RS) over our point selection in P-T-X space. Remember we set temperature to increase along the x-axis and pressure to decrease along the y-axis. With that in mind, we see that the solidus (boundary between liquid and solid) at lower pressures is crossed by lower temperatures (this is a good illustration of the pressure dependence of melting).

For further outputs you can deal directly with the data in the R console (hint: you can plot data directly into GCDkit). To access the data in R console, click the "Console" button on the Rcrust toolbar to launch a browser access. To return to the Rgui at any point type "c" then press [enter].

Melt_rs wt% on (X,Y) grid

	V1	V2	V3	V4
1	0.00	0.00	12.59	14.91
2	0.00	8.74	10.21	11.26
3	0.00	8.20	9.07	10.32

Figure 11 – Grid output Data for Example_simple showing weight percentage of melt in the reactive subsystem for the P-T-X points selected

To view a graphical output of this data toggle the "View" selection to "Plot". This will plot a filled contour graph of the selected data which can be saved directly as a .ps file through the "Save To File" button at the bottom of the selection panel.

9. **Toggle the "View" selection from "Data" to "Plot" and choose "Bottom Axis" as "Temperature", "Left Axis" as "Pressure"**

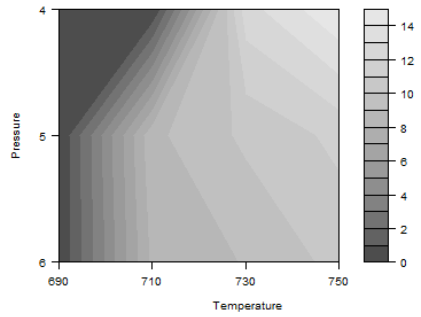


Figure 12 – Grid output Plot for Example_simple showing weight percentage of melt in the reactive subsystem for the P-T-X points selected as a filled contour plot

Example2 – Phase extraction

Phase extractions can remove phases from the reactive subsystem. This is used to simulate scenarios such as melt loss or fractional crystallization.

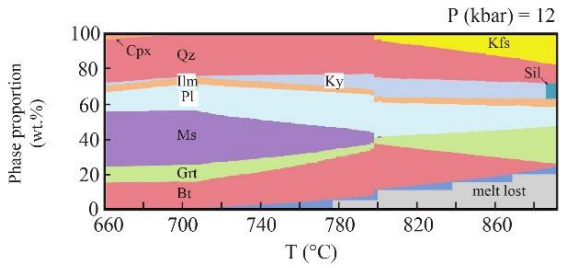


Figure 13 – Example of melt extraction along a P-T-X path from (Mayne et al. 2016)**: Weight percentage of phases verses temperature in degrees Celsius for a fixed pressure of 12 kbar. Starting composition taken as an average amphibolite facies pelite (Table 1). Melt is extracted whenever a 7 vol.% threshold is met.**

Let's perform melt extraction along a path in the P-T-X space explored by Example1.

1. **Load Example2 by opening Rcrust, typing 'Example2' in working file and clicking "Load"**

To ensure that we do not overwrite any data lets rename the Working File.

2. **Rename the file by typing "Example_extract" into Working File then click the Save button from the Rcrust toolbar**

This will save the current Rcrust GUI inputs into a new file named "Example_extract". The Rcrust GUI should now look like the images below:

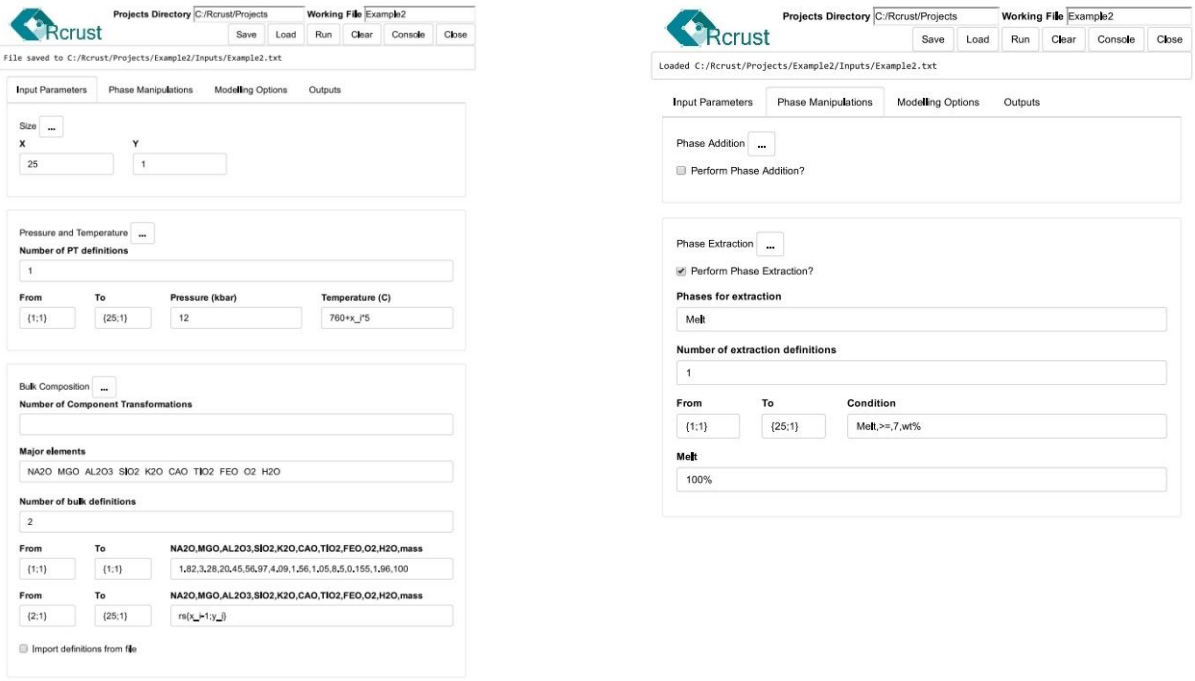


Figure 14 - Rcrust GUI inputs for Example_extract

This example calculates an open system dependent path in P-T-X space. The bulk composition of the Reactive Subsystem is altered by phase manipulations encountered along the path. Phase Manipulations are grouped into collapsible panels:

- Phase Addition

Phases such as intruding fluids, segregated melts or residual crystals can be incorporated into the reactive subsystem. These additions are defined by the major elemental oxides chosen in “Bulk composition”.

