The Geochemist’s Workbench®

Release 18

GWB Reaction Modeling Guide

The Geochemist's Workbench®

Release 18

GWB
Reaction
Modeling Guide

Craig M. Bethke
Brian Farrell
Melika Sharifi

Aqueous Solutions, LLC
Champaign, Illinois

Printed May 11, 2025

The GWB software was originally developed by the students, staff, and faculty of the Hydrogeology Program in the Department of Geology at the University of Illinois Urbana–Champaign. The package is currently developed and maintained by Aqueous Solutions LLC.

Address inquiries to:

Aqueous Solutions LLC
301 North Neil Street, Suite 400
Champaign, IL 61820 USA

Warranty: The Aqueous Solutions LLC warrants only that it has the right to convey license to the GWB software. Aqueous Solutions makes no other warranties, express or implied, with respect to the licensed software and/or associated written documentation. Aqueous Solutions disclaims any express or implied warranties of merchantability, fitness for a particular purpose, and non-infringement. Aqueous Solutions does not warrant, guarantee, or make any representations regarding the use, or the results of the use, of the Licensed Software or documentation in terms of correctness, accuracy, reliability, currentness, or otherwise. Aqueous Solutions shall not be liable for any direct, indirect, consequential, or incidental damages (including damages for loss of profits, business interruption, loss of business information, and the like) arising out of any claim by Licensee or a third party regarding the use of or inability to use Licensed Software. The entire risk as to the results and performance of Licensed Software is assumed by the Licensee. See License Agreement for complete details.

License Agreement: Use of the GWB is subject to the terms of the accompanying License Agreement. Please refer to that Agreement for details.

Cover photo: Salinas de Janubio by Jorg Hackemann

Contents

Chapter List

Detailed Contents

Introduction

In geochemical reaction modeling, as the name suggests, you construct numerical simulations of reactions that might occur in the geochemical environment. You might, for example, wish to model the consequences of a specific fluid reacting with a certain set of minerals. For a detailed description of such modeling, as well as a large number of fully worked examples, please see the *Geochemical and Biogeochemical Reaction Modeling* text by Craig M. Bethke, available from Cambridge University Press.

In The Geochemist's Workbench, you use **React** to trace reaction models and **Gtplot** to render graphically the results of **React** simulations.

1.1 React program

React is a flexible program that models equilibrium states and geochemical processes in systems that contain an aqueous fluid. Like **SpecE8**, **React** can calculate the equilibrium distribution of aqueous species in a fluid, a fluid's saturation state with respect to minerals and solid solutions, and the fugacity and partial pressure of gases dissolved in a fluid.

The program can also trace the evolution of a system as it undergoes reversible or irreversible reaction in an open or closed system, either at a given temperature or polythermally. **React** can integrate kinetic rate laws and simulate the fractionation of stable isotopes in a reacting system.

React is, in fact, an extension of program **SpecE8**, which is described in the *GWB Essentials Guide*, in the section **Using SpecE8**. You should, therefore, become familiar with **SpecE8** and its documentation before beginning to work with **React**.

The next three sections of this manual introduce the program, show how to construct reaction paths, show how to trace kinetic reaction paths, and give details on constructing custom rate laws.

1.2 Gtplot program

Gtplot is used to render the results of **React** simulations graphically. The program is driven interactively, using the mouse, and offers the capability of displaying reaction models in a variety of ways. The Using Gtplot section of this manual shows how to use this program.

1.3 Phase2 program

Phase2 is a program for calculating phase and predominance diagrams of virtually any type. The program works by tracing a series of stacked reaction paths across the diagram. Each traverse is conceptually identical to a **React** simulation. As such, **Phase2** can account not only for multicomponent chemical equilibrium, but reaction kinetics, external buffers, sorption and surface complexation, stable isotope fractionation, and, in fact, any of **React**'s many capabilities.

A **Phase2** diagram differs from a simple diagram of the type constructed by **Act2** and **Tact** in that each point in the diagram represents the complete solution to the equations describing the system's distribution of mass. If you were to use **Phase2** to calculate an Eh-pH diagram, for example, you would find that, unlike the result from **Act2**, the boundary lines are curved, rather than linear. Because of the calculation's completeness, some **Phase2** diagrams differ qualitatively from their **Act2** counterparts. For this reason, the diagrams are sometimes referred to as "true" Eh-pH or "true" activity diagrams.

Phase2 diagrams, furthermore, can be plotted over a wide range of variable choices, and in a variety of ways. To learn more about **Phase2** and its abilities, visit the Using Phase2 section of this manual.

1.4 P2plot program

P2plot is used to render the results of **Phase2** simulations graphically. The program is similar to **Gtplot** and **Xtplot**, which are described in the Using Gtplot section of this manual and the **Using Xtplot** section of ***GWB Reactive Transport Modeling Guide***, respectively. Specifically, **P2plot** can diagram phase assemblages, predominant phases and species, gas pressures, isotope compositions, and even cross sections through the diagram. The Using P2plot section of this manual shows how to use this program.

1.5 Example input files

Subsequent chapters of this manual contain examples of applying **React** to trace reaction models in various configurations, and using **Phase2** to create diagrams of differing types. An input file corresponding to each example is available in the GWB installation directory (e.g., “\Program Files\GWB”) under the subdirectory “Script”.

The example files for program **React** are:

• “Feldspar.rea”	Reaction with potassium feldspar
• “Dolomite.rea”	Pumping waste into an injection well
• “IonEx.rea”	Titration path with an ion exchange reaction
• “Evaporate.rea”	Evaporation of seawater
• “Flush.rea”	Reactive fluid migration through quartzite
• “Steam.rea”	Steam flood passing through a reservoir
• “Pyrite.rea”	Dissolution of pyrite
• “Degas.rea”	Gas escaping from a groundwater
• “Uranium.rea”	Distribution of uranium species
• “Surface.rea”	Two-layer surface complexation
• “Goethite_Se.rea”, “Goethite_NaCl.rea”, “Goethite_NaClO4.rea”	Triple-layer surface complexation
• “Goethite_Cu.rea”, “Goethite_P.rea”	CD-MUSIC surface complexation
• “Flash.rea”	Mixing of two fluids
• “Vapor.rea”	Closed and open vapor phases
• “Boiling.rea”	Isobaric boiling
• “Condensation.rea”	Condensation of a vapor phase
• “Isotope.rea”	Tracing stable isotope fractionation
• “Kinetic.rea”	Tracing the kinetic dissolution of albite
• “Albite.rea”	Kinetic dissolution with promoting species
• “Nonlinear.rea”	Nonlinear rate laws
• “Assosciation.rea”	Kinetics of complexation
• “Desorption.rea”	Kinetics of sorption
• “GasTransfer.rea”	Kinetics of degassing
• “Redox.rea”	Kinetics of redox reactions
• “Catalysis.rea”	Catalysis on mineral surfaces
• “Enzyme.rea”	Enzymes and biotransformations
• “Microbes.rea”	Microbial metabolism and growth

The example **Phase2** files are:

“CuSH2O.ph2”	Speciation diagram
“AlSolubility.ph2”	Mineral solubility
“O2Solubility.ph2”	Gas solubility
“CO2_frac.ph2”	Isotope fractionation
“ZnSorption.ph2”	Surface complexation
“Promoting.ph2”	Kinetic diagram

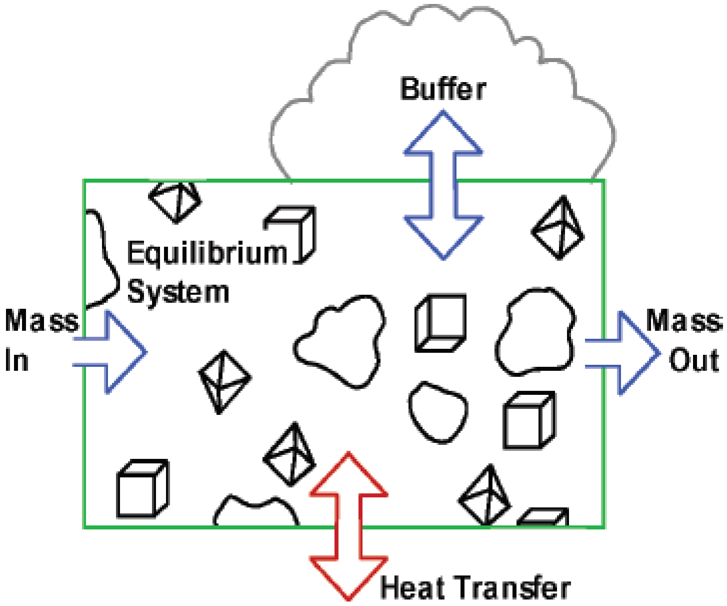
Getting Started with React

React, as described in the Introduction to this guide, is a program designed for modeling irreversible reaction process in geochemical systems. This section provides an introduction to the program, and subsequent sections give details on constructing various types of reaction paths, accounting for kinetic reactions, and constructing custom rate laws.

2.1 Conceptual model

A **React** model calculation has two main components: the initial system and the process that alters it.

React begins a simulation by calculating the system's initial equilibrium state. The program then changes the system by adding (or removing) reactants to vary the system's composition, changing the temperature, or varying the fugacity of gases in an external buffer. This process is the reaction path; a representation is shown below.

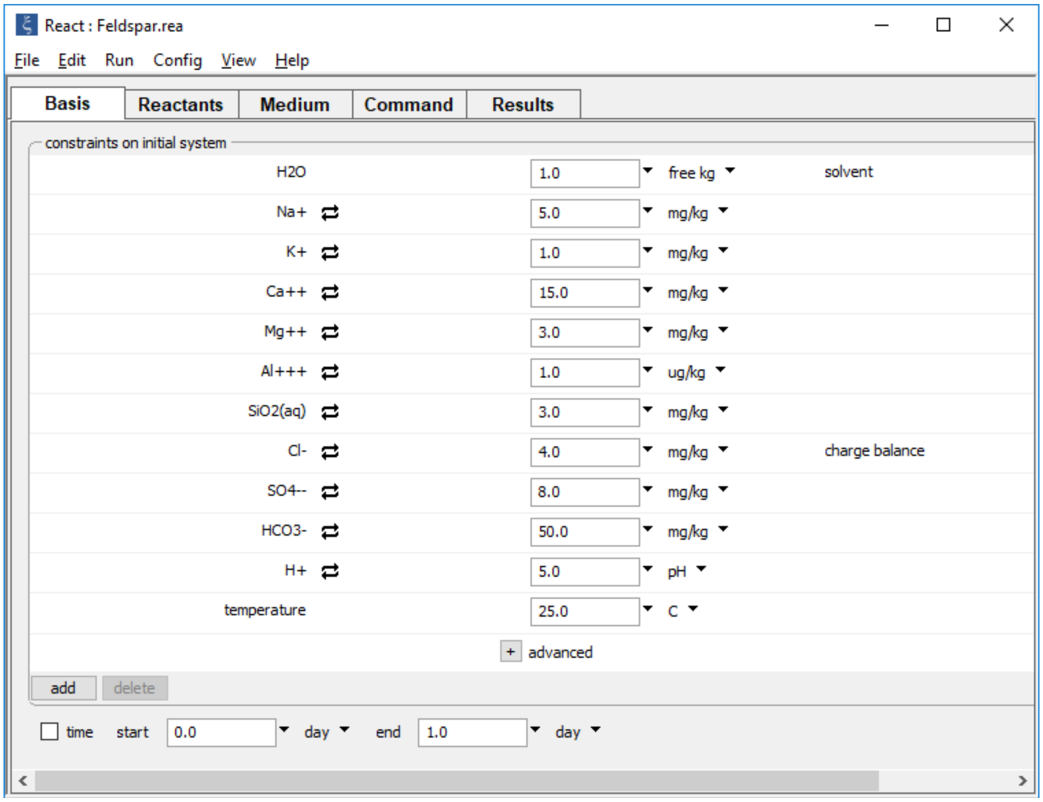


In the figure, a system containing an aqueous fluid and perhaps minerals remains in equilibrium as its composition, temperature, or both change over the course of a reaction path. Changes in composition can reflect explicit transfer of reactants into and out of the system, and the transfer implicit in buffering gas fugacities and species activities.

Progress along the reaction path is measured arbitrarily in terms of a reaction progress variable ξ, which varies from 0 initially to 1 at the end of the path. Variable ξ, which is dimensionless, has meaning only in terms of how the user has defined the reaction path.

2.2 Example calculation

The input file “Feldspar.rea” equilibrates a hypothetical water sample at 25°C and then reacts it with potassium feldspar. Double-click the file, which can be found in the “Scripts” subfolder of the GWB installation directory, to launch **React**. The **Basis** pane



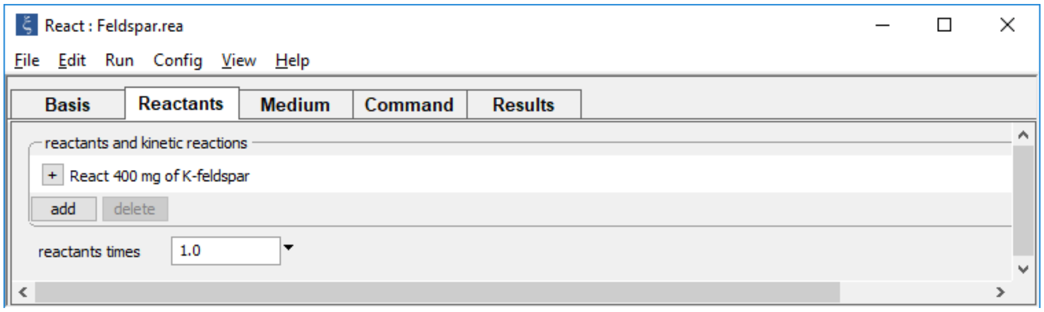
constraints on initial system

H2O	1.0	free kg	solvent
Na+	5.0	mg/kg	
K+	1.0	mg/kg	
Ca++	15.0	mg/kg	
Mg++	3.0	mg/kg	
Al+++	1.0	ug/kg	
SiO2(aq)	3.0	mg/kg	
Cl-	4.0	mg/kg	charge balance
SO4--	8.0	mg/kg	
HCO3-	50.0	mg/kg	
H+	5.0	pH	
temperature	25.0	C	

□ time start 0.0 day end 1.0 day

describes the composition of the hypothetical water.

Move to the **Reactants** pane


React : Feldspar.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
reactants and kinetic reactions
+ React 400 mg of K-feldspar
add delete
reactants times 1.0

to see how potassium feldspar is reacted into the fluid.

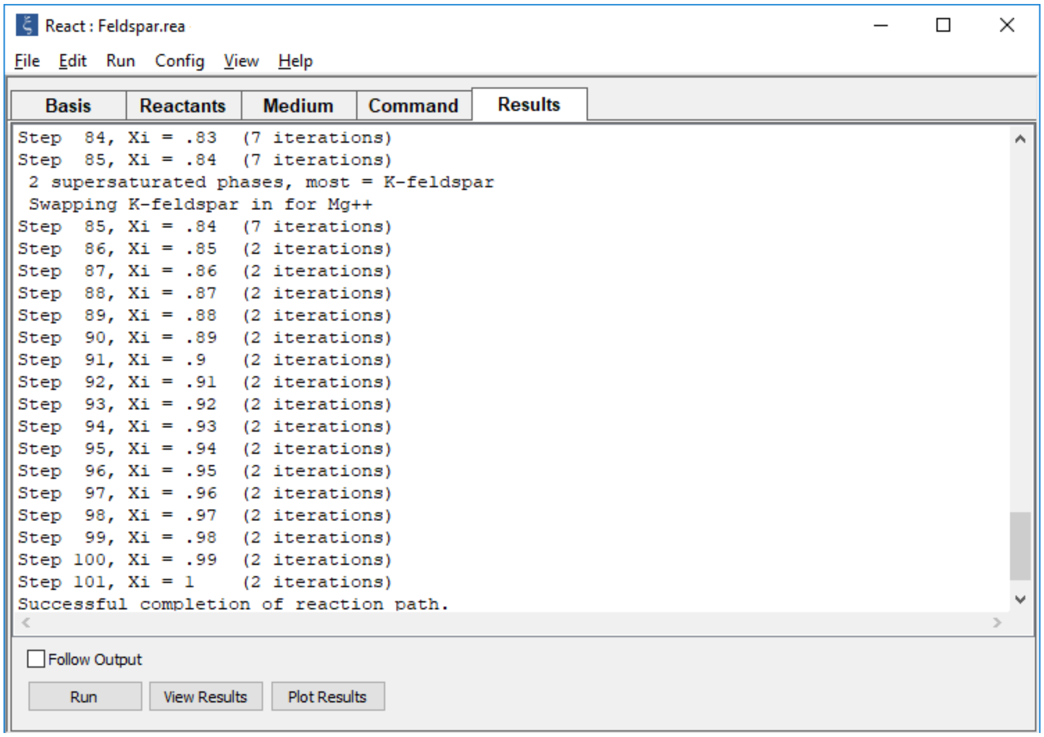
The following commands

```
Na+ = 5 mg/kg
K+ = 1 mg/kg
Ca++ = 15 mg/kg
Mg++ = 3 mg/kg
Al+++ = 1 ug/kg
SiO2(aq) = 3 mg/kg
Cl- = 4 mg/kg
SO4-- = 8 mg/kg
HCO3- = 50 mg/kg
pH = 5
react 400 mg of K-feldspar
```

configure the same reaction path.

When setting the concentration or activity of species, the mass of minerals, or the fugacity or partial pressure of gases in the **Basis** pane, it is important to avoid 0 (zero) values, as these will prevent the GWB programs from solving for the distribution of mass. Where the concentration of a particular component is negligible, set a small positive value. Log values, such as log activity, pH, pe, or Eh, however, can be set to 0.

Press the **Run** button on the **Results** pane to trace the reaction path.

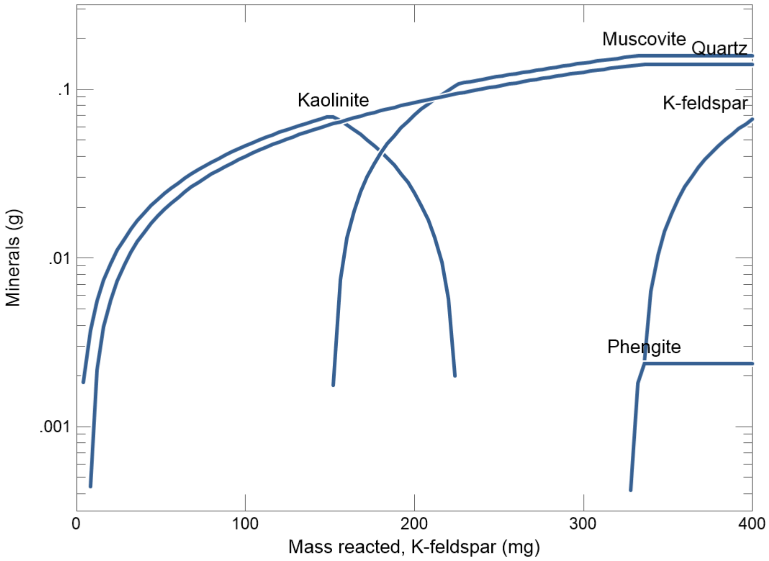


```
Step  84, Xi = .83  (7 iterations)
Step  85, Xi = .84  (7 iterations)
 2 supersaturated phases, most = K-feldspar
 Swapping K-feldspar in for Mg++
Step  85, Xi = .84  (7 iterations)
Step  86, Xi = .85  (2 iterations)
Step  87, Xi = .86  (2 iterations)
Step  88, Xi = .87  (2 iterations)
Step  89, Xi = .88  (2 iterations)
Step  90, Xi = .89  (2 iterations)
Step  91, Xi = .9   (2 iterations)
Step  92, Xi = .91  (2 iterations)
Step  93, Xi = .92  (2 iterations)
Step  94, Xi = .93  (2 iterations)
Step  95, Xi = .94  (2 iterations)
Step  96, Xi = .95  (2 iterations)
Step  97, Xi = .96  (2 iterations)
Step  98, Xi = .97  (2 iterations)
Step  99, Xi = .98  (2 iterations)
Step 100, Xi = .99  (2 iterations)
Step 101, Xi = 1    (2 iterations)
Successful completion of reaction path.
```

You can alternatively select **Run** → **Go** from the menubar or type `go` in the **Command** pane.

React produces as output a dataset "React_output.txt" (**View Results**) that shows calculation results in tabular form, and a dataset "React_plot.gtp" (**Plot Results**) that passes more complete information to the graphics program **Gtplot** (see Using Gtplot). **Act2** and **Tact** can project traces of reaction paths on stability diagrams, and **P2plot** on its 2D diagrams, in which case they also read the "React_plot.gtp" dataset.