- Phase Extraction

Phases in the reactive subsystem can be extracted when set conditions are met. These conditions are defined as logical arguments such as “TRUE”/”FALSE” to extract for every point/no points respectively. Alternatively a logical argument can be built of the form “phase,operand,value,unit” where phase = name of the phase/solution model, operand = (<,<=,==,>=,>,!=), value = a number and unit = the phase property to test. In this example we want to trigger extraction whenever a melt threshold is reached so our conditional argument is “Melt,>=,7,wt%”. *note that each argument in the condition is separated by a comma.

When the condition is met phase extraction is triggered on the reactive subsystem. For each phase listed in **Phases for Extraction** we need to define the amount of the phase to extract. This can be a numeric value (interpreted as grams relative to the starting mass in grams defined in the Bulk Composition Panel), or a percentage of the current value. In this example we extract all melt (100%).

Figure 15 - Phase extraction definition for Example_extract

To save you time we have pre-run this calculation so you can directly view the results by selecting the **Outputs** tab.

3. Select the Outputs tab in the Rcrust GUI

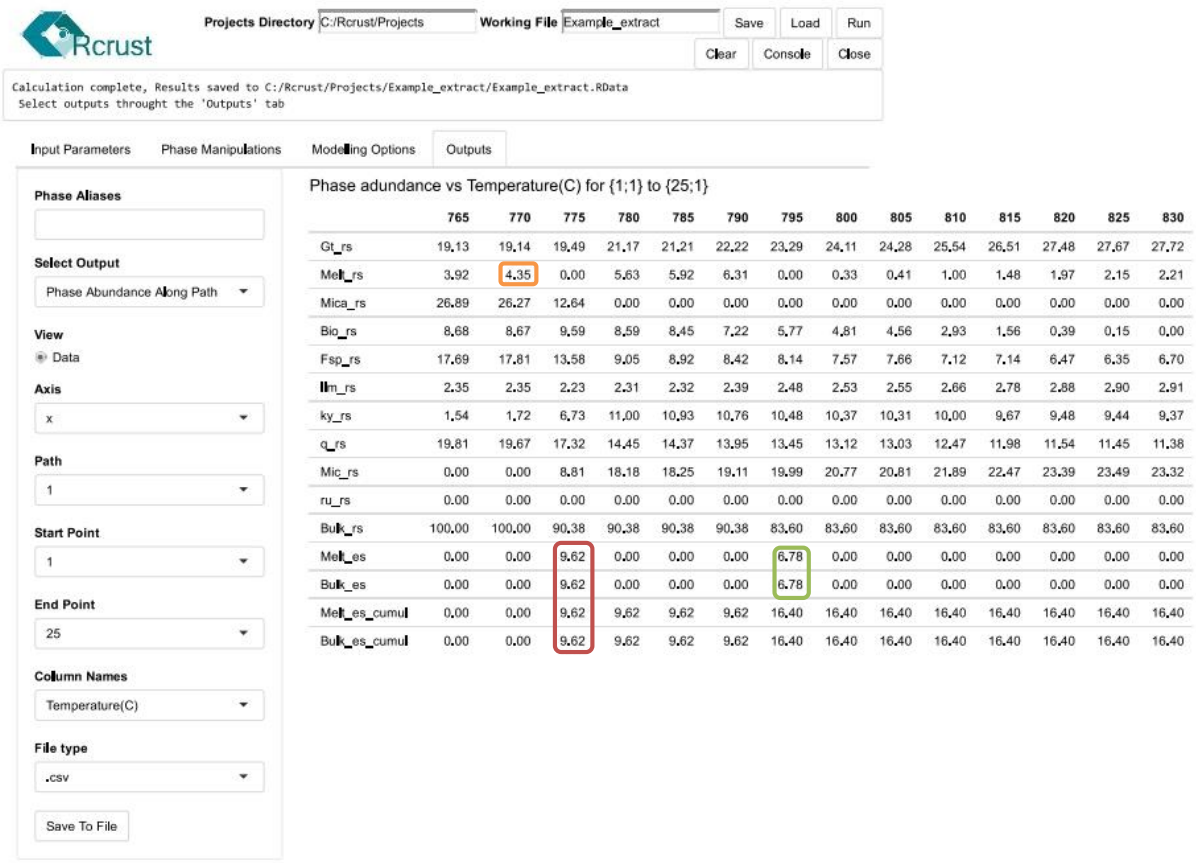
A custom output selection is available for viewing phase abundances along a path.

4. Choose “Phase Abundance Along Path” under “Select Output”

Here you can select which axis the path traverses (axis), which path you wish to consider (path), select the (Start Point) and (End Point) of the path and add a label for the column names.

5. Set “End Point” as 25

The output should now match Figure 16.