The mineralogic results of the reaction, as plotted by **Gtplot**,

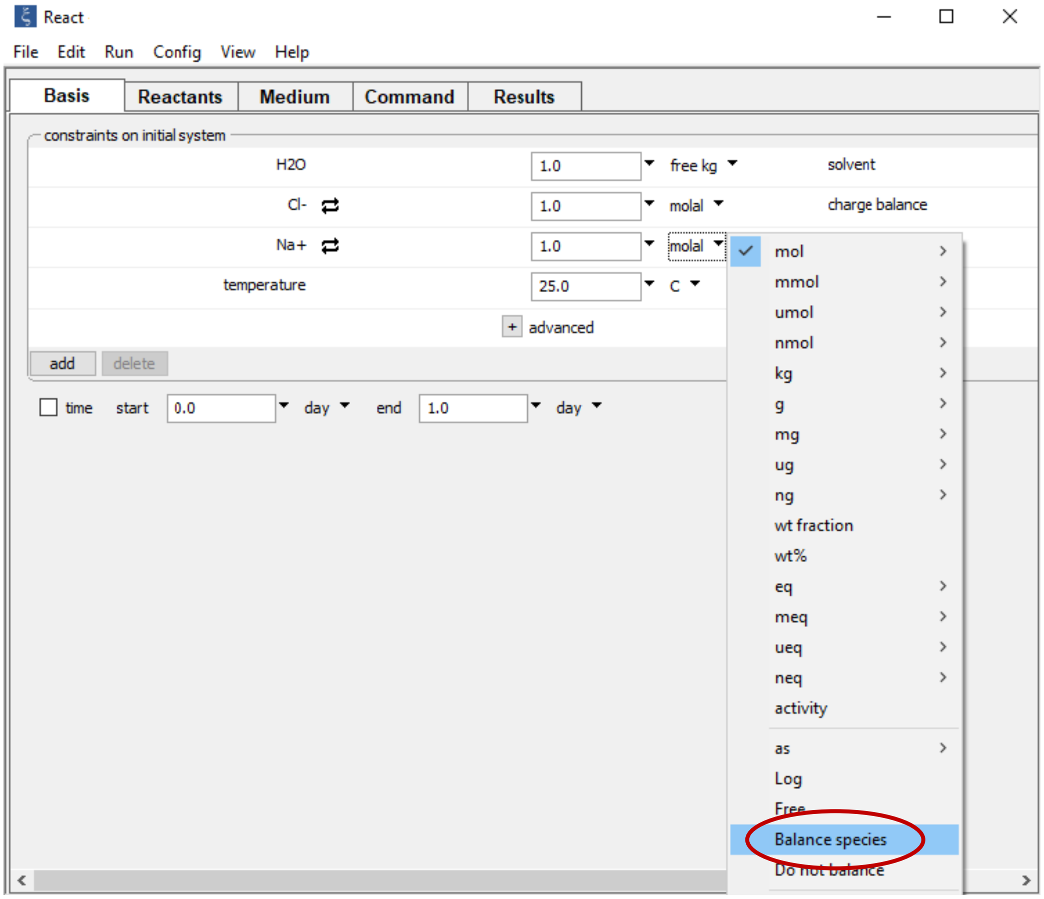


show the results of reacting potassium feldspar into a dilute water at 25°C. The reaction continues until, when about 330 mg of reactant have been added, the fluid reaches equilibrium with K-feldspar.

2.3 Initial system

The initial system in a **React** simulation is an equilibrium model of the aqueous fluid and any coexisting minerals, solid solution end members, and gases. You constrain the initial system precisely as you would set up an equilibrium model with **SpecE8** (see **Equilibrium models** under **Using SpecE8** in the ***GWB Essentials Guide***).

The two programs differ in that **React**, by default, forces electrical charge balance on the system, whereas **SpecE8** does not. To achieve identical output, you should either set **SpecE8** to charge balance on the same basis entry as **React**, or disable charge balancing in **React**. Specify a basis entry to charge balance on by selecting “Balance species” under the entry’s unit pulldown on the **Basis** pane,



constraints on initial system

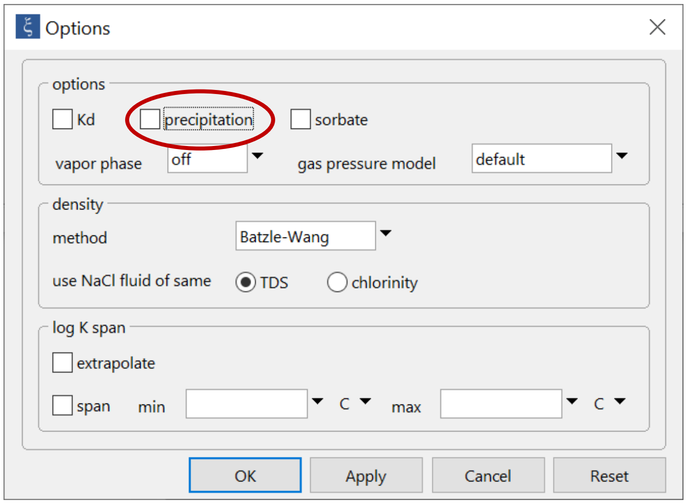
H2O	1.0	free kg	solvent
Cl-	1.0	molal	charge balance
Na+	1.0	molal	
temperature	25.0	C	

+ advanced

□ time start 0.0 day end 1.0 day

and disable charge balance by deselecting “Balance species” from the pulldown. You can also disable charge balance from any charged species’ pulldown, not just the current balancing ion, by selecting “Do not balance” from the pulldown. Since **React**, by default, uses Cl^- as the charge balancing ion, this option is especially useful for systems that don’t include any Cl^-.

The programs also differ in that **React**, having figured the distribution of species in the initial system, will by default attempt to find the system's true equilibrium state by allowing any supersaturated minerals or solid solutions to precipitate. If you do not want the program to take this step, click on **Config → Options…**



Options

options

□ Kd □ precipitation □ sorbate

vapor phase: off

gas pressure model: default

density

method: Batzle-Wang

use NaCl fluid of same: ● TDS ○ chlorinity

log K span

□ extrapolate

□ span min C max C

and uncheck the box labeled "precipitation".

If you run **React** on an initial system containing supersaturated minerals, and do not disable the precipitation option, you should expect two blocks of output in "React_output.txt" (**File → View → React_output.txt**). Each block contains calculation results, including the concentration of a number of aqueous species, the saturation state of various minerals and solid solutions, the fugacity and partial pressure of gases, and the system's bulk composition.

If, for example, you run **React** on the seawater example provided for **SpecE8** (the input is given in **Equilibrium models** under **Using SpecE8** in the ***GWB Essentials Guide***), the output dataset will contain two blocks. The top block gives results of the first calculation performed by **React**. The calculation predicts that seawater is supersaturated with respect to a number of minerals, including antigorite and dolomite. The results in this block, then, represent seawater in a metastable state. The fluid's bulk composition (but not the concentrations of free species), as well as the CO_2 and O_2 fugacity, reflect the input constraints.

The program continues to calculate the equilibrium state that seawater, according to the model, would reach if supersaturated minerals were allowed to precipitate. The second block in the dataset gives results of this calculation, which predicts that, at equilibrium, dolomite and quartz coexist with seawater. Precipitating these minerals alters the fluid composition so that the pH and CO_2 fugacity reported in the second block differ from the original values.

Finding the true equilibrium state of seawater is a somewhat academic pursuit because mineral precipitation from seawater is too slow to observe. You can tell **React** to not allow any new minerals or solid solutions to form by turning off the "precipitation" option on the **Config → Options…** dialog, or with the `precip off` command. In this case, only one block of results would appear in "React_output.txt".

2.4 Redox disequilibrium

React, like the other GWB programs, can account for the redox disequilibrium commonly observed in natural waters. You control the extent to which **React** honors redox equilibrium when calculating a chemical model by interactively enabling and disabling redox coupling reactions.

See **Redox couples** in the **Configuring the programs** section of the ***GWB Essentials Guide*** for information about decoupling redox reactions, and **Redox disequilibrium** in the **Using SpecE8** section of that guide for specific instructions on constructing disequilibrium models in **React.**

2.5 Activity coefficients

Like **SpecE8**, **React** employs various methods for calculating activity coefficients, including:

- The "B-dot" equation, an extended form of the Debye-Hückel equation
- The Specific ion Interaction Theory (SIT method)
- The Harvie-Møller-Weare implementation of the "Pitzer equations"
- The activity models implemented in the MINTEQ, PHREEQC, PHRQPITZ, and WATEQ4F geochemical modeling programs, as well as the Davies equation

For more information, see **Activity coefficients** in the **Using SpecE8** section of the ***GWB Essentials Guide*.**

2.6 Solid solutions

React, like the other GWB programs, can account for binary solid solutions (See **Solid solutions** in the **Configuring the programs** section of the ***GWB Essentials Guide***). Like **SpecE8**, it can consider continuous and discrete modes, and in either case it can treat solid solutions as ideal or calculate activity coefficients according to various models:

- Site-mixing
- Guggenheim
- Regular
- Cubic Maclaurin
- Third-order Maclaurin

For more information on including solid solutions in **React**, see **Solid solutions** in the **Using SpecE8** section of the ***GWB Essentials Guide***.

2.7 Sorption onto mineral surfaces

React can model the sorption of species from solution onto mineral surfaces, according to several methods (see **Sorption onto mineral surfaces** in the **Configuring the programs** section of the ***GWB Essentials Guide***). The methods include

- Two-layer surface complexation
- Triple-layer surface complexation
- CD-MUSIC
- Ion exchange
- Distribution coefficients (K_d)
- Freundlich isotherms
- Langmuir isotherms

For more information on applying these methods within **React** simulations, see **Sorption onto mineral surfaces** in the **Using SpecE8** section of the ***GWB Essentials Guide***.

2.8 Settable variables

React allows you to alter the values of certain variables carried in the calculation. Table 2.1 lists these variables and their units and default values. To change a variable, type the new value on the **Basis** or **Medium** pane or in the dialog boxes under **Config**, or enter as a command the variable name followed by the new value. Examples:

```
itmax = 1000
delxi = 5/100
```

The value given must be in the units carried internally by the program, as listed in Table 2.1. To restore a variable to its default value, enter a blank field in the dialog box, or on the **Command** pane type its name without a value or followed by a `?`:

```
delxi
delxi = ?
```

Current values of the variables can be viewed at any time on the relevant dialog box, or using the command `show variables`.

Table 2.1 Settable variables, their units and default values

Pane or Dialog	*Variable*	*Default*	*Unit*	*Description*
Basis	`density`	*not set*	g/cm^3	Density of initial fluid.
	`pressure`	*auto*	bar	Confining pressure.
	`TDS`	*not set*	mg/kg	Total dissolved solids, initial fluid.
Medium	`cpw`	1	cal/g°C	Fluid (water) heat capacity (polythermal mixing).
	`cpr`	.2	cal/g°C	Mineral (rock) heat capacity (polythermal mixing).
	`cpv`	*auto*	cal/g°C	Vapor heat capacity (boiling).
	`hvapor`	*auto*	cal/mol	Heat of vaporization.
	`inert`	0	cm^3	Unreactive volume (porosity calculation).
Iteration	`epsilon`	5×10^{-11}	–	Convergence criterion.
	`itmax`	999	–	Allowed number of iterations.
	`nswap`	30	–	Allowed number of basis changes to find stable mineral assemblage, reaction step.
	`pitz_dgamma`	0.1	–	For virial activity coefs., maximum fractional change per iteration.
	`pitz_precon`	10	–	For virial activity coefs., maximum number of pre-conditioning steps.
	`pitz_relax`	0.5	–	For virial activity coefs., relaxation factor during iteration.
	`simax`	3	molal	Maximum stoichiometric ionic strength used to

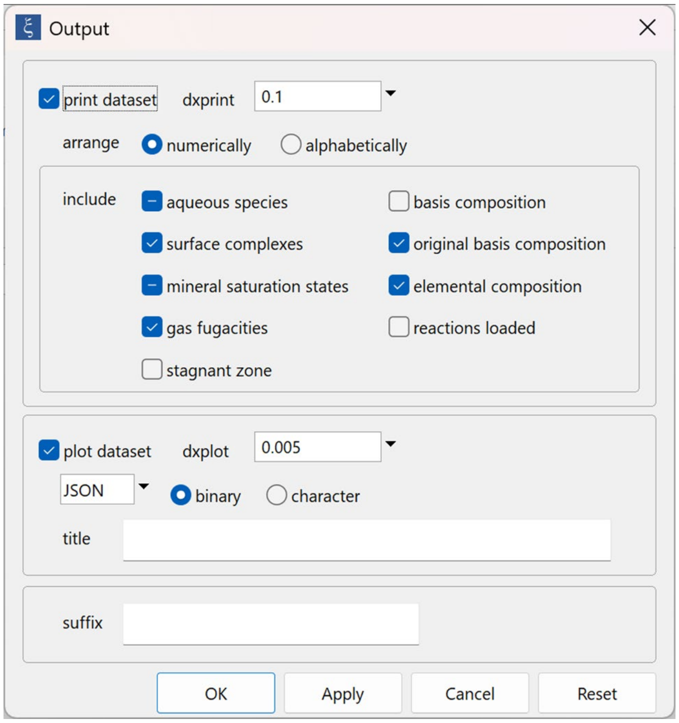
				calculate water activity (Debye-Hückel model).
	`timax`	3	molal	Maximum true ionic strength used to calculate activity coefficients (Debye-Hückel model).
Stepping	`cpu_max`	not set	s	Amount of computing time a simulation may take.
	`delQ`	.1	–	$\Delta Q/Q$ allowed over a step, for kinetic reactions.
	`delxi`	.01	–	Step size in reaction progress, log/linear stepping.
	`dx_init`	*not set*	–	Initial step size in reaction progress.
	`step_increase`	1.5	–	Maximum proportional increase in step size.
	`step_max`	*not set*	–	Number of reaction steps the program may take to trace a simulation.
	`theta`	.6	–	Time weighting for tracing kinetic rate laws.
	`Xstable`	1	–	Factor for stability criterion within stagnant zones.
Output	`dxplot`	.005	–	Interval in reaction progress between plot points.
	`dxprint`	.1	–	Interval between printing results.

2.9 Controlling the printout

React lets you control the amount of detail presented in the calculation results written to "React_output.txt". The dataset is known commonly as the "printout", although it need never be printed. The printout can contain information about (1) the concentration and activity of dissolved species, (2) the saturation indices of various minerals, (3) gas fugacity and partial pressure, (4) surface complexes, and (5) the system's bulk composition in terms of elements or either the current or original basis, and (6) the diffusion profile within the stagnant zone. The printout can also contain a list of the chemical reactions considered in the calculation and their log *K*s, as well as values of virial interaction coefficients.

You can specify that any of these lists be excluded from the printout or included in a short or long format. The program, by default, produces lists of dissolved species and mineral saturation states in a short format, which includes only species present in concentrations greater than 10^{-8} molal and minerals with saturation indices greater than –3. The fugacity and partial pressure of each gas appears in the printout, as does composition in terms of elements and the original basis.

To vary the printout, the user can set each aspect in Table 2.2 to "none", "long", or "short" on the **Config → Output…** dialog



Output

☑ print dataset dxprint 0.1

arrange ● numerically ○ alphabetically

include
⊟ aqueous species □ basis composition
☑ surface complexes ☑ original basis composition
⊟ mineral saturation states ☑ elemental composition
☑ gas fugacities □ reactions loaded
□ stagnant zone

☑ plot dataset dxplot 0.005

JSON ● binary ○ character

title

suffix

or in a command. For example,

```
print species = long
print saturations = none
```

The current print settings can be viewed at any time by looking at the three-way check boxes on the **Config → Output…** dialog. A blank check box ☐ indicates exclusion from the printout, a check ☑ indicates inclusion in long format, and the indeterminant selection ⊟ indicates inclusion in short format. Alternatively, type

```
show printout
```

Table 2.2 Print options and default settings in React

Option	**Default**	**Description**
`print dataset`	on	To “React_output.txt”
`print alphabetically`	off	Arrange entries in dataset alphabetically
`species`	short	Aqueous species
`surfaces`	long	Surface complexes
`saturations`	short	*S.I.* of mineral and solid solutions
`gases`	long	Gas fugacity and partial pressure
`basis`	none	Composition, current basis
`orig_basis`	long	Composition, original basis
`elements`	long	Elemental composition
`reactions`	none	Chemical reactions
`stagnant`	none	Composition of stagnant zone

2.10 React command line

You can start **React** by clicking the icon on the GWB dashboard, opening a “.rea” file, or entering the command `react.exe` from the Windows “Command Prompt”.

When you start **React** from the command line (as opposed to clicking on the icon), you can specify a number of arguments. For example, the command

```
react -i my_script -d my_thermo.tdat
```

causes **React** to read input commands from a file “my_script”, and to use “my_thermo.tdat” as the thermodynamic database.

The following options are available from the command line:

`-cd`	Change the working directory to the directory containing the input script specified with the –i option.
`-nocd`	Do not change the working directory.

`<input_script>`
`-i <input_script>`
	Set a file from which to read input commands.

`-gtd <gtdata_dir>`	Set directory to search for thermodynamic datasets.
`-cond <cond_data>`	Set the dataset for calculating electrical conductivity.
`-d <thermo_data>`	Set the thermodynamic dataset.
`-iso <isotope_data>`	Set the dataset of isotope fractionation factors.
`-s <surface_data>`	Set a dataset of surface sorption reactions.

Tracing Reaction Paths

React can trace several types of reaction paths, including

- Titration paths, in which a reactant is gradually added to a geochemical system.
- Polythermal paths, in which temperature varies.
- Sliding paths, in which a species activity or gas fugacity varies.
- Kinetic paths, in which the rate of reaction progress is described by a kinetic rate law.

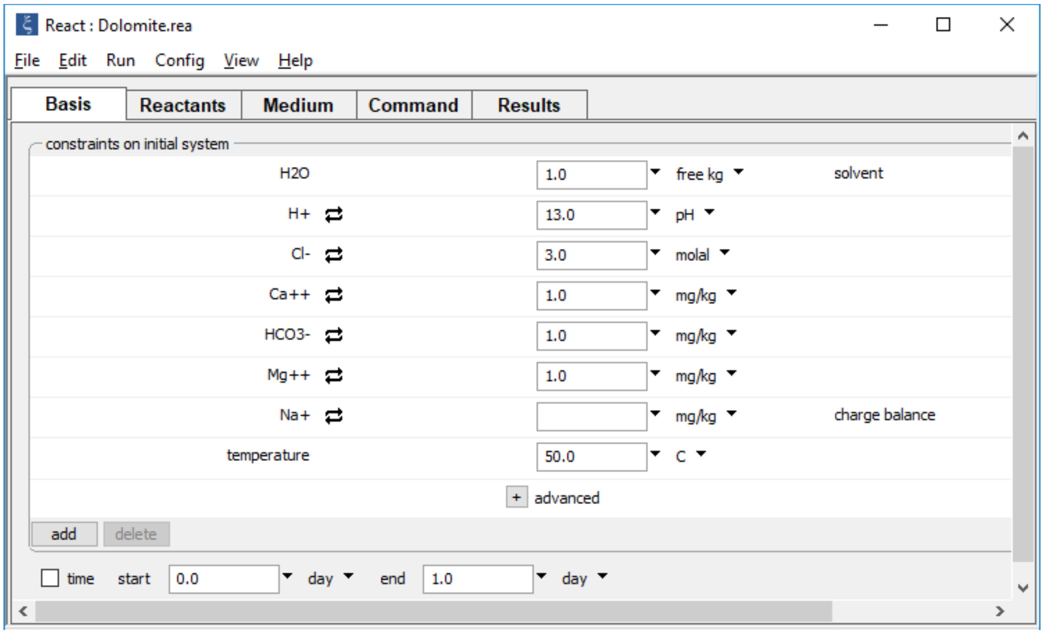
These types of paths can be combined into a single **React** simulation.

This section of the guide describes how to construct the first three types of reaction paths, and the following two sections give details on constructing kinetic reaction paths.

3.1 Titration paths

In simple reaction paths, one or more reactants are gradually added to (or removed from) the system over the course of the calculation. Such paths are sometimes called "titration paths" because the program repeatedly adds a small aliquot of reactant and then recalculates the system's equilibrium state as it steps forward in reaction progress. Other methods of varying the system composition, such as **buffered paths**, **sliding activity and fugacity paths**, and **kinetic paths**, are discussed in later sections of this manual.

In the following example, the program simulates reaction of an alkaline industrial waste that has been pumped into an injection well with a dolomite formation receiving the waste. Double-click on file "Dolomite.rea" (the file can be found in the "Scripts" folder within the GWB installation directory), and when **React** opens, look at the **Basis** pane,



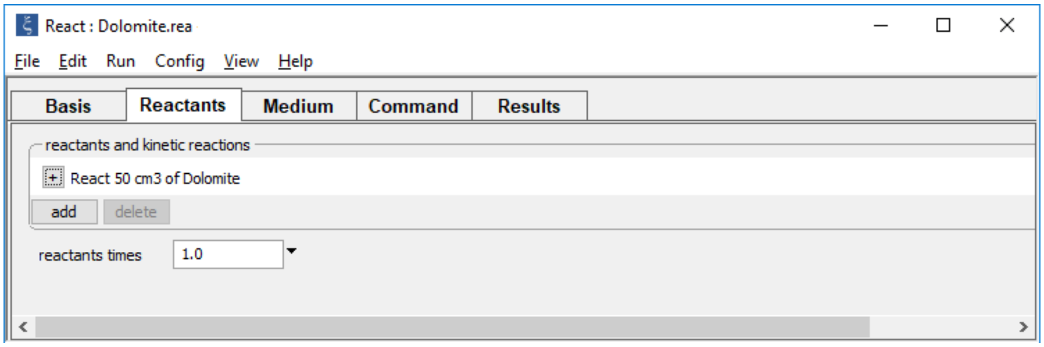
H2O	1.0	free kg	solvent
H+	13.0	pH	
Cl-	3.0	molal	
Ca++	1.0	mg/kg	
HCO3-	1.0	mg/kg	
Mg++	1.0	mg/kg	
Na+		mg/kg	charge balance
temperature	50.0	C	

□ time start 0.0 day end 1.0 day

which contains the composition of the alkaline waste. You don’t need to constrain the concentration of Na^+ because it’s selected as the charge balancing ion.

If the fluid composition wasn’t already set, you would do so by clicking the add button to add entries to the basis, then select units from the pulldown for each entry and constrain their values. To set Na^+ as the charge balancing ion, select the “Balance species” option from its unit pulldown.

Move to the **Reactants** pane.



React 50 cm3 of Dolomite

reactants times 1.0

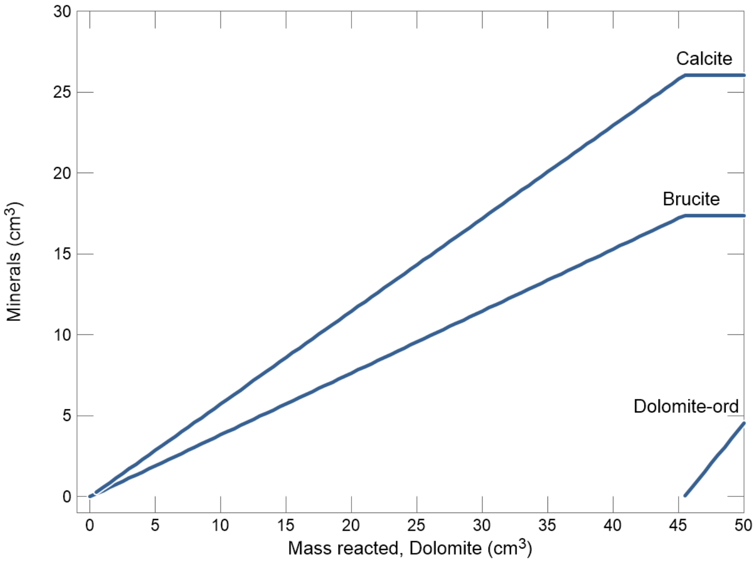
The input sets up the irreversible reaction of dolomite into the NaOH-NaCl waste. If the titration path isn’t already set up, you can do so by going to add → **Simple** → **Mineral…** → **Dolomite** and specifying the amount to react.

You could also have used the commands

```
T = 50
pH = 13
Cl- = 3 molal
Ca++ = 1 mg/kg
HCO3- = 1 mg/kg
Mg++ = 1 mg/kg
balance on Na+
react 50 cm3 Dolomite
go
```

The initial commands correspond to the settings on the **Basis** pane, while the final step matches the input to the **Reactants** pane.

In the example, the program reacts 50 cm^3 of dolomite over the reaction path. If the program takes 100 reaction steps of $\Delta\xi = .01$, then a half cm^3 of dolomite is added to the system at each step. As can be seen in the calculation results,



dolomite reacts with the fluid to form calcite and brucite until the fluid becomes saturated. Reaction stops at this point, so that any dolomite further added to the system simply accumulates.

Aqueous species and gases, as well as minerals, solid solution end members, and fictive oxide components, may serve as reactants in titration paths. Reactant quantities can be set in a variety of units, listed in Table 3.1.

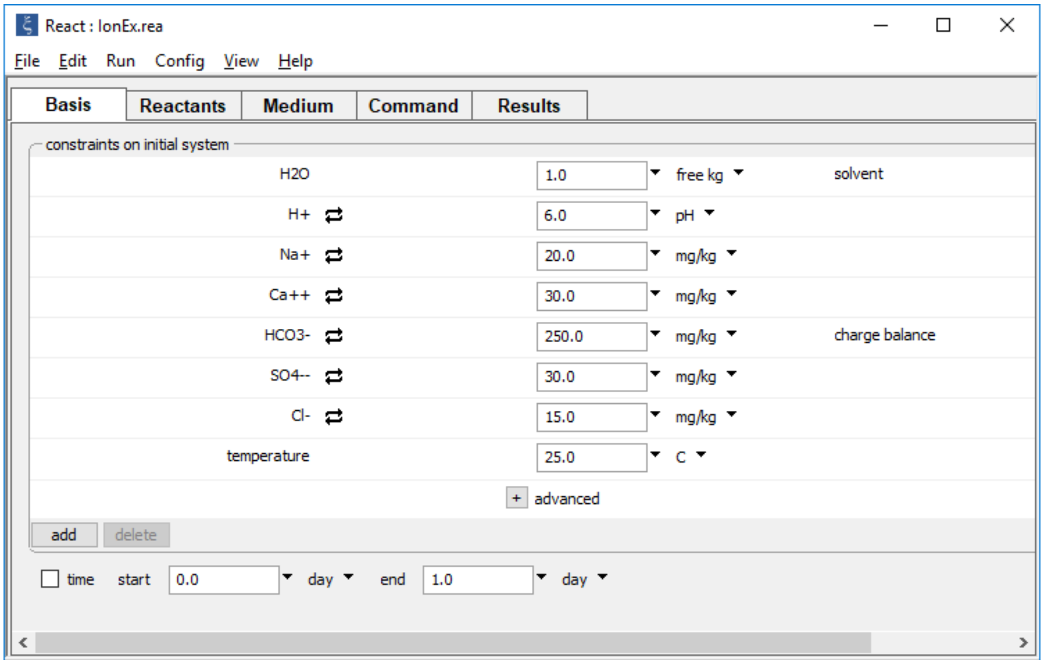
Table 3.1 Units for specifying reaction rates

By amount to add over the reaction path:
mol, mmol, umol, nmol
kg, g, mg, ug, ng
cm3
eq, meq, ueq, neq
mol/kg, mmol/kg, umol/kg, nmol/kg
molal, mmolal, umolal, nmolal
mol/l, mmol/l, umol/l, nmol/l
g/kg, mg/kg, ug/kg, ng/kg, wt fraction, wt%
eq/kg, meq/kg, ueq/kg, neq/kg
eq/l, meq/l, ueq/l, neq/l
mol/cm3, mmol/cm3, umol/cm3, nmol/cm3
kg/cm3, g/cm3, mg/cm3, ug/cm3, ng/cm3
mol/m3, mmol/m3, umol/m3, nmol/m3
kg/m3, g/m3, mg/m3, ug/m3, ng/m3
volume%, vol. fract.
By absolute rate (append):
/s, /min, /hr, /day, /yr, /m.y.

As many reactants as desired can be specified: click add → **Simple** on the **Reactants** pane, or use a `react` command for each. If multiple reactants are set, the program adds each to the system at a rate in proportion to its total mass.

The role of reactants is to change the system's composition. For this reason, only the reactant's composition is important. Quartz, for example, could be replaced as a reactant by the silica polymorph tridymite, SiO_2(aq), or even the fictive oxide SiO_2 without affecting the calculation results.

For an example of a titration path accounting for an ion exchange reaction, double-click on file “IonEx.rea”. When it opens, go to the **Basis** pane,


React : IonEx.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete

constraints on initial system

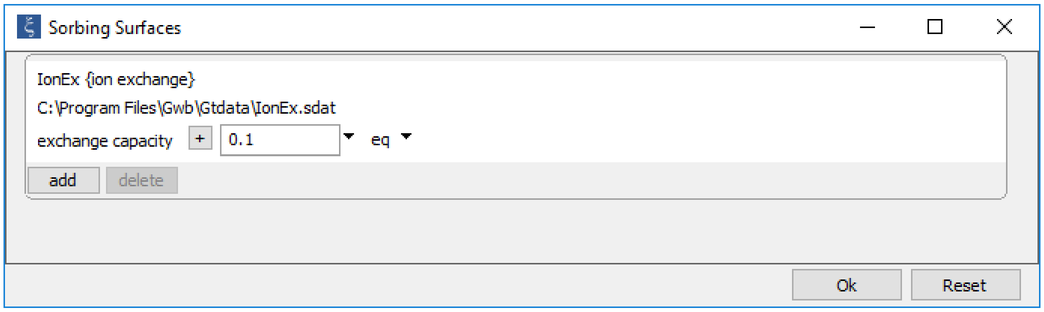
H2O	1.0	free kg	solvent
H+	6.0	pH	
Na+	20.0	mg/kg	
Ca++	30.0	mg/kg	
HCO3-	250.0	mg/kg	charge balance
SO4--	30.0	mg/kg	
Cl-	15.0	mg/kg	
temperature	25.0	C	

+ advanced

□ time start 0.0 day end 1.0 day

which describes a fluid containing Ca^{++} and Na^{+}.

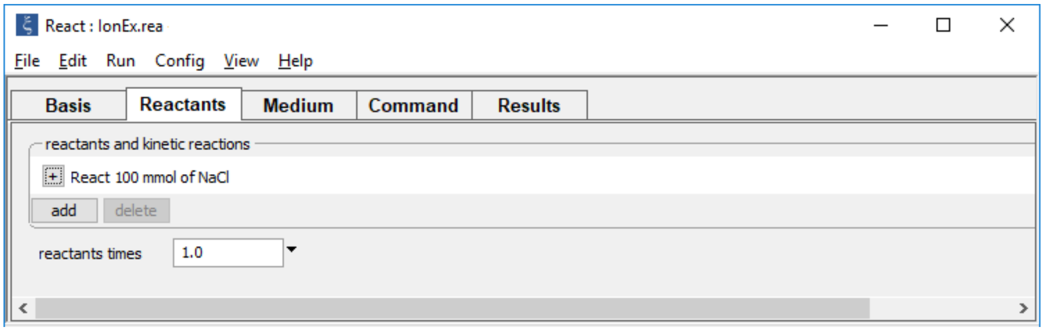
Load an exchanging surface into the program by going to **File → Open → Sorbing Surfaces…** and add the “IonEx.sdat” dataset from the “Gtdata” folder within the GWB installation directory.



IonEx {ion exchange}
C:\Program Files\Gwb\Gtdata\IonEx.sdat
exchange capacity + 0.1 eq

The dataset contains selectivity coefficients for exchange between Ca^{++} and Na^{+}.

Move to the **Reactants** pane,

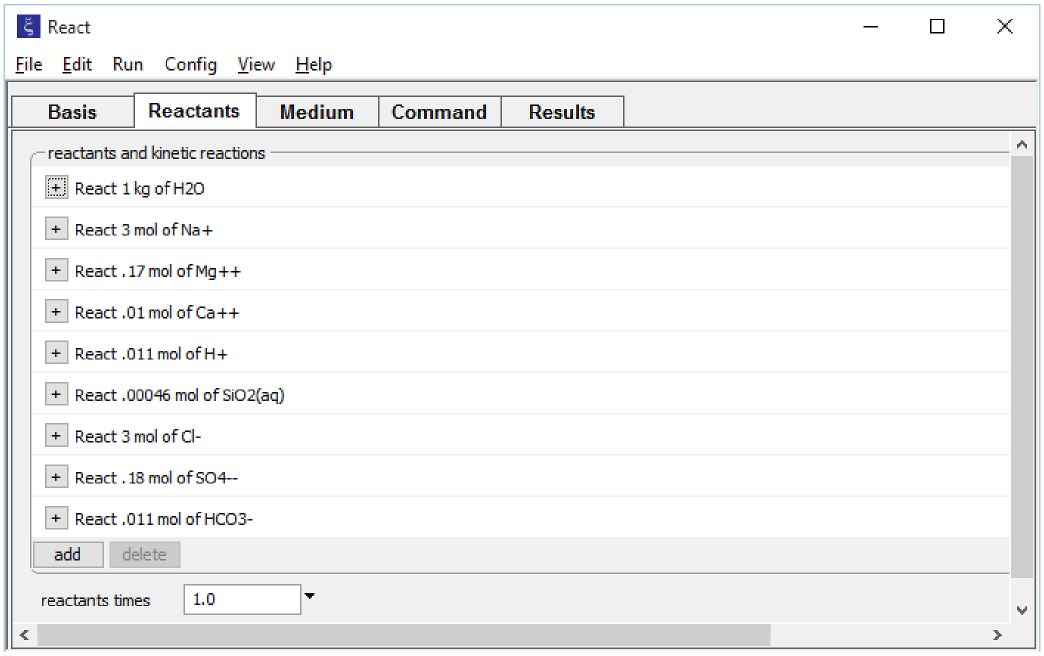

React : IonEx.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
reactants and kinetic reactions
add delete

React 100 mmol of NaCl

reactants times 1.0

where the program is configured to titrate NaCl into the initial system. In this example, the Na^+ added to the solution drives an ion exchange reaction that liberates Ca^{++} from the exchanging surface.

To use a fluid as a reactant, you must determine the fluid's bulk composition by calculating an equilibrium model. Each component in the fluid appears on a separate line. After defining the initial system, the **Reactants** pane might look like

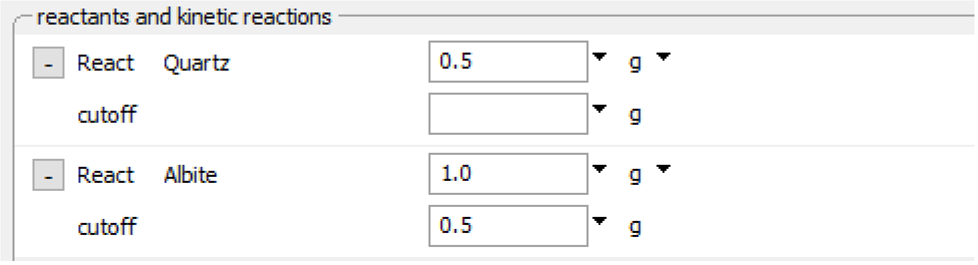

React
File Edit Run Config View Help
Basis Reactants Medium Command Results
reactants and kinetic reactions
add delete

React 1 kg of H2O
React 3 mol of Na+
React .17 mol of Mg++
React .01 mol of Ca++
React .011 mol of H+
React .00046 mol of SiO2(aq)
React 3 mol of Cl-
React .18 mol of SO4--
React .011 mol of HCO3-

reactants times 1.0

React provides a convenient method to take the fluid composition resulting from an equilibrium model and use it as a reactant, without reentering the reactant's composition as done here (see Picking up the results of a run, later in this section).

React normally adds reactants at rates that remain constant over the path, but a cutoff value may be set on the **Reactants** pane (click + next to the reactant in question), or with the `react` command. The input



reactants and kinetic reactions

-	React	Quartz	0.5	g
	cutoff			g
-	React	Albite	1.0	g
	cutoff		0.5	g

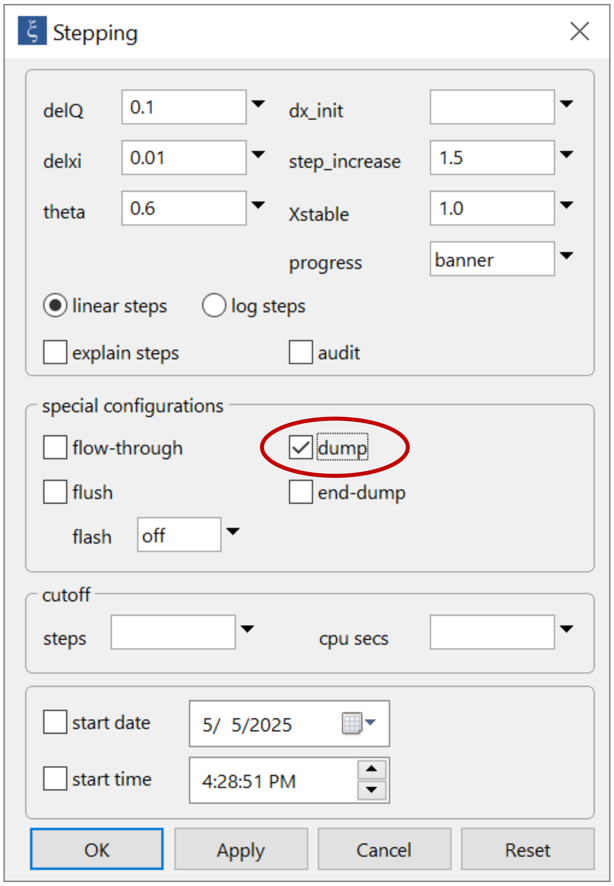
or the equivalent steps

(define an initial system)

```
react 1/2 g Quartz
react 1 g Albite  cutoff = 1/2
```

cause a half gram of quartz and albite to be reacted. The program adds quartz continuously over the path, whereas it reacts albite twice as quickly as quartz over the first half of the path and not at all over the second.

A final feature of **React** is the "dump" option. When this option is set, the program removes any minerals present in the initial system before beginning the reaction path. The option is useful for simulating separation of a fluid from a rock with which it was in equilibrium, before beginning a path. For example, you may want to separate a geothermal fluid from the reservoir rocks before modeling how the fluid could react to form scale. Set the option from the **Config → Stepping…** dialog or by typing `dump`.



delQ 0.1
delxi 0.01
theta 0.6
dx_init
step_increase 1.5
Xstable 1.0
progress banner
● linear steps ○ log steps
□ explain steps □ audit

special configurations
□ flow-through ☑ dump
□ flush □ end-dump
flash off

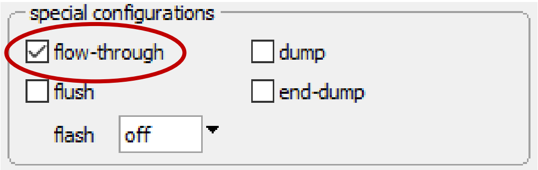
cutoff
steps
cpu secs

□ start date 5/ 5/2025
□ start time 4:28:51 PM

Unchecking “dump” or using the `dump off` command turns off the option.

3.2 Flow-through model

React features a “flow-through” model in which minerals and end members are prevented from dissolving once they precipitate. The model is useful for modeling reaction of a packet of fluid as it migrates through an aquifer. Once minerals form, they are isolated from the packet so that they cannot back-react with the fluid should they become undersaturated farther along the flow path. Implement the flow-through model from the **Config → Stepping…** dialog, or by typing `flow-through`.