	765	770	775	780	785	790	795	800	805	810	815	820	825	830
Gt_rs	19.13	19.14	19.49	21.17	21.21	22.22	23.29	24.11	24.28	25.54	26.51	27.48	27.67	27.72
Melt_rs	3.92	4.35	0.00	5.63	5.92	6.31	0.00	0.33	0.41	1.00	1.48	1.97	2.15	2.21
Mica_rs	26.89	26.27	12.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bio_rs	8.68	8.67	9.59	8.59	8.45	7.22	5.77	4.81	4.56	2.93	1.56	0.39	0.15	0.00
Fsp_rs	17.69	17.81	13.58	9.05	8.92	8.42	8.14	7.57	7.66	7.12	7.14	6.47	6.35	6.70
Ilm_rs	2.35	2.35	2.23	2.31	2.32	2.39	2.48	2.53	2.55	2.66	2.78	2.88	2.90	2.91
ky_rs	1.54	1.72	6.73	11.00	10.93	10.76	10.48	10.37	10.31	10.00	9.67	9.48	9.44	9.37
q_rs	19.81	19.67	17.32	14.45	14.37	13.95	13.45	13.12	13.03	12.47	11.98	11.54	11.45	11.38
Mic_rs	0.00	0.00	8.81	18.18	18.25	19.11	19.99	20.77	20.81	21.89	22.47	23.39	23.49	23.32
ru_rs	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bulk_rs	100.00	100.00	90.38	90.38	90.38	90.38	83.60	83.60	83.60	83.60	83.60	83.60	83.60	83.60
Melt_es	0.00	0.00	9.62	0.00	0.00	0.00	6.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bulk_es	0.00	0.00	9.62	0.00	0.00	0.00	6.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Melt_es_cumul	0.00	0.00	9.62	9.62	9.62	9.62	16.40	16.40	16.40	16.40	16.40	16.40	16.40	16.40
Bulk_es_cumul	0.00	0.00	9.62	9.62	9.62	9.62	16.40	16.40	16.40	16.40	16.40	16.40	16.40	16.40

Figure 16 – Phase Abundance Along Path output for Example_extract showing mass of each phase in the full system (FS) across the P-T-X points selected.

This output highlights a few key features of the phase extraction function:

- The first melt extraction event extracts more than 7 wt% melt (relative to the full system)(red box) even though the melt extraction threshold was set to be 7 wt%

Reason -> Evaluations are only performed at each point thus if the resolution (number of points) is low then large changes can occur between each point. (the previous point (orange box) had less than 7 wt% melt so did not exceed the melt extraction threshold.

- Subsequent melt extraction events may appear less than the melt extraction threshold (green boxes)

Reason -> The melt extraction threshold is evaluated relative to the Reactive Subsystem (which itself is shrinking due to melt extraction events) thus equivalent proportions of melt equate to different proportions when compared to the full system (FS).

Further points to note about phase extraction are:

- Extracts are not thermodynamically re-equilibrated to the reactive subsystem or the *P-T* conditions and can be compilations of multiple phases in solid solution.
- The reactive subsystem is re-equilibrated after each extract so may produce new instances of a phase that were extracted.

Example3 - Multi-path functionality

Multiple paths can be compiled in P-T-X space to produce path-dependent P-T mode diagrams. In these diagrams a plane in P-T space is filled with points originating from dependent paths.

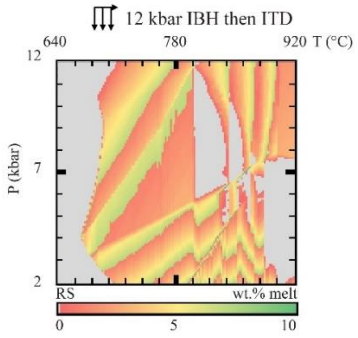


Figure 17 – Example of a composite path-dependent P-T mode diagram from (Mayne et al. 2016): Colours scale the weight percentage of melt in the reactive subsystem (RS). Starting composition at 640 °C and 12 kbar taken as an average amphibolite facies pelite (Table 1). Melt is extracted whenever a 7 wt.% threshold is met and leaves behind 1 wt.% approximating melt retention on grain boundaries. The simulation space is filled by a number of isothermal decompression paths that each originate off a 12 kbar isobaric heating path.

Let's create a path-dependent P-T mode diagram by decompressing off of the path investigated in Example2.

1. **Load Example3 by opening Rcrust, typing 'Example3' in working file and clicking "Load"**

To ensure that we do not overwrite any data lets rename the Working File.

2. **Rename the file by typing "Example_multi" into Working File then click the Save button from the Rcrust toolbar**

This will save the current Rcrust GUI inputs into a new file named "Example_multi". The Rcrust GUI should now look like Figure 18.