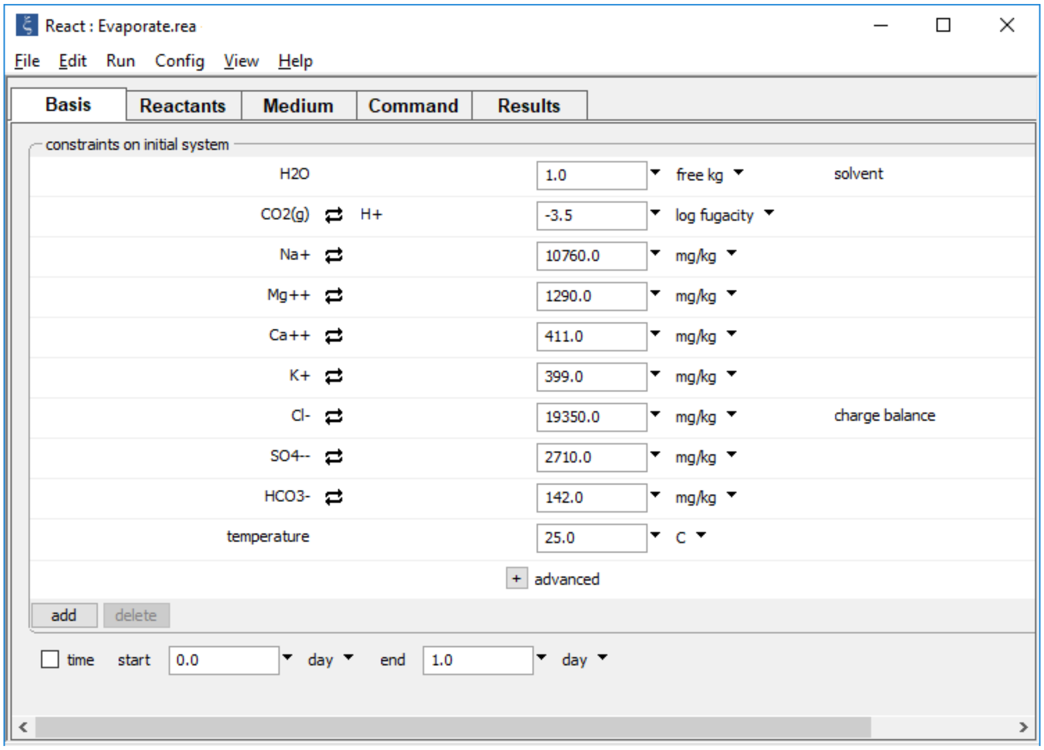
special configurations
☑ flow-through □ dump
□ flush □ end-dump
flash off

To turn the model off, uncheck “flow-through” or type `flow-through off`.

The flow-through model can be applied to a variety of problems in which minerals are unlikely, for physical or chemical reasons, to dissolve once they form. To simulate evaporation of seawater, for example, the model provides a way to account for the settling of minerals to the sea floor. Double-click the “Evaporate.rea” input file to launch **React**.

The Harvie-Møller-Weare activity model is specified to account for the high ionic strength of evaporated seawater. If not already loaded, you could do so by going to **File → Open → Thermo Data…** and selecting “thermo_hmw.tdat”. You can view the contents of the dataset by going to **File → View → .\thermo_hmw.tdat**.

The **Basis** pane

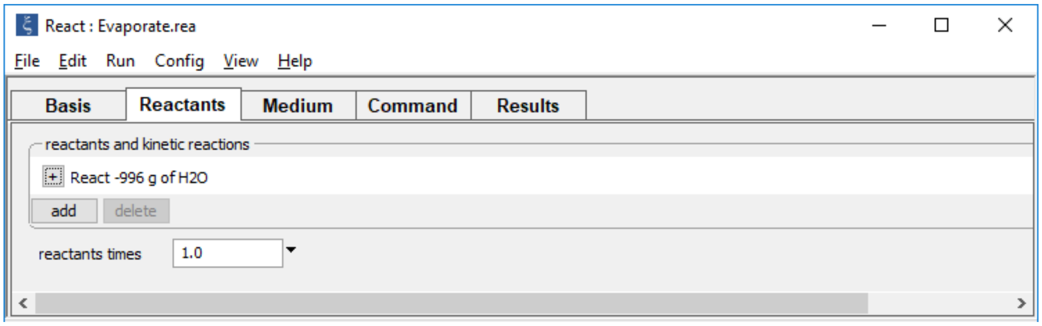


H2O		1.0	free kg	solvent
CO2(g) ⇄	H+	-3.5	log fugacity	
Na+ ⇄		10760.0	mg/kg	
Mg++ ⇄		1290.0	mg/kg	
Ca++ ⇄		411.0	mg/kg	
K+ ⇄		399.0	mg/kg	
Cl- ⇄		19350.0	mg/kg	charge balance
SO4-- ⇄		2710.0	mg/kg	
HCO3- ⇄		142.0	mg/kg	
temperature		25.0	C	

□ time start 0.0 day end 1.0 day

defines the model of surface seawater.

Move to the **Reactants** pane.



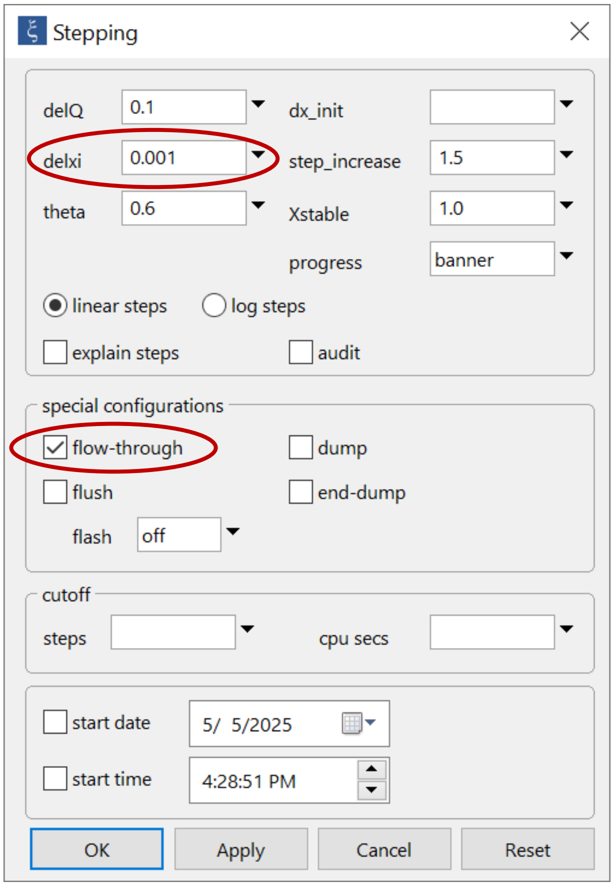
reactants and kinetic reactions

React -996 g of H2O

reactants times 1.0

H2O is set as a simple reactant with a negative mass, so over the reaction path the program removes nearly all the solvent water from the system.

On the **Config → Stepping…** dialog, the value specified for "delxi" (see **Settable variables** in the section **Getting Started with React**) causes the program to take reaction steps $\Delta\xi$that are smaller than the default.



delQ 0.1

dx_init

delxi 0.001

step_increase 1.5

theta 0.6

Xstable 1.0

progress banner

● linear steps ○ log steps

□ explain steps □ audit

special configurations

☑ flow-through □ dump

□ flush □ end-dump

flash off

cutoff

steps

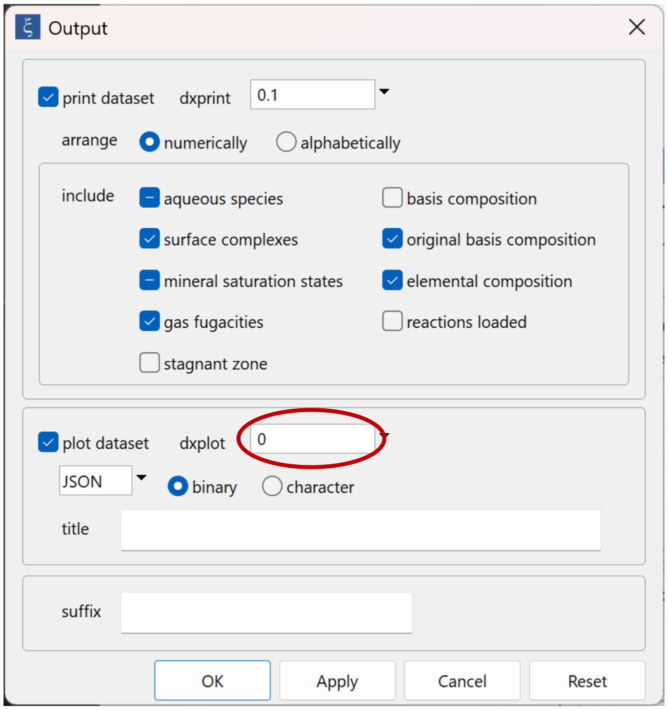
cpu secs

□ start date 5/ 5/2025

□ start time 4:28:51 PM

The flow-through setting, as described above, has been applied to simulate fractionation of the precipitating minerals. Click **OK**.

On the **Config → Output…** dialog, setting “dxplot” to 0 causes each reaction step to be written to “React_plot.gtp”. Click **OK**.



Output

☑ print dataset dxprint 0.1

arrange ● numerically ○ alphabetically

include

⊟ aqueous species ☐ basis composition

☑ surface complexes ☑ original basis composition

⊟ mineral saturation states ☑ elemental composition

☑ gas fugacities ☐ reactions loaded

☐ stagnant zone

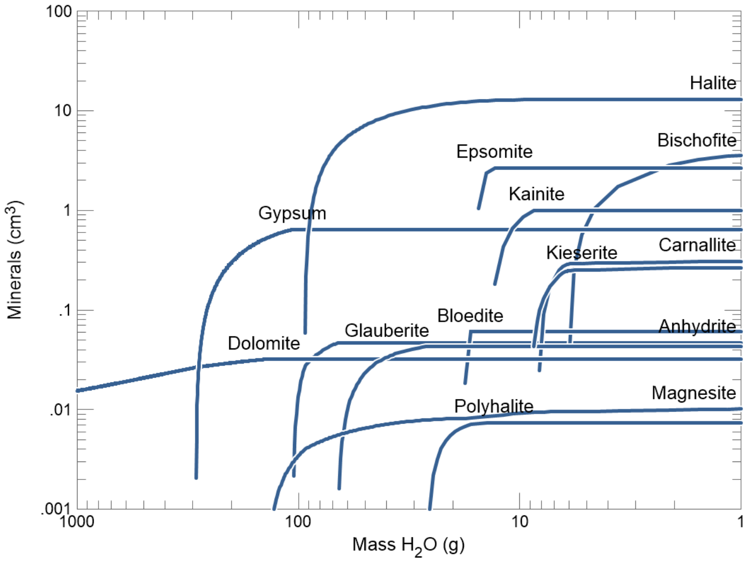
☑ plot dataset dxplot 0

JSON ● binary ○ character

title

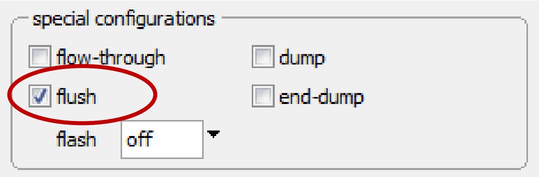
suffix

Note that **React** carries the reaction path to the point of near complete desiccation, at which point it abandons any attempt to further trace the path; the “error” message produced is expected behavior and can be ignored. The calculation results show the sequence of minerals that precipitate during the evaporation of seawater at 25°C.



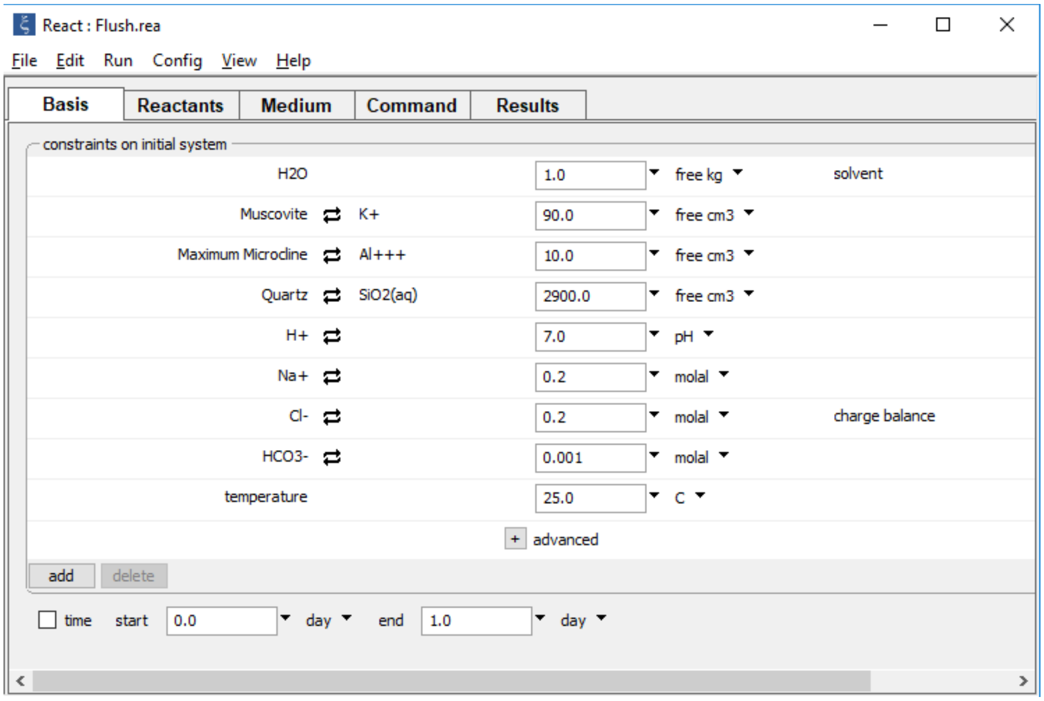
3.3 Flush model

A flush model is a special reaction path in which a fluid, added as a reactant, displaces existing fluid from the system. In contrast to the reference frame of a flow-through model, which follows a migrating fluid, a flush model is traced from the reference frame of the rock through which the fluid migrates. Set a flush model from the **Config → Stepping…** dialog, or type the command `flush`.


special configurations
□ flow-through
□ dump
☑ flush
□ end-dump
flash off

To turn off the option, uncheck “flush” or type the command `flush off`.

To simulate, for example, reaction in a quartzite containing muscovite and microcline as CO_2-charged fluid migrates through it, double-click the “Flush.rea” input file. When **React** opens, go to the **Basis** pane.



constraints on initial system

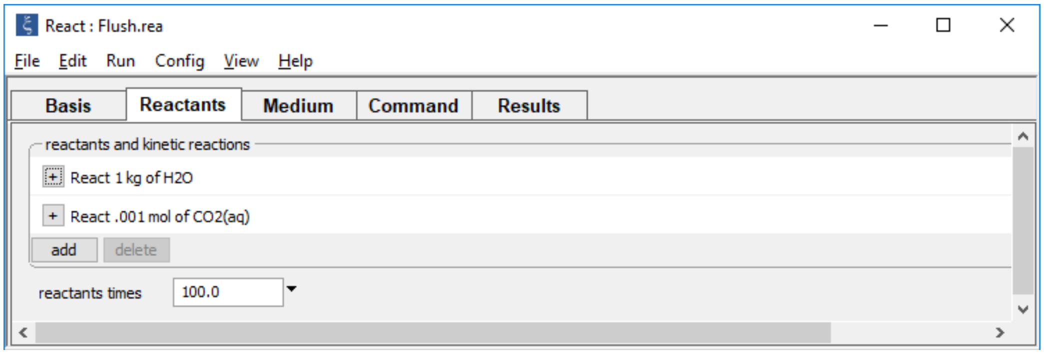
H2O		1.0	free kg	solvent
Muscovite ⇄	K+	90.0	free cm3	
Maximum Microcline ⇄	Al+++	10.0	free cm3	
Quartz ⇄	SiO2(aq)	2900.0	free cm3	
H+ ⇄		7.0	pH	
Na+ ⇄		0.2	molal	
Cl- ⇄		0.2	molal	charge balance
HCO3- ⇄		0.001	molal	
temperature		25.0	C	

+ advanced

□ time start 0.0 day end 1.0 day

The input describes the initial mineralogy and pore water fluid of the quartzite.

Move to the **Reactants** pane,



reactants and kinetic reactions

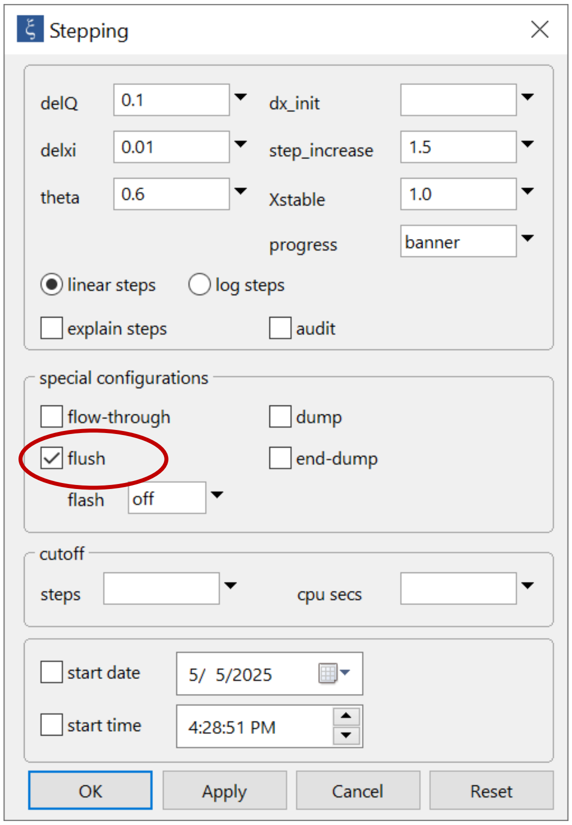
+ React 1 kg of H2O

+ React .001 mol of CO2(aq)

reactants times 100.0

which sets the CO_2-charged water to be reacted into the system. Because of the value set for “reactants times”, 100 kg of fluid will be reacted.

On the **Config → Stepping…** dialog,



Stepping

delQ 0.1 — dx_init

delxi 0.01 — step_increase 1.5

theta 0.6 — Xstable 1.0

progress banner

● linear steps ○ log steps

□ explain steps □ audit

special configurations

□ flow-through □ dump

☑ flush □ end-dump

flash off

cutoff

steps — cpu secs

□ start date 5/ 5/2025

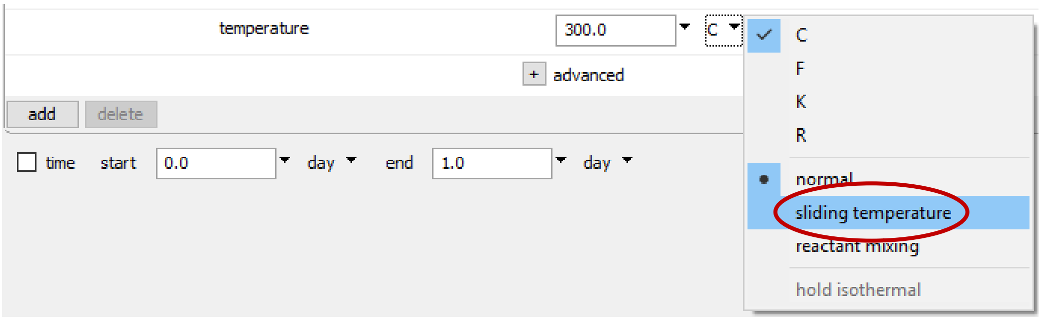
□ start time 4:28:51 PM

the “flush” option has been selected. As a result, the reactant fluid will displace an equal amount of fluid (in terms of water mass) from the system.

3.4 Polythermal reaction paths

React traces polythermal reactions in one of three ways. It can vary temperature linearly from an initial to a final value over a reaction path—such a calculation is called a “sliding temperature” path. The program can also calculate temperature from the mixing of reactants at one temperature into an initial system of another, assuming approximate values for heat capacity. This second option is called a “polythermal mixing” or “quench” model. Finally, the program can account for heat produced or consumed internally in the domain, as described in the **Internal heat production** section of the ***GWB Reactive Transport Modeling Guide***. A heat source can be applied in a “normal” path in which you only set the initial temperature of the system, or in conjunction with a “polythermal mixing” path, but not with the “sliding temperature” option. A heat source traced in combination with a vapor phase constitutes a boiling model (see **Vapor phase** and **Boiling and condensation** for more information). Polythermal paths can trace the simple heating or cooling of a chemical system, or can be combined with other **React** features, such as reactants, kinetic rate laws, and isotopic fractionation.

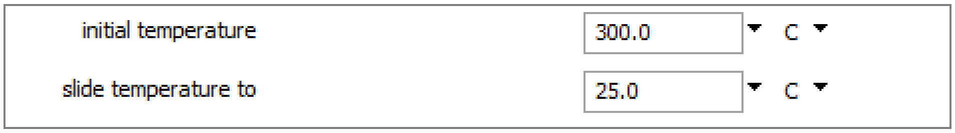
To set a sliding temperature model, specify initial and final temperatures in the range of the thermo data, typically 0°C–300°C. For example, to simulate cooling, set the system's composition in the usual way and enter the beginning and ending temperatures: click the pulldown next to the unit for temperature on the **Basis** pane, then choose "sliding temperature".



temperature 300.0 C

□ time start 0.0 day end 1.0 day

Upon selecting this option, "temperature" will be relabeled "initial temperature" and a new field labeled "slide temperature to" will appear below it.



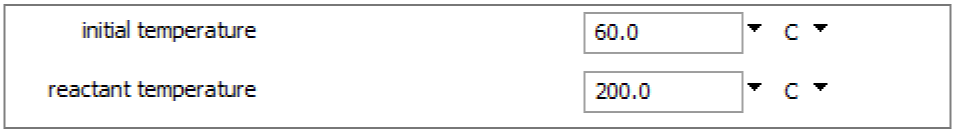
initial temperature 300.0 C

slide temperature to 25.0 C

In this case, the system begins at 300°C and cools to 25°C. Alternatively, set the sliding temperature path with the command

```
temperature initial = 300, final = 25
```

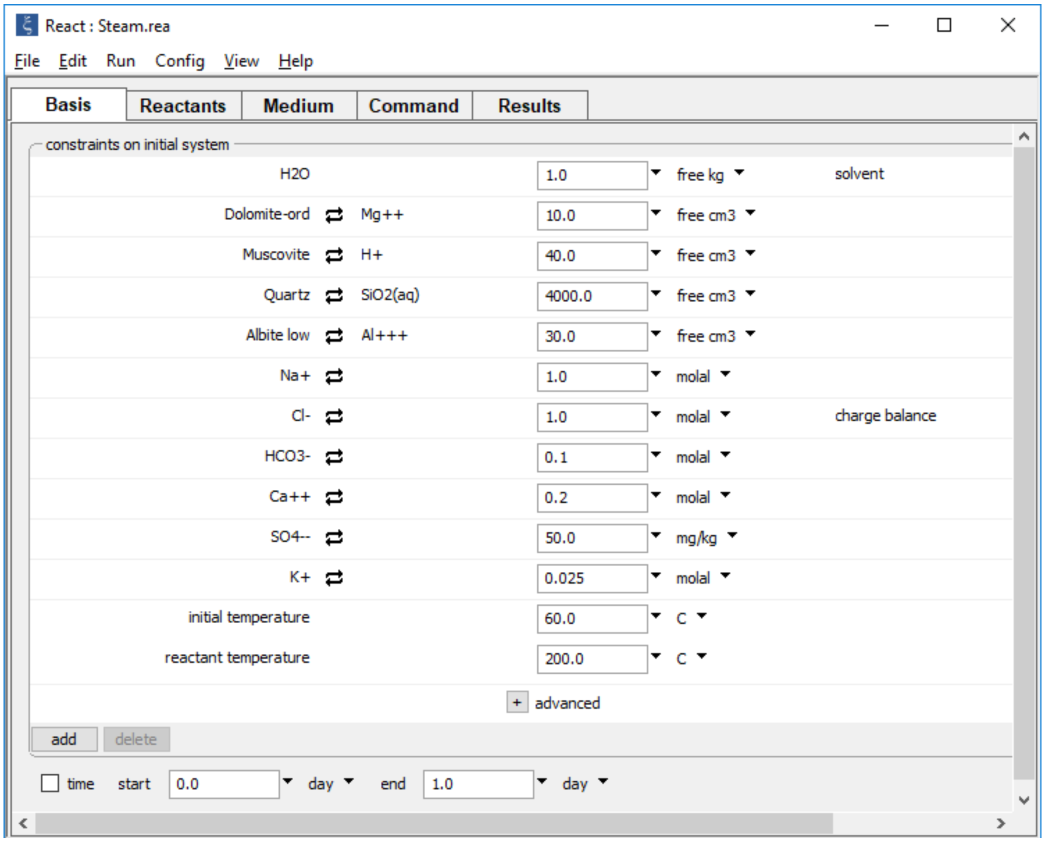
In a polythermal mixing model, set the temperatures for the reactants and the initial system. Over the path, the program calculates the temperature from the mass of reactant that has been added to the system, using the approximate heat capacities for fluids (water) and minerals (rock) stored in variables "cpw" and "cpr" (set on the **Medium** pane; see Settable variables in the section **Getting Started with React**). To set the polythermal mixing model, choose "reactant mixing" from the pulldown, then set a value for the "reactant temperature" field that appears.



initial temperature 60.0 C

reactant temperature 200.0 C

In the following example, the program is applied to model a steam flood of a hypothetical petroleum reservoir composed of quartz, illite (muscovite), albite, and (ordered) dolomite. The “steam” here is actually hot water containing CO_2. Double-click on the “Steam.rea” input file. When **React** opens, go to the **Basis** pane,



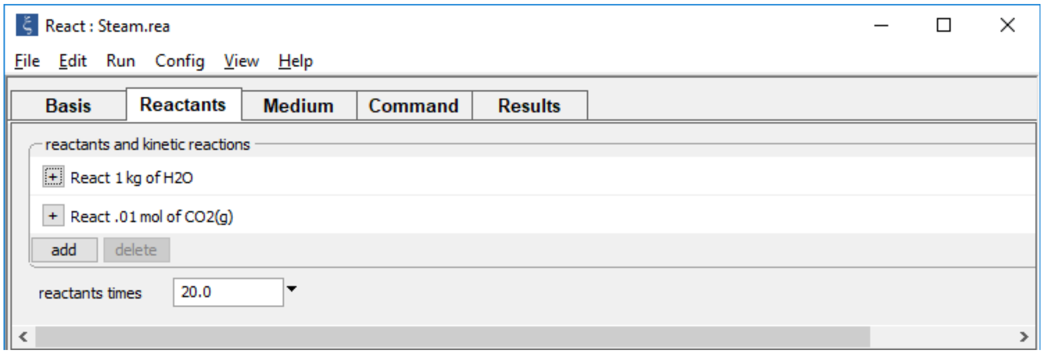
constraints on initial system

H2O			1.0	free kg	solvent
Dolomite-ord	⇄	Mg++	10.0	free cm3	
Muscovite	⇄	H+	40.0	free cm3	
Quartz	⇄	SiO2(aq)	4000.0	free cm3	
Albite low	⇄	Al+++	30.0	free cm3	
Na+	⇄		1.0	molal	
Cl-	⇄		1.0	molal	charge balance
HCO3-	⇄		0.1	molal	
Ca++	⇄		0.2	molal	
SO4--	⇄		50.0	mg/kg	
K+	⇄		0.025	molal	
initial temperature			60.0	C	
reactant temperature			200.0	C	

□ time start 0.0 day end 1.0 day

which sets the composition of the formation and initial fluid, as well as the temperature of the reactant fluid and initial system.

Move to the **Reactants** pane,



reactants and kinetic reactions

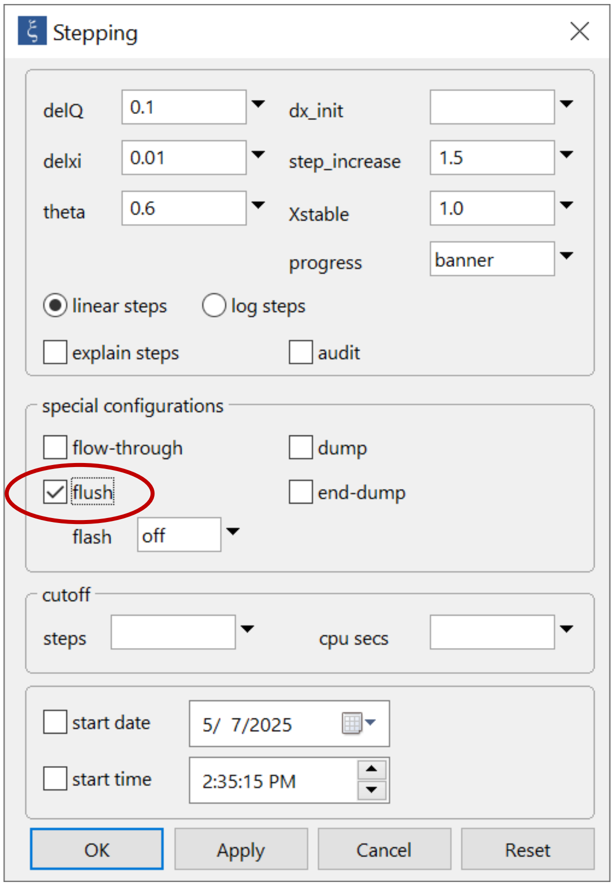
+ React 1 kg of H2O

+ React .01 mol of CO2(g)

reactants times 20.0

where the composition of the reactant fluid is defined.

On the **Config → Stepping…** dialog,



Stepping

delQ 0.1
delxi 0.01
theta 0.6
dx_init
step_increase 1.5
Xstable 1.0
progress banner

● linear steps ○ log steps

□ explain steps □ audit

special configurations

□ flow-through □ dump

☑ flush □ end-dump

flash off

cutoff

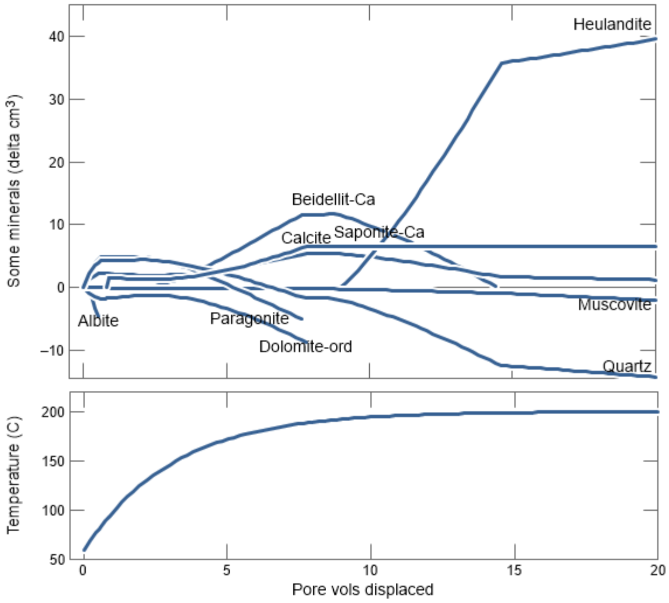
steps
cpu secs

□ start date 5/ 7/2025

□ start time 2:35:15 PM

the “flush” option has been enabled. This setting, combined with the value set for “reactants times” on the **Reactants** pane, causes the hot reactant fluid to pass through the initially cooler formation 20 times.

The figure below shows the calculation results.

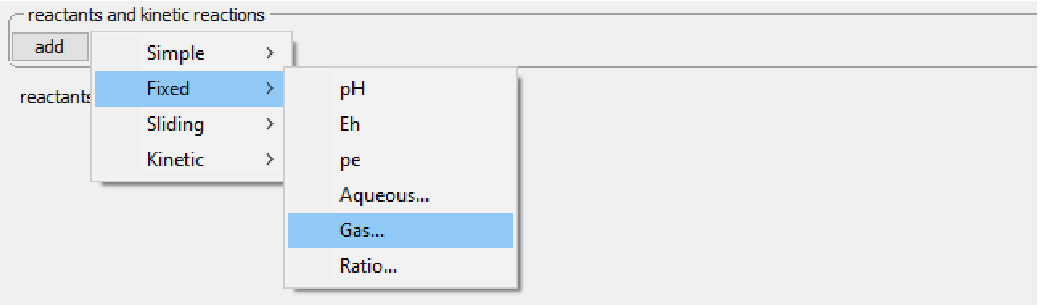


The reservoir, initially at 60°C, is progressively heated as the 200°C flood passes through. The horizontal axis represents the number of times the 1 kg of pore water in the system is replaced.

3.5 Buffered paths

React can trace reaction paths in which one or more species activities or gas fugacities are held constant. Values of pH, Eh, and pe may also be fixed. Fixed activity and fixed fugacity paths provide a means of modeling systems buffered in nature by contact with a gas reservoir such as the atmosphere, or in the laboratory by a pH-stat or similar device.

To model a buffered system, fix an activity or fugacity from the **Reactants** pane: click on add → **Fixed**, then select the desired option. Selecting **Aqueous…** or **Gas…** will bring up a pulldown of species to choose from.



You can also set a buffered path by typing a command such as

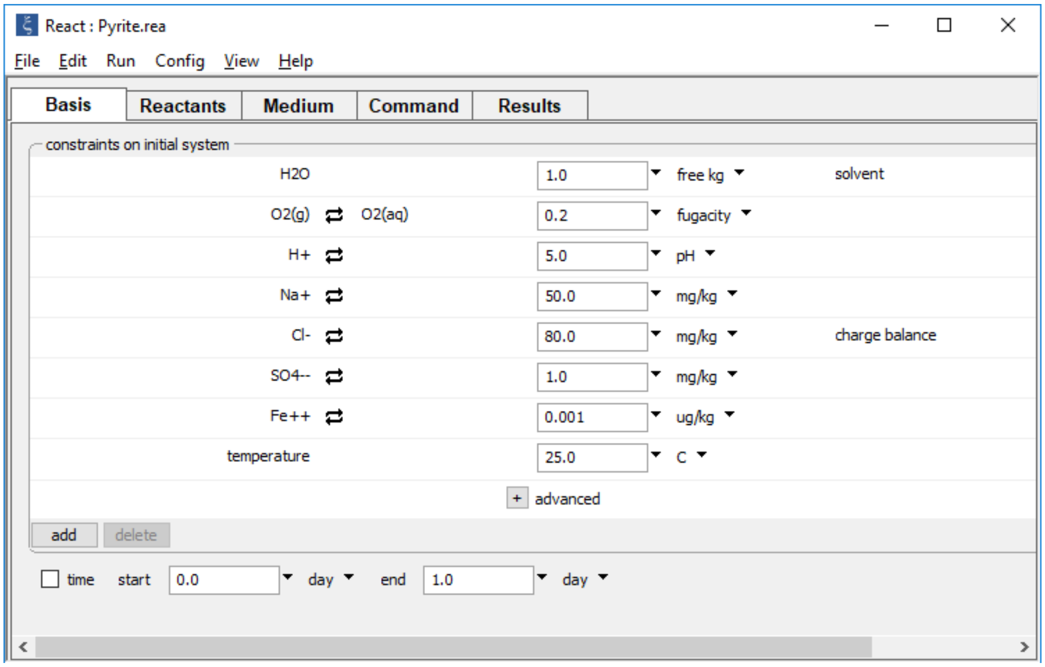
```
fix f CO2(g)
```

or

```
fix pH
```

In these cases, the program maintains the fluid's CO_2 fugacity and pH at their initial values.

To simulate pyrite dissolving into a water that remains oxidized by contact with the atmosphere, double-click on the "Pyrite.rea" input file. When **React** opens, go to the **Basis** pane

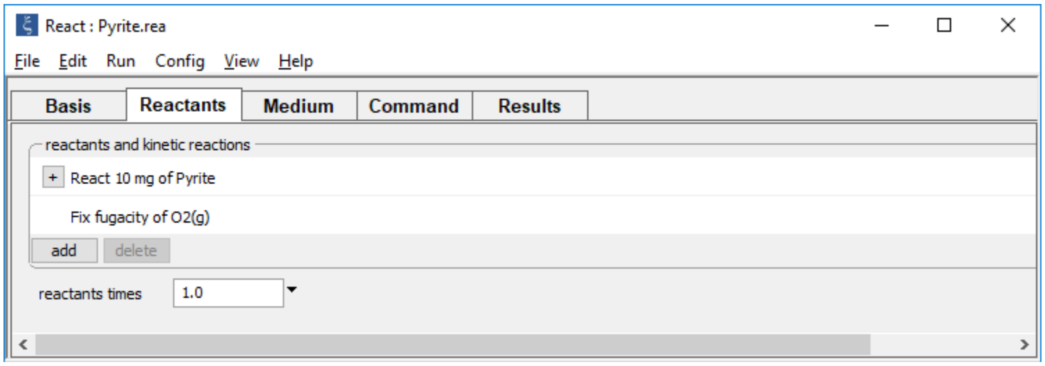


H2O		1.0	free kg	solvent
O2(g) ⇄	O2(aq)	0.2	fugacity	
H+ ⇄		5.0	pH	
Na+ ⇄		50.0	mg/kg	
Cl- ⇄		80.0	mg/kg	charge balance
SO4-- ⇄		1.0	mg/kg	
Fe++ ⇄		0.001	ug/kg	
temperature		25.0	C	

□ time start 0.0 day end 1.0 day

to see the initial composition, which is set to be in equilibrium with O2(g) in the atmosphere.

On the **Reactants** pane,



reactants and kinetic reactions

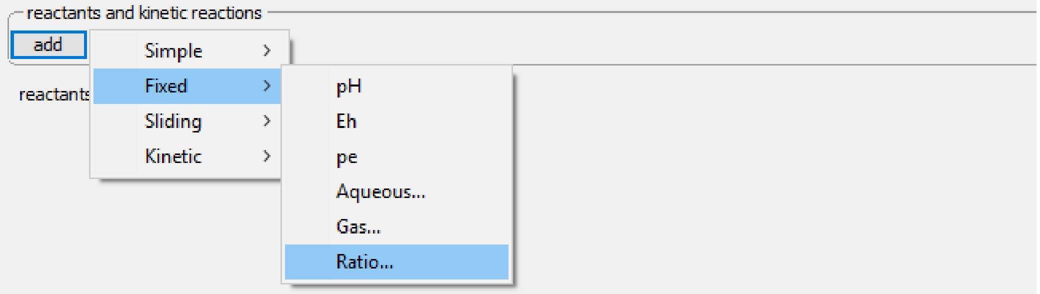
React 10 mg of Pyrite

Fix fugacity of O2(g)

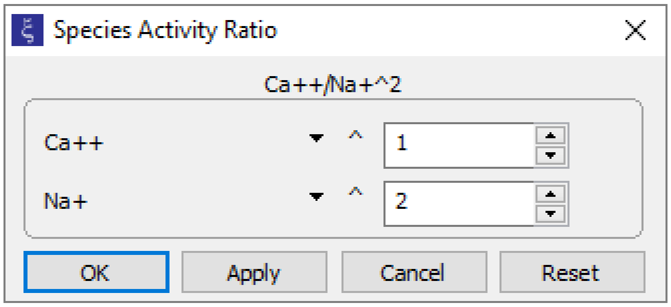
reactants times 1.0

the fixed O2(g) fugacity buffer ensures that the water remains oxidized as Pyrite is titrated into the fluid.

To hold an activity ratio constant over the course of a reaction path, “fix” the ratio on the **Reactants** pane by clicking on add → **Fixed** → **Ratio…**



then choosing species, and their powers, for the numerator and denominator. For example, the setting below



Ca++/Na+^2

Ca++ ^ 1

Na+ ^ 2

or the equivalent command

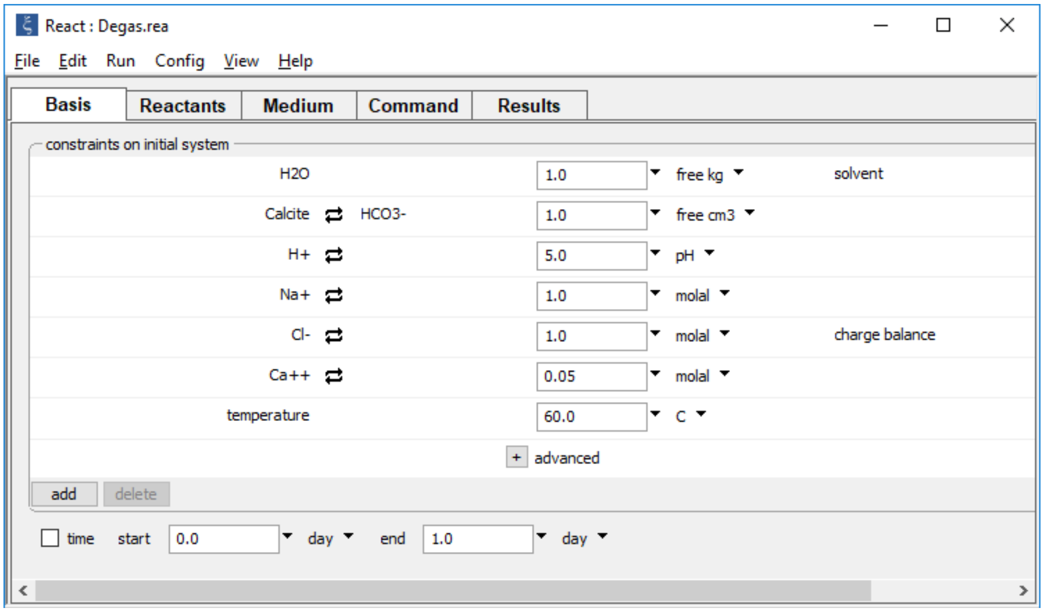
```
fix ratio Ca++/Na+^2
```

holds invariant the activity ratio for calcium and sodium ions to reflect a simple ion exchange buffer.