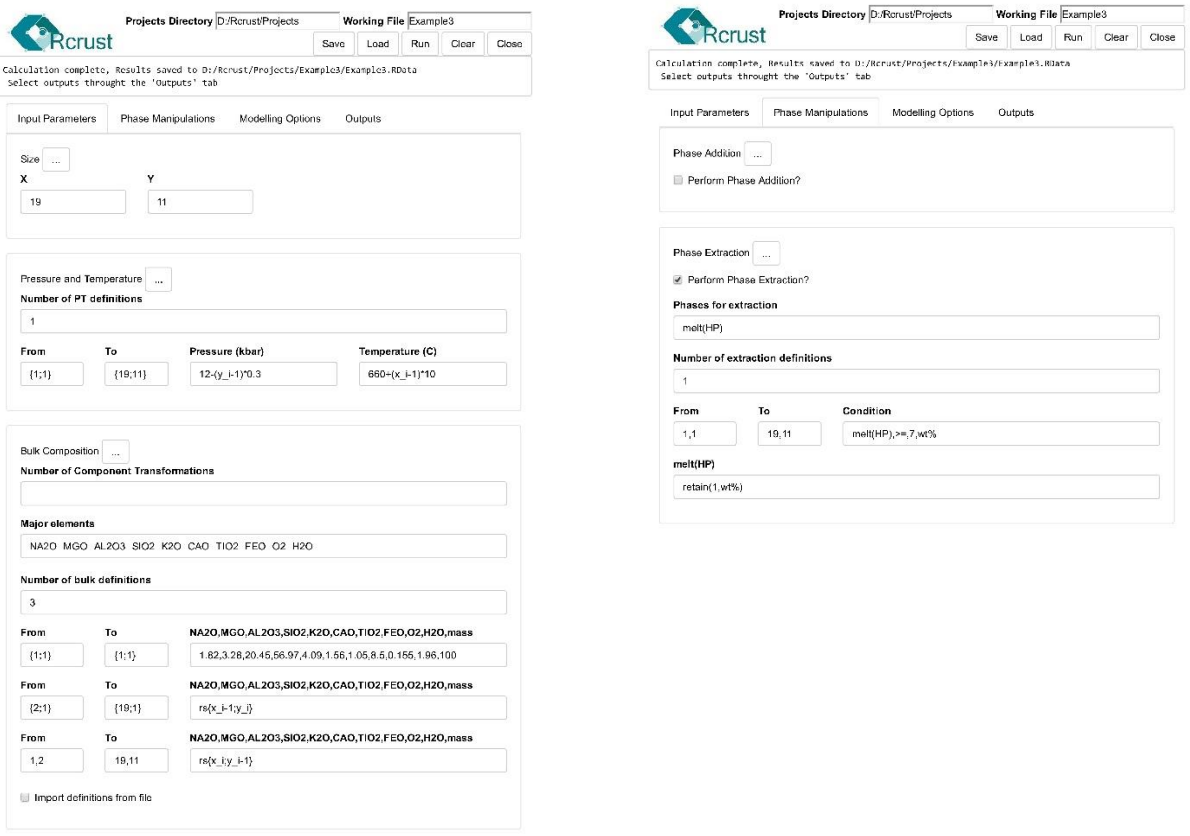


Figure 18 - Rcrust GUI inputs for Example_multi

This example creates a composite path-dependent P-T mode diagram. It does this by first calculating an open system isobaric heating path at 12 kbar (IBH12) and then calculating a number of isothermal decompression paths that each originate from a point on IBH12. Points along IBH12 are each dependent on the reactive subsystem of the point one to the left of itself on the x-axis. Points along decompression paths are each dependent on the point one above itself on the y-axis. Melt loss is defined to occur whenever a 7 wt.% melt threshold is met and melt is extracted until 1 wt.% melt is left behind (this is achieved using the "retain(amount,unit)" function).

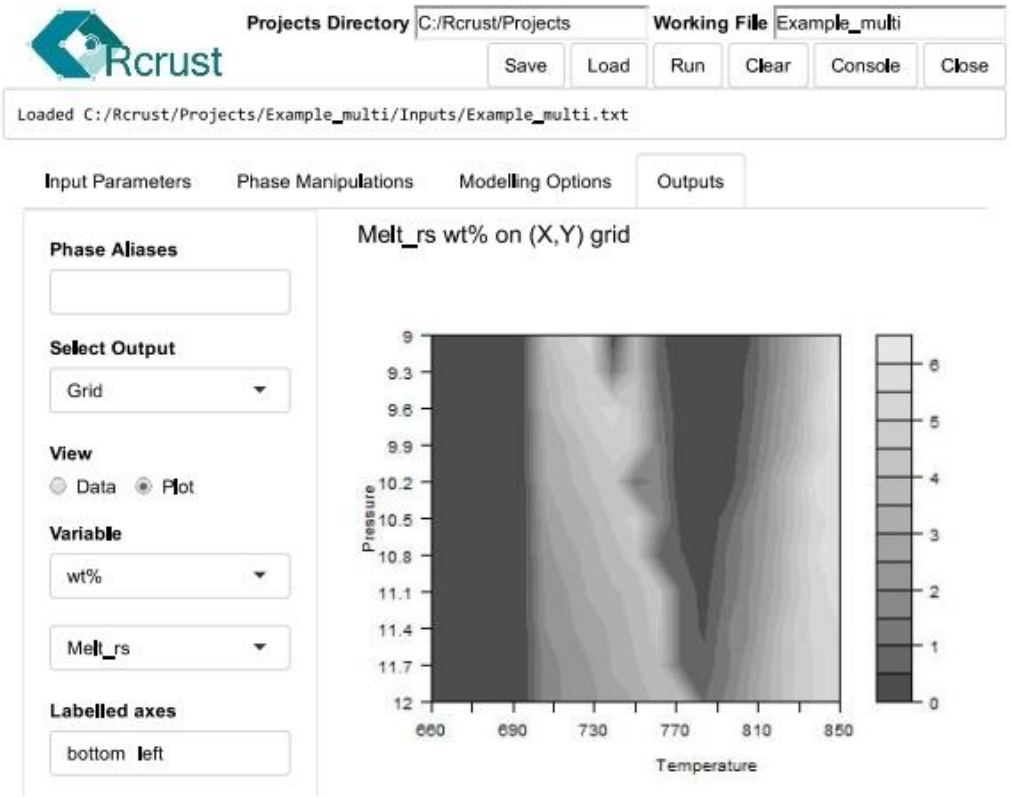
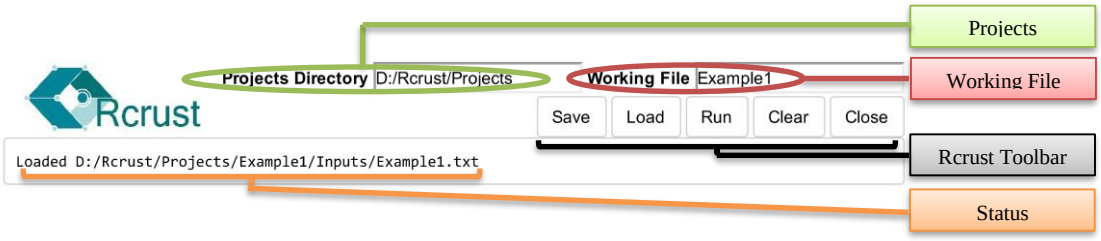


Figure 19 - Grid output for Example_multi showing weight percent of melt in the reactive subsystem (RS)

REFERENCE MANUAL

Rcrust File Management

The top line of the Rcrust GUI hosts a toolbar of file management buttons. User inputs are saved in a text document (**Working File**) which is located in the Projects folder of Rcrust (**Projects Directory**). This file is written, read or run in Rcrust by the **Save**, **Load** and **Run** buttons.