3.6 Sliding activity and fugacity

React can not only hold species activities and gas fugacities constant, but can vary one or more of these values over a reaction path. Calculations of this type are known as sliding activity paths and sliding fugacity paths. To set a sliding activity or fugacity, move to the **Reactants** pane and choose add → **Sliding**, or use the `slide` command.

To calculate the effect at 60°C of CO_2 escaping from a groundwater in equilibrium with calcite, for example, double-click on the “Degas.rea” input file. The **Basis** pane



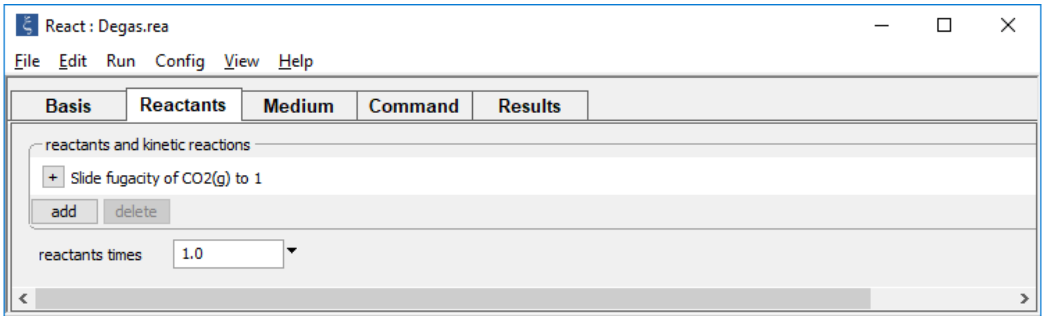
constraints on initial system

H2O		1.0	free kg	solvent
Calcite	⇄ HCO3-	1.0	free cm3	
H+	⇄	5.0	pH	
Na+	⇄	1.0	molal	
Cl-	⇄	1.0	molal	charge balance
Ca++	⇄	0.05	molal	
temperature		60.0	C	

□ time start 0.0 day end 1.0 day

defines a groundwater in equilibrium with calcite.

Move to the **Reactants** pane


React : Degas.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete

reactants and kinetic reactions

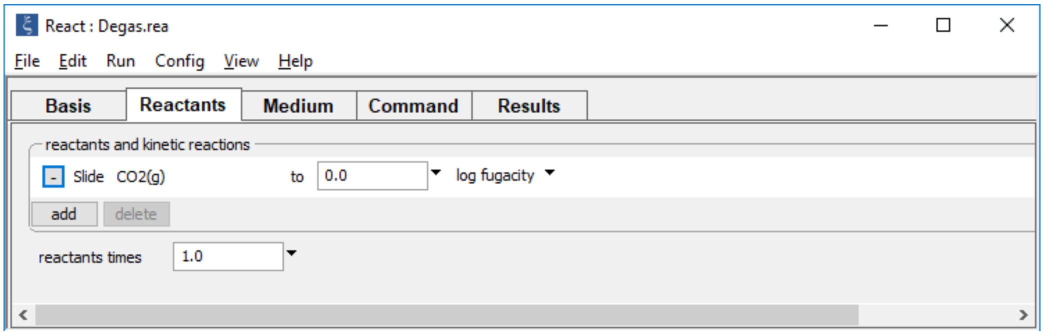
+ Slide fugacity of CO2(g) to 1

reactants times 1.0

to set the sliding fugacity path, or use the command

```
slide f CO2(g) to 1
```

The CO_2 fugacity in this example slides linearly from its initial value of about 35 to 1 at the end of the path. If, on the other hand, you had specified log units on the **Reactants** pane


React : Degas.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete

reactants and kinetic reactions

- Slide CO2(g) to 0.0 log fugacity

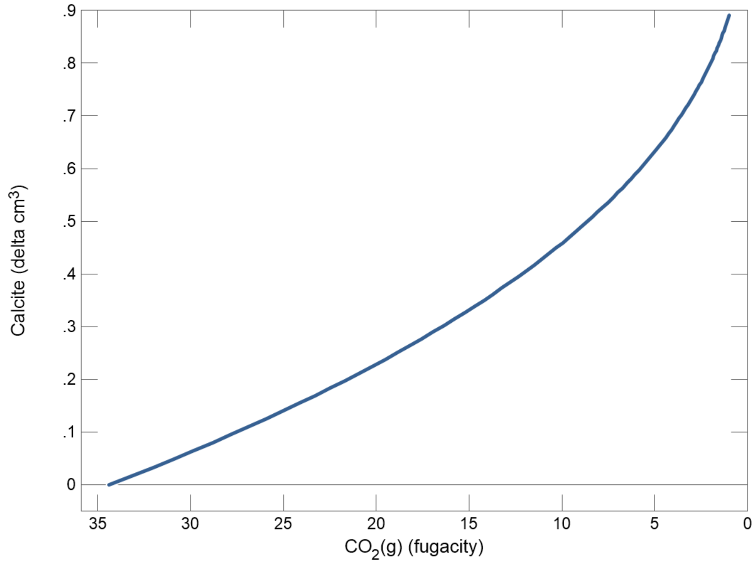
reactants times 1.0

or with the command

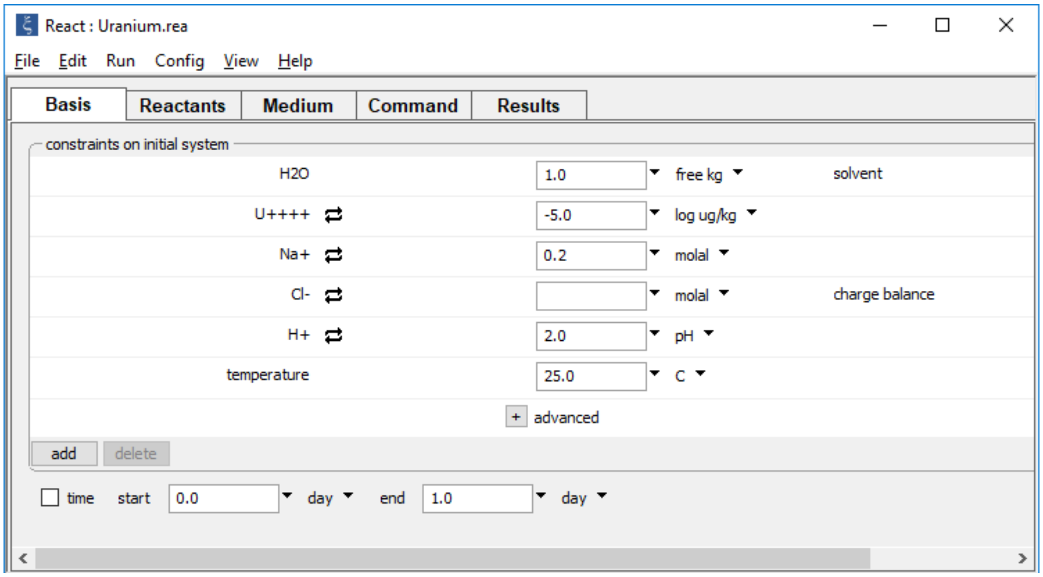
```
slide log f CO2(g) to 0
```

the program would have varied the *logarithm* of the CO_2 fugacity linearly from its initial to final value.

The figure below shows calcite precipitated over the course of CO_2 degassing at 60°C, as simulated by the sliding fugacity path.



As a second example, calculate the species distribution of uranium(IV) versus pH by sliding pH from 2 to 12. Double-click on the “Uranium.rea” input file and, when **React** opens, move to the **Basis** pane,

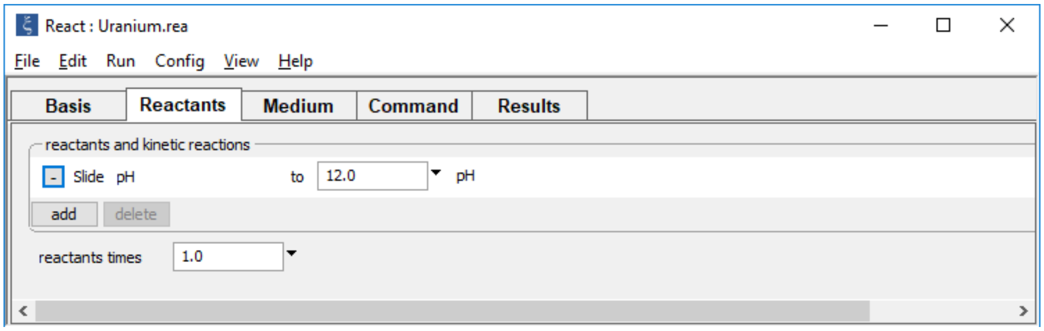

React : Uranium.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
constraints on initial system
+ advanced
add delete

H2O	1.0	free kg	solvent
U++++	-5.0	log ug/kg	
Na+	0.2	molal	
Cl-		molal	charge balance
H+	2.0	pH	
temperature	25.0	C	

□ time start 0.0 day end 1.0 day

which sets up the initial system: a fluid containing uranium with a pH of 2. Note that the program maintains electrical neutrality over the path by varying the Cl^- concentration.

On the **Reactants** pane,



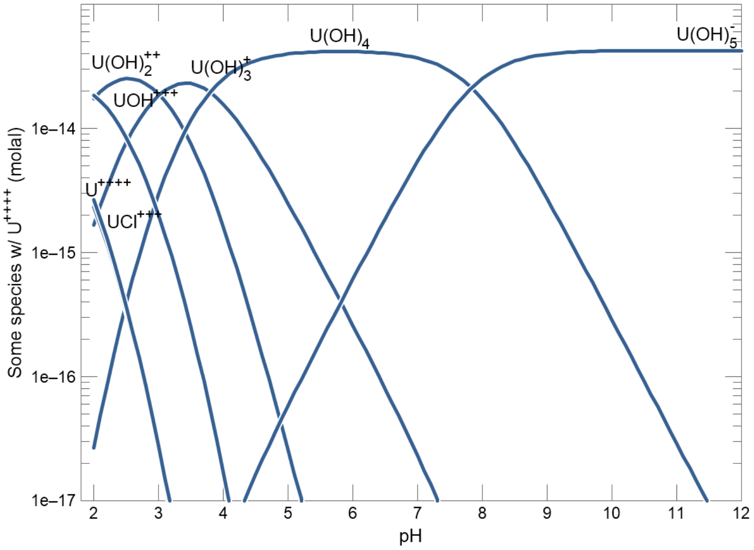
reactants and kinetic reactions

Slide pH to 12.0 pH

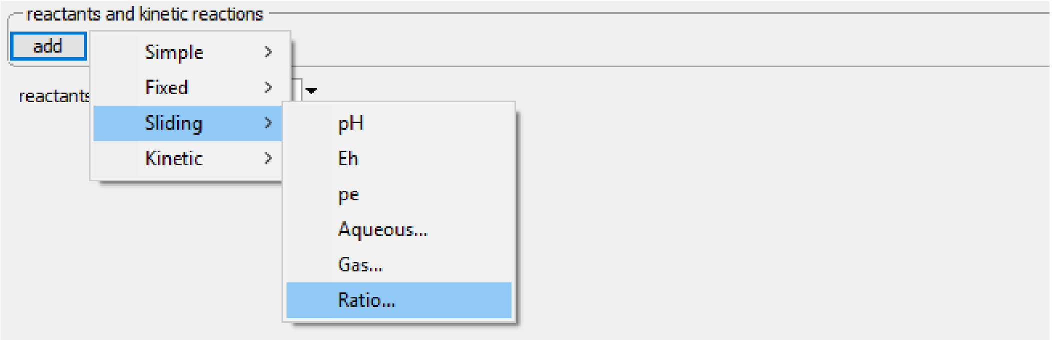
reactants times 1.0

the sliding pH path is implemented. The program will vary pH from its initial value of 2 to the target value of 12.

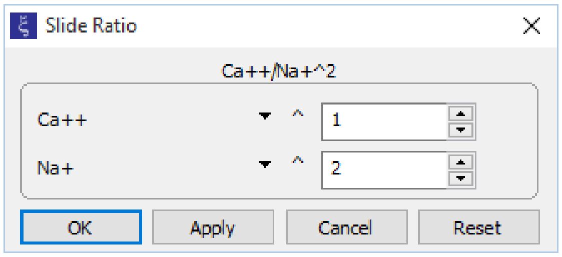
The figure below shows the distribution of uranium(IV) species as a function of pH, calculated as a sliding activity path.



You can vary an activity ratio over the course of a path in a similar fashion. Choose on the **Reactants** pane add → **Sliding** → **Ratio…**



then set the species in the ratio and their powers.



Ca++/Na+^2

Ca++ ^ 1

Na+ ^ 2

Alternatively, use the command

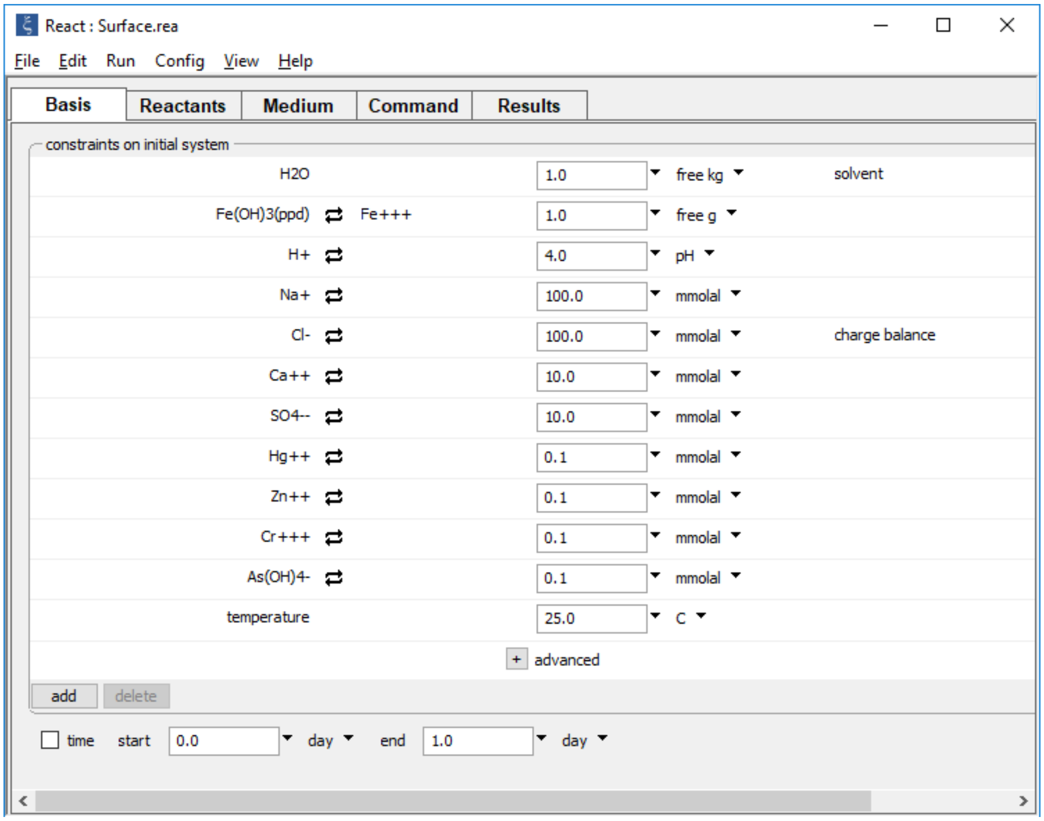
```
slide Ca++/Na+^2 to .01
```

3.7 Surface complexation

The examples in this section show how to incorporate surface complexation into reaction modeling with **React**.

3.7.1 Two-layer model

In a first example, we compute the effect of pH on the surface complexation of a variety of dissolved species, according to the two-layer model. Double-click on the “Surface.rea” input file and when **React** opens go to the **Basis** pane to see the initial fluid composition

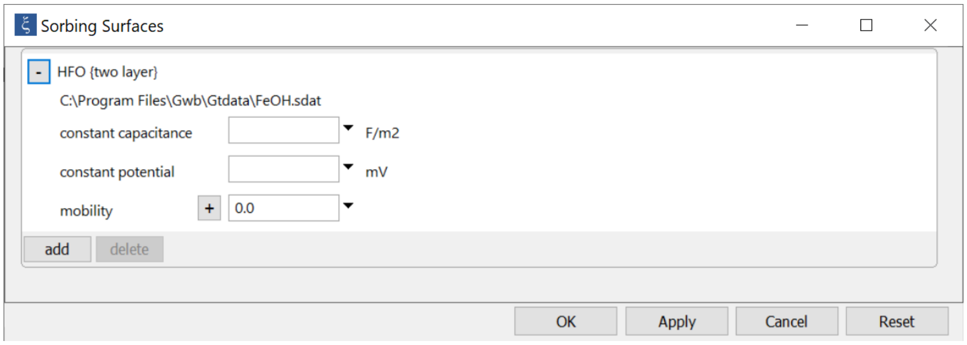


H2O		1.0	free kg	solvent
Fe(OH)3(ppd) ⇄	Fe+++	1.0	free g	
H+ ⇄		4.0	pH	
Na+ ⇄		100.0	mmolal	
Cl- ⇄		100.0	mmolal	charge balance
Ca++ ⇄		10.0	mmolal	
SO4-- ⇄		10.0	mmolal	
Hg++ ⇄		0.1	mmolal	
Zn++ ⇄		0.1	mmolal	
Cr+++ ⇄		0.1	mmolal	
As(OH)4- ⇄		0.1	mmolal	
temperature		25.0	C	

□ time start 0.0 day end 1.0 day

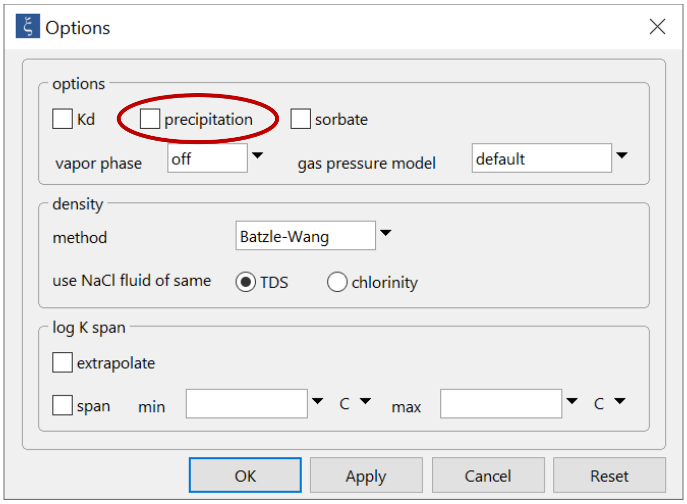
Here, Fe(OH)3(ppd) serves as a proxy for hydrous ferric oxide.

Load the Dzombak and Morel dataset for ion complexation with hydrous ferric oxide by going to **File → Open → Sorbing Surfaces…** and adding the “FeOH.sdat” dataset.



HFO (two layer)
C:\Program Files\Gwb\Gtdata\FeOH.sdat
constant capacitance F/m2
constant potential mV
mobility 0.0

On the **Config → Options…** dialog,



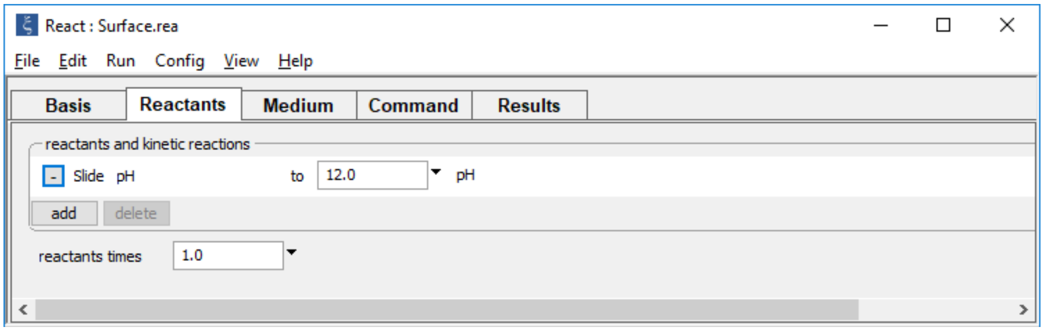
options
□ Kd □ precipitation □ sorbate
vapor phase off gas pressure model default

density
method Batzle-Wang
use NaCl fluid of same ● TDS ○ chlorinity

log K span
□ extrapolate
□ span min C max C

the “precipitation” option has been deselected. As a result, the more stable ferric oxide minerals are prevented from forming.

The **Reactants** pane



React : Surface.rea

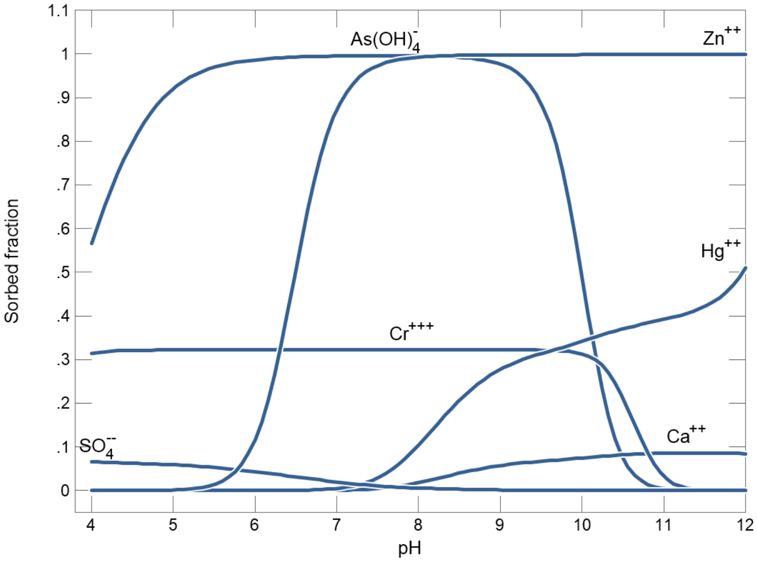
reactants and kinetic reactions

- Slide pH to 12.0 pH

reactants times 1.0

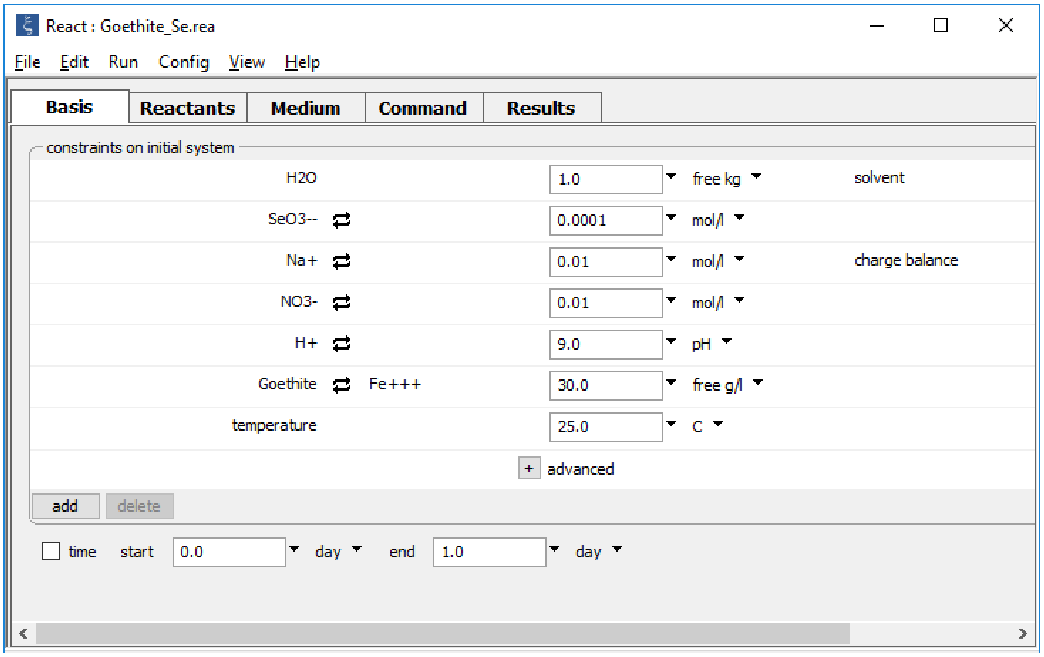
sets up a sliding pH path.

Running the calculation, the results show sorption of various components from solution onto a hydrous ferric hydroxide surface, as a function of pH



3.7.2 Triple-layer model

This example shows how pH controls the sorption of selenite onto goethite, according to the triple-layer model. Double-click on the “Goethite_Se.rea” input file. When **React** opens, go to the **Basis** pane to see the initial fluid composition. The background electrolyte is 0.01 M $NaNO_3$ at pH 9; it contains 0.1 mmol/l selenite and is in equilibrium with goethite.



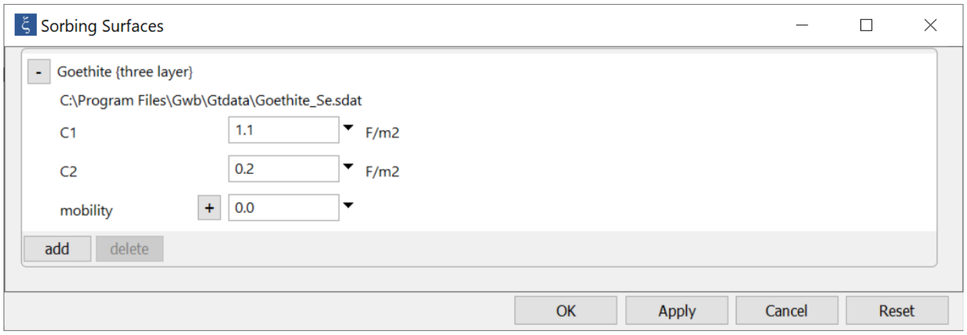
constraints on initial system

H2O		1.0	free kg	solvent
SeO3--		0.0001	mol/l	
Na+		0.01	mol/l	charge balance
NO3-		0.01	mol/l	
H+		9.0	pH	
Goethite	Fe+++	30.0	free g/l	
temperature		25.0	C	

+ advanced

□ time start 0.0 day end 1.0 day

Go to **File → Open → Sorbing Surfaces…** to load dataset “Goethite_Se.sdat”, which describes selenite complexation with goethite

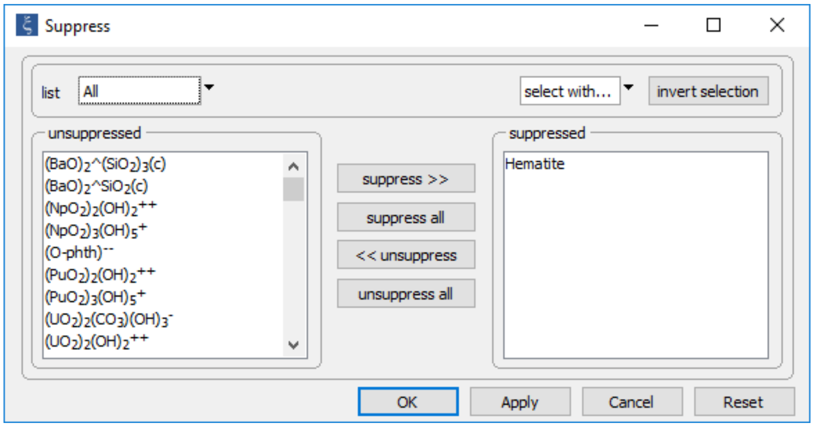


- Goethite (three layer)

C:\Program Files\Gwb\Gtdata\Goethite_Se.sdat

C1	1.1	F/m2
C2	0.2	F/m2
mobility	0.0	

On the **Config → Suppress…** dialog,



list: All

unsuppressed

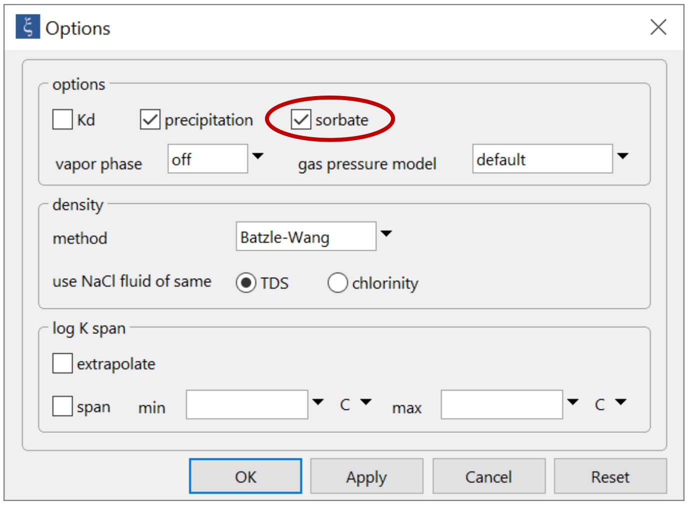
$(BaO)_2$^$(SiO_2)_3$(c)
$(BaO)_2$^SiO_2(c)
$(NpO_2)_2(OH)_2^{++}$
$(NpO_2)_3(OH)_5^{+}$
(O-phth)$^{--}$
$(PuO_2)_2(OH)_2^{++}$
$(PuO_2)_3(OH)_5^{+}$
$(UO_2)_2(CO_3)(OH)_3^{-}$
$(UO_2)_2(OH)_2^{++}$

suppressed

Hematite

the stable ferric oxide mineral Hematite is suppressed to prevent it from forming.

On the **Config → Options** dialog, “sorbate” option is checked, specifying the SeO3^{--} concentration in the basis entry represents the combined mass of dissolved and sorbed selenite



options

□ Kd ☑ precipitation ☑ sorbate

vapor phase: off gas pressure model: default

density

method: Batzle-Wang

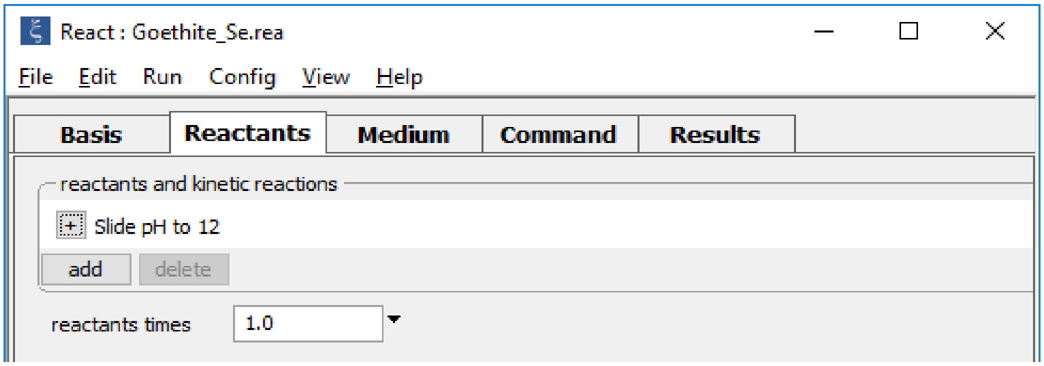
use NaCl fluid of same ● TDS ○ chlorinity

log K span

□ extrapolate

□ span min C max C

The **Reactants** pane

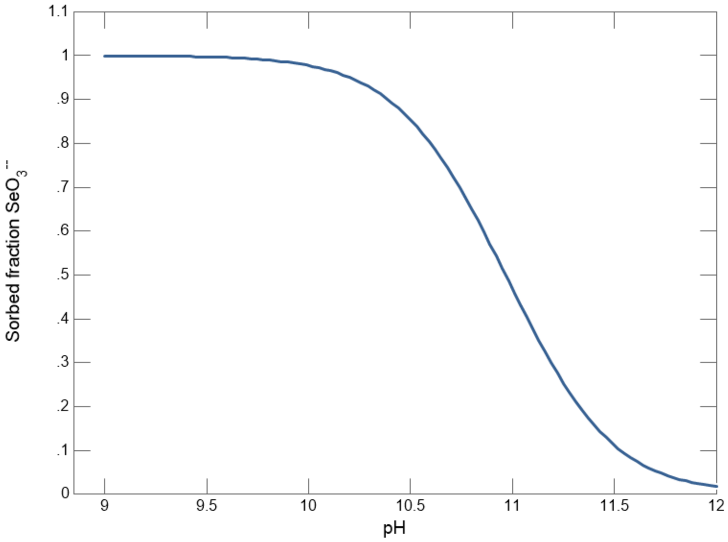


reactants and kinetic reactions

⊞ Slide pH to 12

reactants times 1.0

sets up the sliding pH path. Running the simulation shows that selenite sorption decreases markedly as the pH increases.

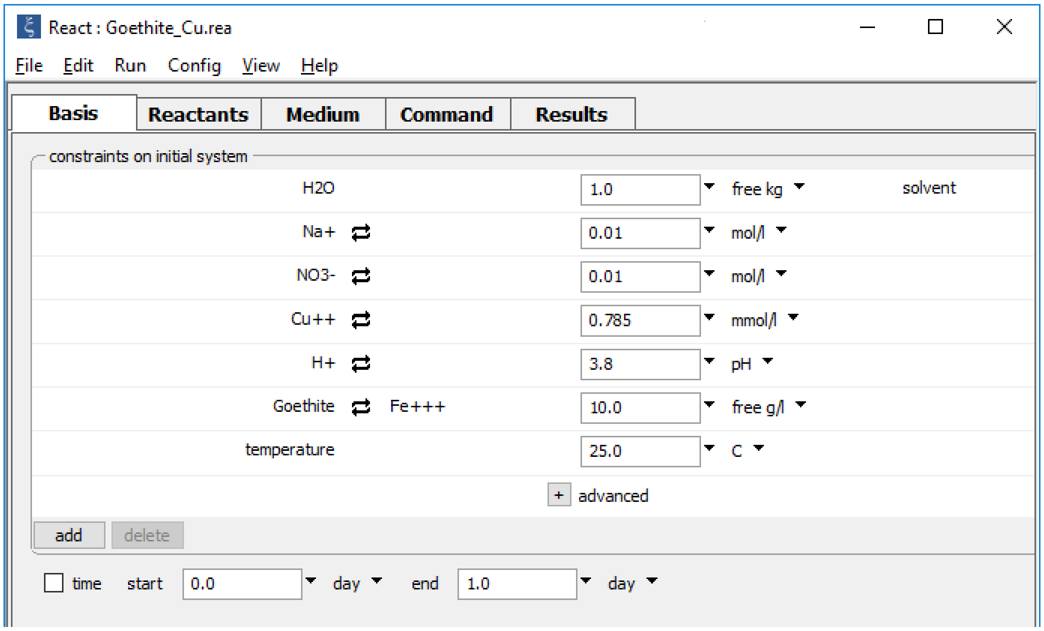


You can perform similar experiments using input files “Goethite_NaCl.rea” and “Goethite_NaClO4.rea”.

3.7.3 CD-MUSIC method

In a final example, we consider the effect of pH on the sorption of cupric copper onto goethite in the presence of 0.01M $NaNO_3$, according to the CD-MUSIC surface complexation model.

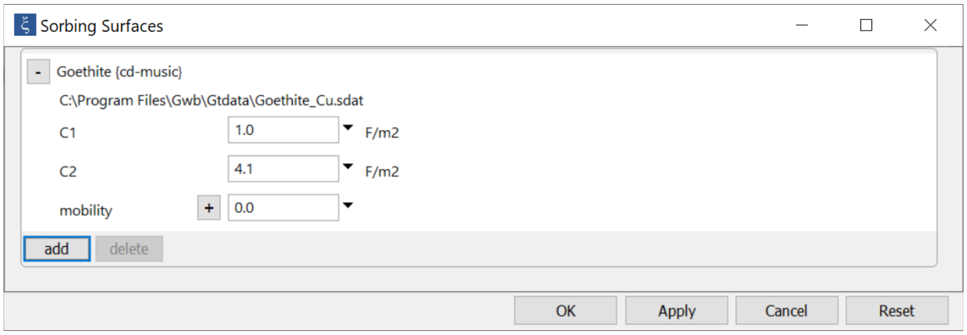
Start React by double-clicking on file “Goethite_Cu.rea”. When the program opens, go to the **Basis** pane to see the initial condition, a fluid at pH 3.8 containing 0.785mM copper in equilibrium with 10 g/l Goethite.


React : Goethite_Cu.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
constraints on initial system
+ advanced
add delete

H2O		1.0	free kg	solvent
Na+ ⇄		0.01	mol/l	
NO3- ⇄		0.01	mol/l	
Cu++ ⇄		0.785	mmol/l	
H+ ⇄		3.8	pH	
Goethite ⇄	Fe+++	10.0	free g/l	
temperature		25.0	C	

□ time start 0.0 day end 1.0 day

Load dataset “Goethite_Cu.sdat” describing complexation according to the CD-MUSIC model by going to **File → Open → Sorbing Surfaces…**

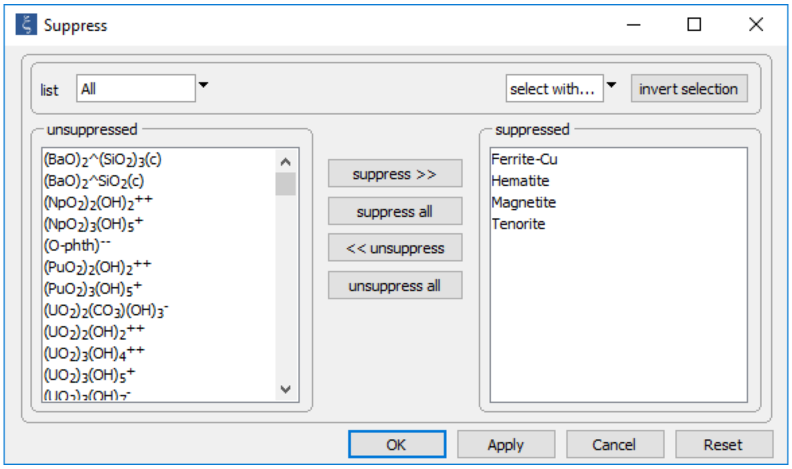


- Goethite (cd-music)

C:\Program Files\Gwb\Gtdata\Goethite_Cu.sdat

C1	1.0	F/m2
C2	4.1	F/m2
mobility +	0.0	

On the **Config → Suppress…** dialog



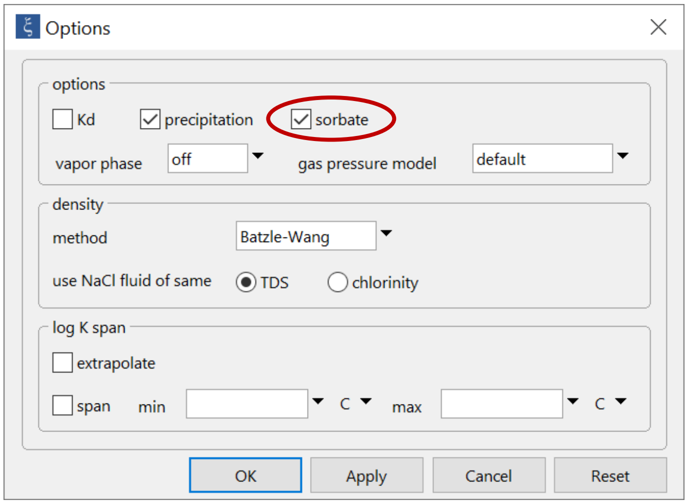
list: All
select with…
invert selection

unsuppressed:
$(BaO)_2$^$(SiO_2)_3(c)$
$(BaO)_2$^$SiO_2(c)$
$(NpO_2)_2(OH)_2^{++}$
$(NpO_2)_3(OH)_5^{+}$
$(O\text{-}phth)^{--}$
$(PuO_2)_2(OH)_2^{++}$
$(PuO_2)_3(OH)_5^{+}$
$(UO_2)_2(CO_3)(OH)_3^{-}$
$(UO_2)_2(OH)_2^{++}$
$(UO_2)_3(OH)_4^{++}$
$(UO_2)_3(OH)_5^{+}$
$(UO_2)_3(OH)_7^{-}$

suppressed:
Ferrite-Cu
Hematite
Magnetite
Tenorite

several relatively stable copper and ferric oxide minerals are suppressed to prevent them from forming.