Save

Saves the **Working File's** inputs and calculation results. Each working file is assigned its own project folder in the **Projects Directory**. Parameters currently in the Rcrust GUI are saved to the Inputs folder as a text document. Additional parameters can be passed to Rcrust by placing them in the text document after the line (# Additional Parameters). Calculation results are saved as an R workspace in the project folder.

Load

Loads the **Working File's** inputs and calculation results. Reads the working file from the inputs folder and loads its options in the Rcrust GUI. Replaces the current workspace with that of the **Working File's.**

Run

Saves the current Rcrust GUI inputs and runs the Rcrust calculation according to these parameters. Follow prompts in the R console to calculate the results. Once the results are complete you will be prompted to select outputs through the Rcrust GUI. Outputs written to file are saved in the Ouputs folder of the project Advanced users can access the results directly in the R console by pressing [esc] to activate the console (this is helpful for loading data into GCDkit). To relaunch the Rcrust GUI type 'runApp()' then press [Enter]

Clear

Clears current values in the Rcrust GUI

Console

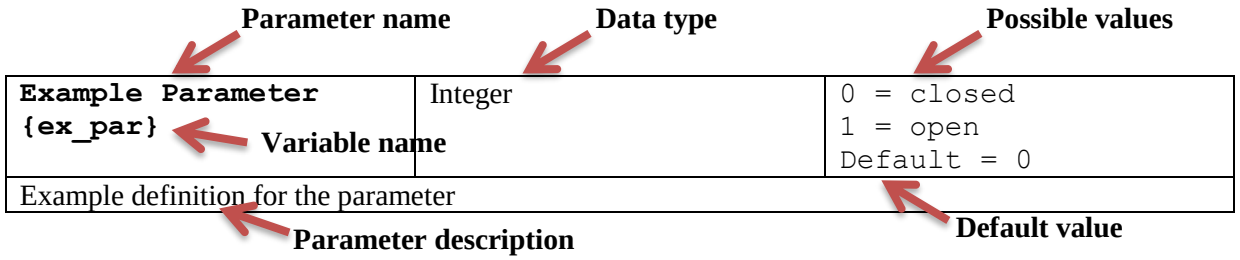
Launches a browser in the R console giving you direct access to the coding environment and all calculated data

Close

Closes the Rcrust GUI. To relaunch the Rcrust GUI type 'runApp()' then press [Enter]

List of Parameters

User inputs are listed here in a systematic fashion for clarity. The parameter name (the name that appears in the Rcrust GUI) is listed first followed by the variable name (the name of the variable accessible in the R console). The data type required for the parameter is listed in the second box. The third box contains possible values for the parameter and identifies any default value. Below this is a description as to what the parameter controls.



Input Parameters

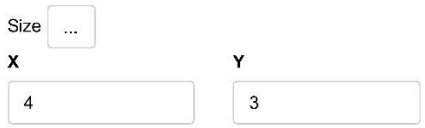
Tuple definitions

From {pt_from_#}	Tuple	{1;1}<= pt_from_#<={x_n;y_n}
The beginning of the definition selection		

To {pt_to_#}	Tuple	pt_from_#<=pt_to_#<={x_n;y_n}
The end of the definition selection		

Size

Specify here the size of the simulation (resolution) you want to calculate: how many points in the X and Y directions.

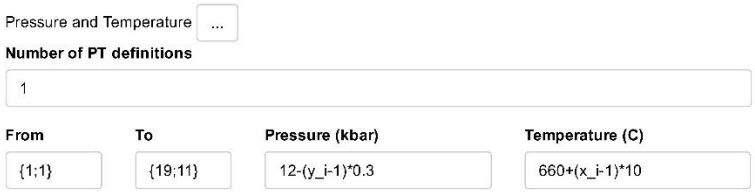


User Interface

X {x_n}	Numeric	1< x_n
The total number of points in the X direction		

Y {y_n}	Numeric	1< y_n
The total number of points in the Y direction		

Pressure and Temperature



User Interface

Number of PT definitions {n_pt_def}	Numeric	1< n_pt_def
The number of definitions to use for assigning pressure and temperature values		

Pressure (kbar) {pressure_#}	Numeric/Expression	
The pressure in kilobars of the Reactive Subsystem (RS). *(1 kbar = 0.1 GPa = 986,92 atm). This can be a constant or an expression built using the variables “x_i” or “y_i”, real numbers, multiplication(*), division(/),addition(+),subtraction(-) or exponents(^).		

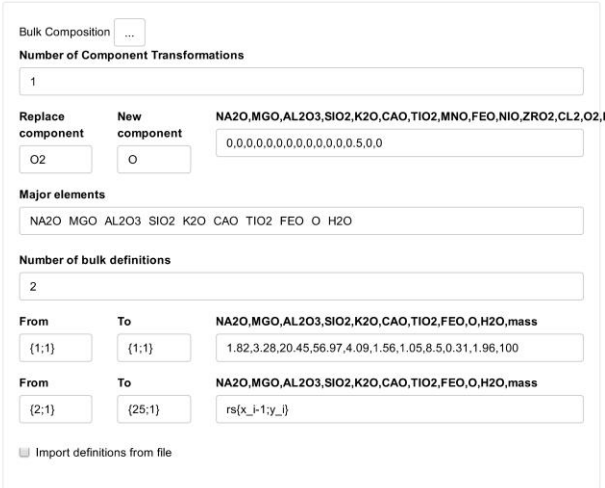
Temperature (°C) {temperature_#}	Numeric/Expression	
The temperature in degrees Celsius of the Reactive Subsystem (RS). *(1 °C = 274.15 K = 33.8°F) This can be a constant or an expression built using the variables “x_i” or “y_i”, real numbers, multiplication(*), division(/),addition(+),subtraction(-) or exponents(^).		

Input File

{pt_def}	String	Options: input
The method of pt definition. *pt definition from file still to come		

{pt_definitions}	Listed Definition	
Pressure and temperature definitions of the form pt_definitions<-list("{x_a,y_a}_{x_b,y_b}"=c("pressure_#","temperature_#"), "{x_a,y_a}_{x_b,y_b}"=c("pressure_#","temperature_#"))		

Bulk composition



User interface

Number of Component Transformations {n_comp_trans}	Numeric	0<= n_comp_trans
The number of component transformations to apply to the currently available oxide components NA2O,MGO,AL2O3,SIO2,K2O,CAO,TIO2,MNO,FEO,NIO,ZRO2,CL2,02,H2O,C02		

Replace component {old_comp_#}	String	
The current component to replace, options are: NA2O,MGO,AL2O3,SIO2,K2O,CAO,TIO2,MNO,FEO,NIO,ZRO2,CL2,02,H2O,C02		

New component {new_comp_#}	String	<6 characters, All capitals
The name of the new component. This name must consist of less than 6 characters and must be all in capital letters.		

NA2O,MGO,... {comp_#}	Comma-separated numeric	
The value of the new component as a factor of the components : NA2O,MGO,AL2O3,SIO2,K2O,CAO,TIO2,MNO,FEO,NIO,ZRO2,CL2,02,H2O,C02. This must be a string of 15 comma separated numbers of which a total of no-more-than 11 can be non-zero. For example, if we wish to use the component O instead of O2 we would use: 0,0,0,0,0,0,0,0,0,0,0,0,0.5,0,0 If we wish to use FE2O3 we would use: 0,0,0,0,0,0,0,0,2,0,0,0,0.5,0,0		

Major elements {major_elements}	Comma-separated-strings	
The major element oxides used to define the bulk composition. The available components are : NA2O,MGO,AL2O3,SIO2,K2O,CAO,TIO2,MNO,FEO,NIO,ZRO2,CL2,02,H2O,C02		

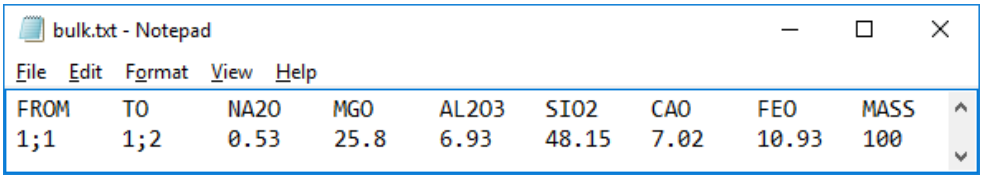
Number of bulk definitions {n_bulk_def}	Numeric	1<= n_bulk_def
The number of definitions to use for assigning bulk compositional values.		

NA2O,MGO,...,mass	Comma-separated-values/expressions	
The wt.% of each elemental oxide listed in “Major elements” (above) as well as the relative starting mass (in grams) of the Reactive Subsystem (RS). Expressions can use real numbers, r operators, x_i, y_i, x_n, y_n and any tuples of the form xs{#;#} where xs is any of rs,as,es,fs and # uses real numbers, r operators, x_i, y_i, x_n and/or y_n		

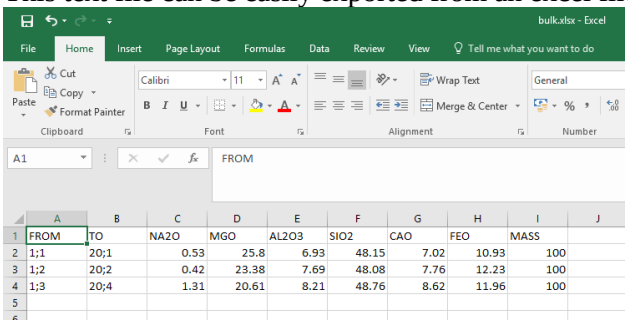
Import definitions from file {bulk_def_file}	Boolean	TRUE = Import from file FALSE = definition via input
Choose whether bulk definitions are read from input or imported from a text file (.txt) located in the Inputs folder.		

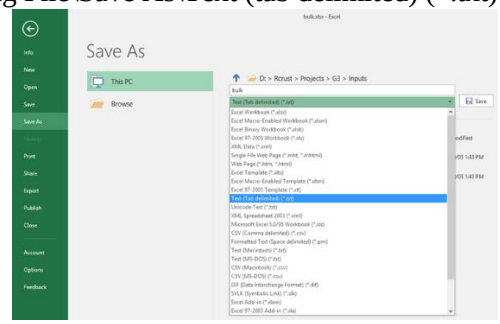
Bulk file {bulk_file}	Character string	e.g. bulk.txt

The name of the input file containing the bulk composition definitions (the name must end with .txt). The bulk file must be a tab delimited text file (.txt) containing the columns “From”, “To”, your selection of major elements and “Mass”. For example:



This text file can be easily exported from an excel file using File\Save As\Text (tab delimited) (*.txt)





The input file must provide bulk definitions for all points in the chosen modelling space {x_n;y_n} with numbers in the “From” and “To” arguments separated by a semicolon (;).

The bulk definitions describe the wt.% of each elemental oxide as well as the relative starting mass (in grams) of the Reactive Subsystem (RS). Expressions can use real numbers, r operators, x_i, y_i, x_n, y_n and any tuples of the form xs{#;#} where xs is any of rs,as,es,fs and # uses real numbers, r operators, x_i, y_i, x_n and/or y_n

Phase Manipulations

Phase Addition

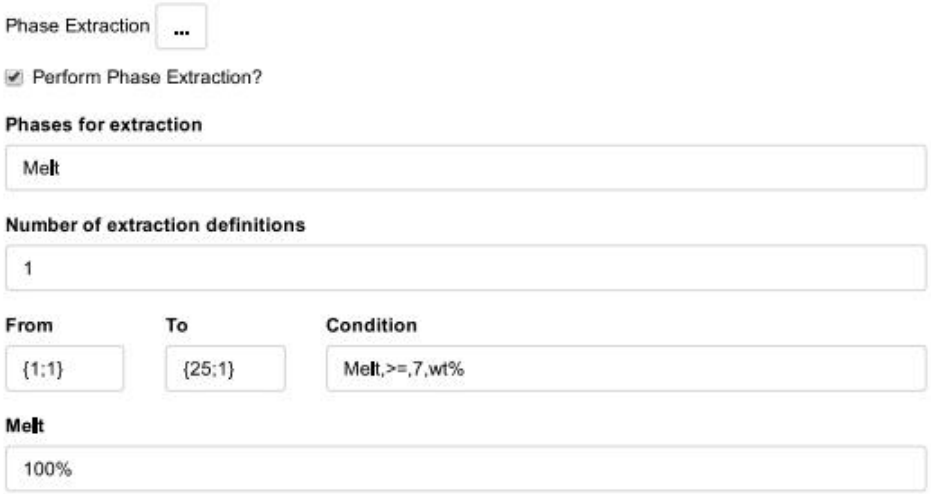
User interface

Perform Phase Addition? {ph_add}	Boolean	TRUE/FALSE
Add phases/components into the Reactive Subsystem (RS) at specified points?		

Number of addition definitions {n_ph_add_def}	Numeric	
The number of definitions to use for assigning phase additions.		

#,#,mass	Comma-separated-values	
The wt.% of each elemental oxide listed in “Major elements” (Bulk composition tab) as well as the relative starting mass (in grams) of the phases/components to add.		

Phase Extraction



User interface

Perform Phase Extraction? {ph_extr}	Boolean	TRUE/FALSE
Extract phases from the Reactive Subsystem (RS) when specified criteria are met?		

Phases for extraction {extr_phases}	Comma-separated-strings	
Phases to be considered for phase extraction. These can be from the list of solution models chosen		

(Phase Models tab) or any pure phase output by Perple_X (see Perple_X Solution Model Glossary) and THERMOCALC's list of mineral abbreviations)

Number of extraction definitions {n_ph_extr_def}	Numeric	
The number of definitions to use for assigning phase extractions.		

Condition	Logical/expression	Options: TRUE FALSE melt(HP),>=,7,wt%
A conditional argument of the form {phase},{condition},{value},{unit} For example to extract phases whenever melt exceeds a 7 vol% threshold you would use the following condition: melt(HP),>=,7,vol%		

[Phase values]	Numeric/expression	Must not be allowed to be greater than the amount present
Define the proportion of phase to extract for each phase listed in "Phases for extraction". Proportions can be given as: 1. A percentage of what is present (e.g. 10%) *you must include the percentage sign for this 2. A set mass (in g) relative to the full system (100g). 3. For melt the expression retain(value,unit) allows extraction to occur until a set amount of melt is left behind where value is the amount and unit is either wt% or vol%.		

Modelling Options

Modelling Data

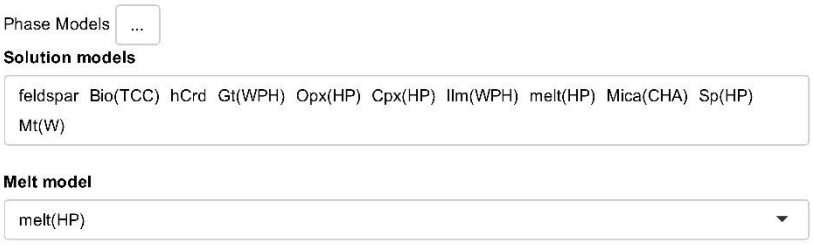
User interface

Thermodynamic Data File {thermodynamic_data_file}	String	Default = hp04ver.dat
The thermodynamic data file present in ~Rcrust/bin/data to be used for phase stability calculations.		

Solution Models File {solution_models_file}	String	Default = solution_model.dat
The solution model file present in ~Rcrust/bin/data from which solution models can be chosen.		

Perple_X Option File {perplex_option_file}	Strings	Default = perplex_option.dat
The perplex option file present in ~Rcrust/bin/data which controls extra settings for phase stability calculations.		

Phase Models



Solution models {use_sol_models}	Comma-separated-strings	
The solution models to use in phase stability calculations sourced from the Solution Models File. (see Perple_X Solution Model Glossary for details)		

Melt model {melt.name}	String	Default = melt(HP)
The solution model to use for melt.		

Outputs

Phase Aliases { phase_aliases}	Comma separated strings	e.g. TiBio(HP)=Bt,Gt(WPH)=Gt
A list of aliases to use for renaming phases of the form TiBio(HP)=Bt,Gt(WPH)=Gt,etc. To hide a phase from plotting use the alias “hide” e.g. TiBio(HP)=hide.		

Select Output = Data File

Select Output = Grid

Select Output = Phase Abundance Along Path

Select Output = PAM

File Management

Functions for file management. See Rcrust File management for more details

Projects Directory {projects_directory}	String	
Absolute location of the projects directory within the Rcrust folder.		

Working File {working_file}	String	{1;1}<= pt_from_#<={x_n;y_n}
Name of the current file under operation. Each file has its own folder within the projects directory		

containing results of simulations (#.RData file), “Inputs” and “Outputs”.

Save {on_save}	Function call	
Saves the current GUI inputs and workspace to the working_file.		

Load {on_load}	Function call	
Loads the previously saved working_file into the GUI inputs and workspace.		

Run {manual_load}	Function call	
Saves the current GUI inputs and workspace to the working_file. Then launches the Rcrust calculation		

Clear {on_clear}	Function call	
Clears the current GUI inputs and workspace.		

Close {stopApp}	Function call	
Closes the current GUI to allow interaction with the R console. To relaunch the Rcrust GUI type 'runApp()' then press [Enter].		

Perple_x options

Options parsed to wrapper calculation set in init_meem.r

Number of chemical components {number_components}	Integer	Default = 15
The number of chemical components to build the major elements from.		

Unit for bulk composition definition	Integer	0 = molar % 1 = weight % Default = 1
The unit proportion to use for bulk composition definition.		

Advanced user options

Static variable options accessible through main.r

Calculation mode {calc_mode}	Character vector	normal Default = normal
Advanced setting toggling the calculation mode.		

PT definition {pt_def}	Character vector	input Default = input
Advanced setting toggling the PT definition mode. Used to allow PT definition from file.		

Reaction buffering {reaction_buffering}	Boolean	T (TRUE) F (FALSE) Default = FALSE
Allows reaction buffering (threshold buffering) whereby phase extractions set on conditions are postponed by the number of reaction buffer steps to ensure continued exceedance of the threshold.		

Commands in the R console

Running Rcrust

Relaunch GUI {.First()} or {Rcrust()}	Function call	
Relaunches the Rcrust GUI from the R console		

Manually initiate calculation {manual_load}	Function call	
Sends the current working file to be calculated		

Outputs

Data file {data_file()}	Function call	
Compiles calculation results into a table		

Write data file {write_data_file}	Function call	
Writes compilation table to file		

Grid data {grid_data()}	Function call	
Compiles an X Y grid of the values of a given variable		

Write grid file {write_grid_file}	Function call	
Writes X Y grid to file		

Useful functions in the R Console

ls()	List all objects in the current environment
Q()	Quits
[Ctrl]+[w]	Toggles buffering of outputs
`Rcrust()`	Manually launches the Rcrust GUI

To access the R console out of a browser click anywhere in the R console window and press [esc]. This will close the Rcrust GUI which has current control over the console. To relaunch the Rcrust GUI at any time simply type runApp() and press [enter].

Rcrust variables

`PT[[bulk]][[step]]$press$temp`	list
List of pressure and temperature conditions for each step in each bulk	

`crust[[bulk]][[step]][phase,detail]`	list
The full system (FS). Contains details of the reactive subsystem (RS) at each step along with cumulative extract (ES) and addition (AS) subsystems. Phases in crust are reported as cumulative weighted averages.	

`c0[detail]`	vector
Bulk composition passed between points	

`workingfile`	Character vector
The current Working File	

`work_dir`	Character vector
The current Working Directory. This is the location of the folder containing the Working File	

Development

Developers of new features should know a few things on the structure of the code. When developing custom functions please stick to these conventions.

The following files are required; they must all be in the same directory (these are contained within the Rcrust folder which should simply be copied to the desired location):

1) From **Perple_X** suite (in a folder called "data"):
    a. The various datafiles you wish to use, these include: thermodynamic datafiles and solution model files, typically *hp04ver.dat* and *solution_model.dat* as well as the Perple_X option file, *perplex_option.dat*.
    b. The rest of perple_x (vertex, build, werami, etc) are not required.

2) From **Rcrust** (directly in the "bin" folder):
    a. ui.r and server.r, these build the Rcrust Graphical User Interface (GUI)
    b. *main.r*, this houses the main calls to run Rcrust
    c. Various init_xxx files, used to transform user input in data structures that Rcrust can understand.
        i. *init_bulk.r* sets the bulk composition(s) of the system
        ii. *init_pt.r* sets the PT conditions for each step/bulk combination
        iii. *init_ph_add.r* sets the phases to add
        iv. *init_ph_extr.r* sets the phases to extract
        v. *init_mem.r* writes user inputs into a meemum build file
        vi. *init_dependene* determines the calculation order of points using the depedences assigned
        vii. *init_wrapper* loads the compiled phase stability calculator from Perple_X
    d. *parse_meem*, a temporary Perple_X build file created to pass data into the wrapper
    e. *run.Rcrust.r*, the Rcrust calculation loop to be called for each point

    Technically, each function works c0 which tracks the bulk compositional changes invoked by phase manipulations. main.r loops through each point, calculating and modifying the phases according to the chosen definitions and eventually stores the final product in a list called crust, whose structure is `crust[[y_i]][[x_i]]`. So, for instance the SiO2 content in the melt of point y_i=4 and x_i=2 is `crust[[4]][[2]]["melt(HP)","SIO2"]` (assuming you use melt(HP) of course).

    f. Various *xxx.dll* files which contain compiled libraries needed to perform calculations within R

Troubleshooting

A list of known errors that are unavoidable or are still to be fixed.

Bulk_ss system properties

Warning: some bulk system properties are reported as molar properties but perplex considers the bulk system to be one mol thus all molar properties need to be adjusted accordingly

Molar phase proportions

Only weight definitions of bulk and phases is currently possible, read.meemum cannot read molar phase proportions. If molar proportions for bulk are entered then bulk is molar but individual phases are weights thus phase extractions crash.

Buffered Output

The R console by default returns a buffered output which forces the console to only refresh when flush.console() is called. To disable the buffering and view run data live deselect from R toolbar Misc/Buffered Output.

REFERENCES

Mayne, M. J., Moyen, J. F., Stevens, G., & Kaislaniemi, L. (2016). Rcrust: a tool for calculating path-dependent open system processes and application to melt loss. *Journal of Metamorphic Geology*, **34**(7), 663-682