On the **Config → Options** dialog, the “sorbate” option is checked. With this option, the concentration of Cu^{++} in the basis entry represents the combined mass of dissolved and sorbed cupric copper species

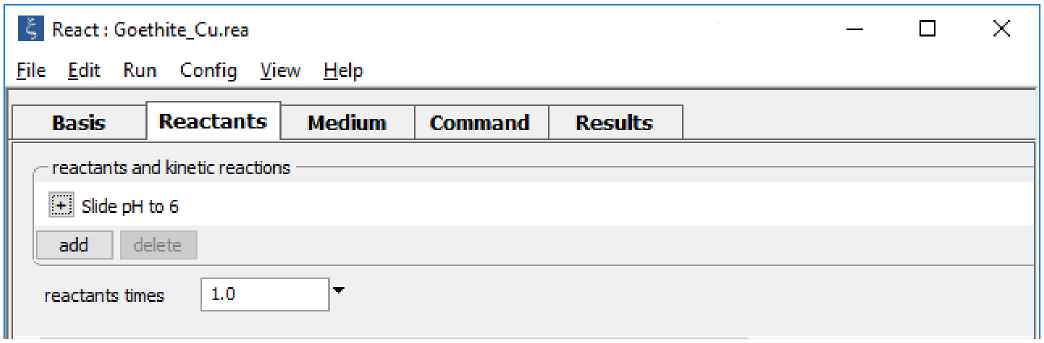


options
□ Kd ☑ precipitation ☑ sorbate
vapor phase: off
gas pressure model: default

density
method: Batzle-Wang
use NaCl fluid of same ● TDS ○ chlorinity

log K span
□ extrapolate
□ span min C max C

The **Reactants** pane

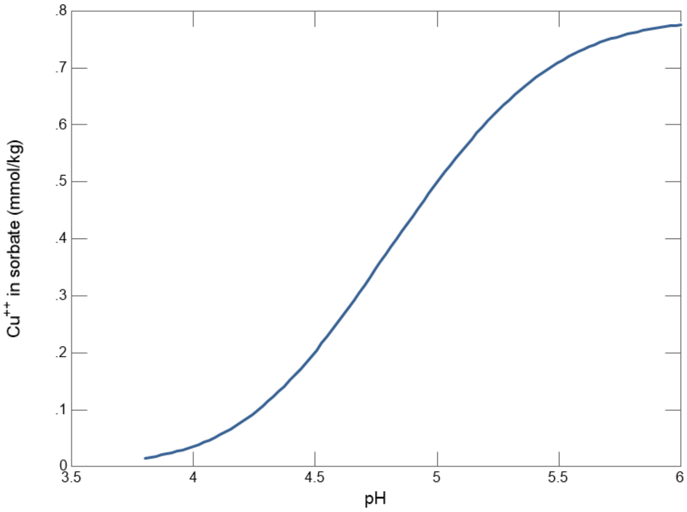


reactants and kinetic reactions

Slide pH to 6

reactants times 1.0

sets a sliding pH path. In the calculation results, copper sorption on goethite increases with pH

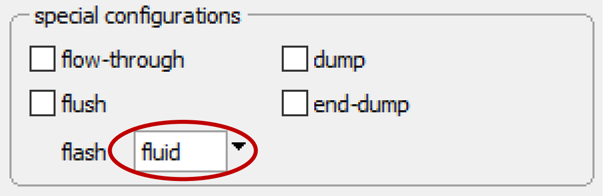


Dataset “Goethite_P.rea” lets you carry out calculations of this sort for the $NaClO_4$-P system in contact with Goethite.

3.8 “Flash diagrams”

You can use **React** to calculate “flash diagrams” of the type employed by reservoir engineers to study scaling. A flash diagram shows the mineral or end member masses that precipitate, or the saturation state of various minerals and solid solutions, when two fluids are mixed in varying proportions. The diagrams do not represent “flashing” in the geochemical sense of gas escaping from a fluid.

In calculating a flash plot, **React** gradually removes the original fluid from the equilibrium system as it titrates in a reactant fluid. Enable the feature from the **Config → Stepping…** dialog



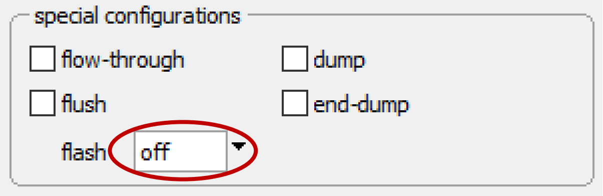
special configurations

□ flow-through □ dump

□ flush □ end-dump

flash fluid

by selecting the “fluid” option from the flash pulldown, or by typing `flash`. Disable it by selecting “off” from the pulldown, or with the command `flash off`.



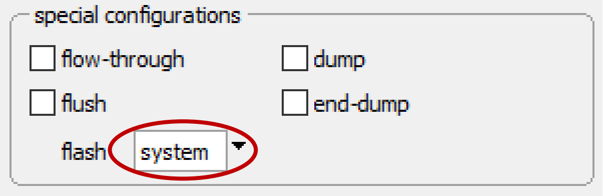
special configurations

□ flow-through □ dump

□ flush □ end-dump

flash off

Selecting “system” from the flash pulldown, or typing `flash system`



special configurations

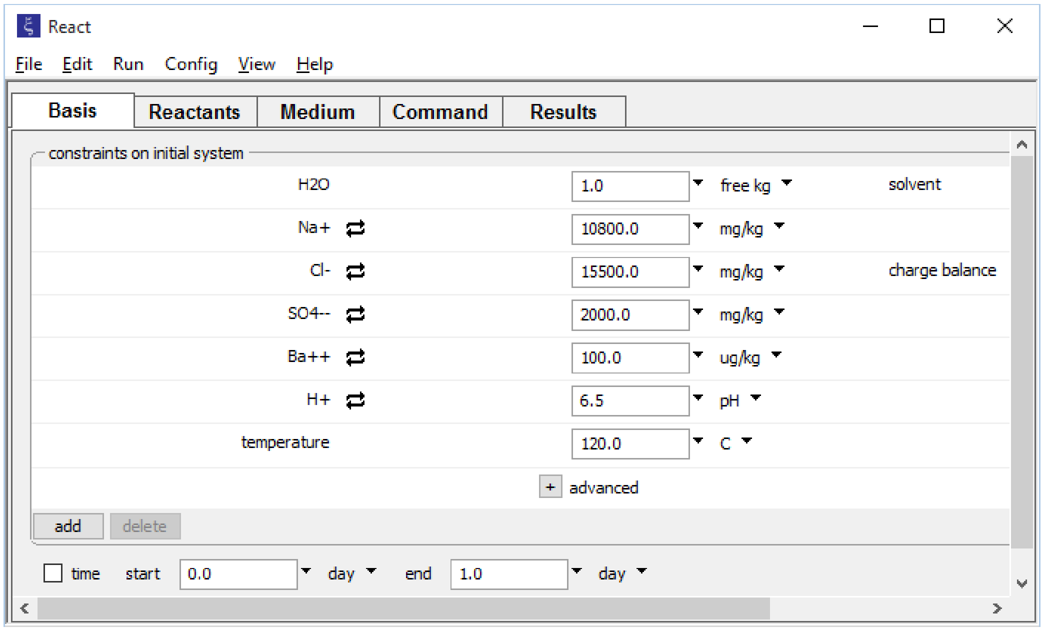
□ flow-through □ dump

□ flush □ end-dump

flash system

removes the initial system, including fluid, minerals, and end members.

For example, the input script “Flash.rea” traces a reaction path representing the mixing of two fluids in all possible mixing ratios. Double-click the file to launch **React**, then move to the **Basis** pane,



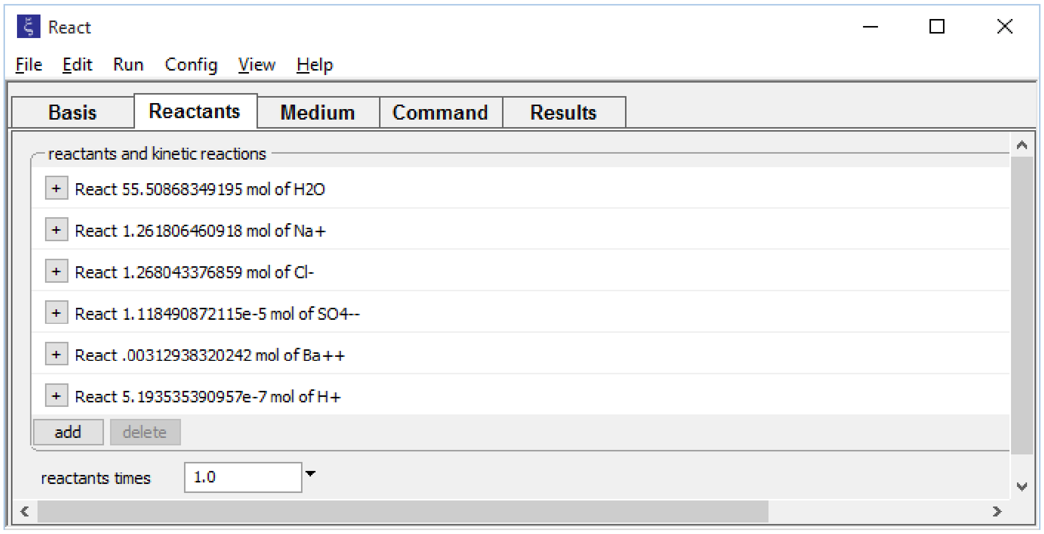
constraints on initial system

H2O	1.0	free kg	solvent
Na+	10800.0	mg/kg	
Cl-	15500.0	mg/kg	charge balance
SO4--	2000.0	mg/kg	
Ba++	100.0	ug/kg	
H+	6.5	pH	
temperature	120.0	C	

□ time start 0.0 day end 1.0 day

where a fluid rich in barium, but depleted in sulfate, is defined.

The **Reactants** pane contains a reactant fluid rich in barium but with little sulfate.



reactants and kinetic reactions

React 55.50868349195 mol of H2O

React 1.261806460918 mol of Na+

React 1.268043376859 mol of Cl-

React 1.118490872115e-5 mol of SO4--

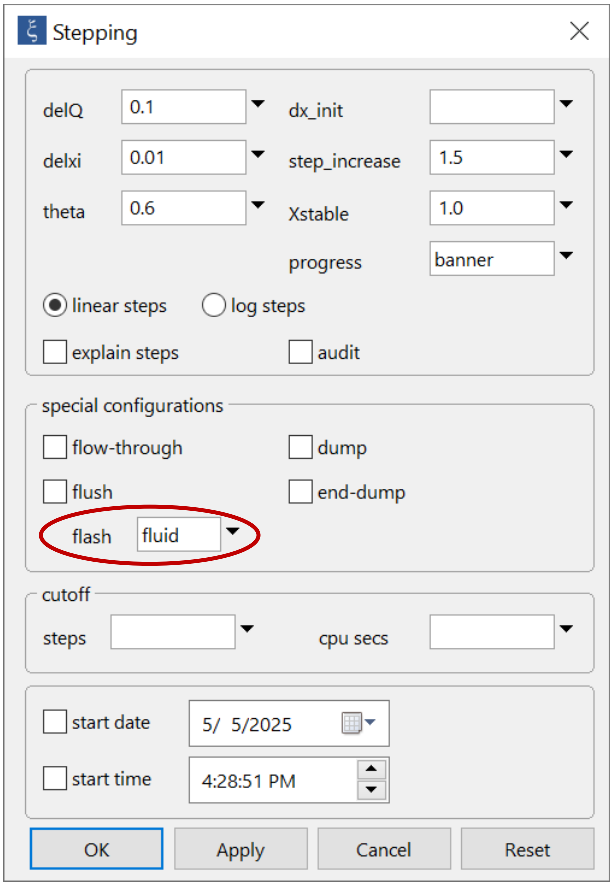
React .00312938320242 mol of Ba++

React 5.193535390957e-7 mol of H+

reactants times 1.0

The reactant fluid was originally entered on the **Basis** pane, equilibrated (**Run → Go**), then "picked up" and moved to the **Reactants** pane (**Run → Pickup → Reactants → Fluid**). The "pickup" command is explained in the section Picking up the results of a run, later in this chapter.

On the **Config → Stepping…** dialog,



delQ 0.1

delxi 0.01

theta 0.6

dx_init

step_increase 1.5

Xstable 1.0

progress banner

● linear steps ○ log steps

□ explain steps □ audit

special configurations

□ flow-through □ dump

□ flush □ end-dump

flash fluid

cutoff

steps

cpu secs

□ start date 5/ 5/2025

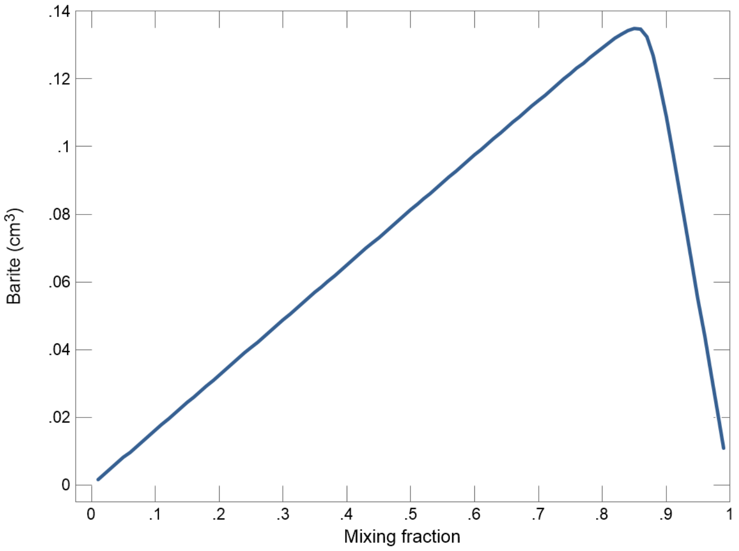
□ start time 4:28:51 PM

the flash option is enabled.

You could have instead used the commands

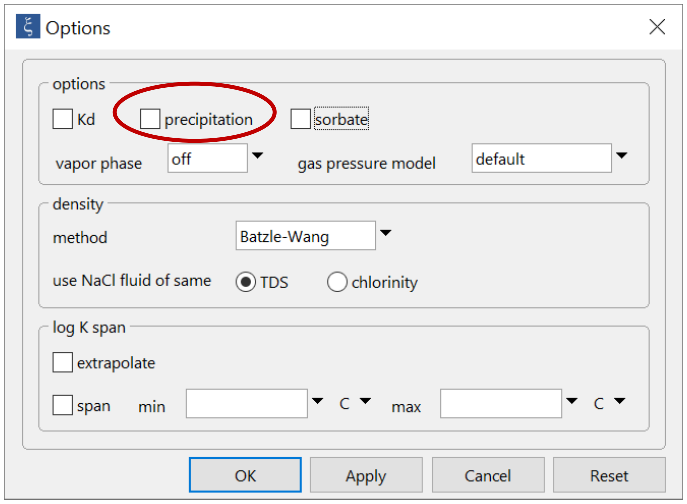
```
T = 120
Na+ = 27000 mg/kg
Cl- = 42000 mg/kg
SO4-- = 1 mg/kg
Ba++ = 400 mg/kg
pH = 5.8
go

pickup reactants = fluid

Na+ = 10800 mg/kg
Cl- = 15500 mg/kg
SO4-- = 2000 mg/kg
Ba++ = 100 ug/kg
pH = 6.5

flash = on
go
```

Run → Go traces the simulation. Using **Gtplot**, you can plot the mineral mass produced versus extent of mixing.



The horizontal axis shows the extent of mixing from purely the first fluid (left) to purely the second (right); a mixing value of 0.4, for example, represents a mixture of 60% of the first fluid and 40% of the second.

By disabling precipitation in the **React** run on the **Config → Options…** dialog,



□ Kd □ precipitation □ sorbate

vapor phase: off

gas pressure model: default

method: Batzle-Wang

use NaCl fluid of same ● TDS ○ chlorinity

□ extrapolate

□ span min C max C

or by typing `precip = off`, you can plot mineral saturation states.

3.9 Vapor phase

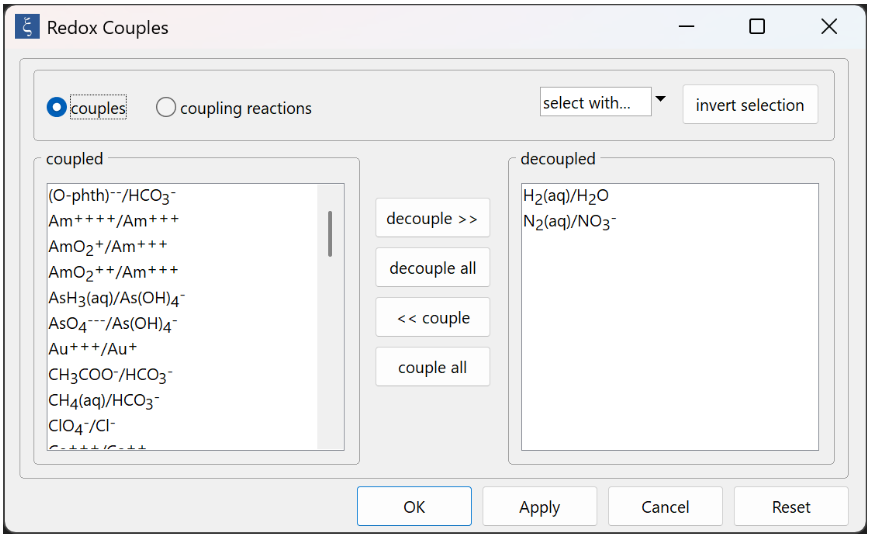
React can allow a vapor phase to evolve from gases dissolved in the fluid phase as either a closed or open system process. Please review **Vapor phase** in the **Using SpecE8** chapter of the **GWB Essentials Guide** for specific instructions on constructing models including a vapor phase before applying this feature in **React.**

The results of tracing a reaction path including a vapor phase can differ significantly depending on whether the simulation is conceptualized as a closed or open system. The vapor phase in a closed system is permitted to back-react with or condense into the fluid phase as conditions change over the course of a path; in an open system, in contrast, any gases that evolve from the fluid are segregated from further reaction.

A vapor phase might form in a **React** simulation due to water-rock interaction, perhaps by dissolving carbonate, or from heating the fluid, as might be the case when following a "sliding temperature" path. In any case, a vapor phase forms if the vapor phase option is enabled when vapor pressure reaches the prescribed confining pressure. The next section of this guide, Boiling and condensation, provides a detailed look at how **React** can be used to model the processes of boiling and condensation.

In tracing a reaction path with **React**, note that the app assumes the existence of an aqueous fluid. Hence, a vapor phase in a simulation may not evolve past the point of complete desiccation. Note as well that neither fixed nor sliding fugacity paths are compatible with the vapor phase option.

We simulate as an example sparging with N_2 gas a fluid equilibrated with atmospheric O_2 and N_2, a process that strips dissolved O_2 into the vapor. Double-click the “Vapor.rea” input file to open **React** and show the **Config → Redox Couples…** dialog



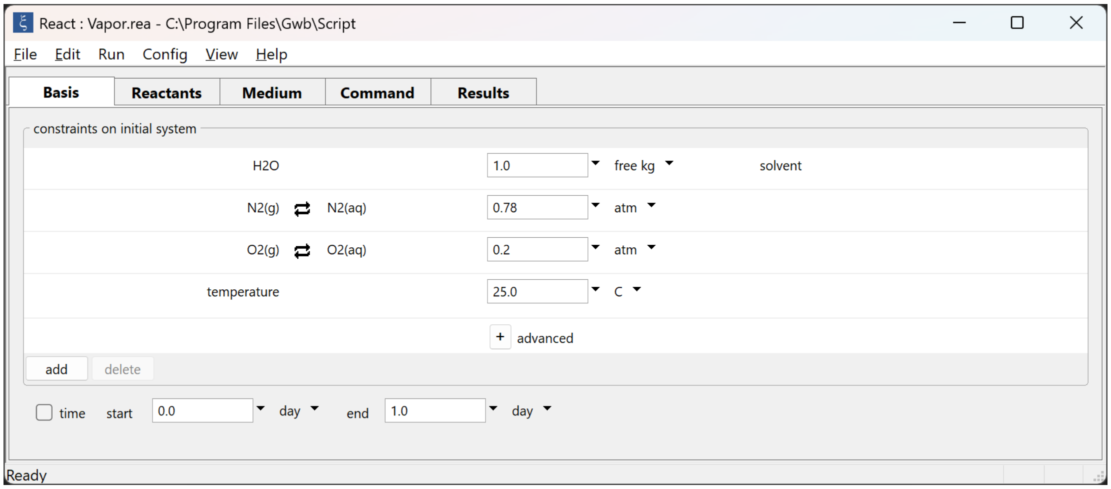
● couples ○ coupling reactions

coupled:
$(O\text{-}phth)^{--}/HCO_3^-$
Am^{++++}/Am^{+++}
AmO_2^{+}/Am^{+++}
AmO_2^{++}/Am^{+++}
$AsH_3(aq)/As(OH)_4^-$
$AsO_4^{---}/As(OH)_4^-$
Au^{+++}/Au^{+}
CH_3COO^-/HCO_3^-
$CH_4(aq)/HCO_3^-$
ClO_4^-/Cl^-

decoupled:
$H_2(aq)/H_2O$
$N_2(aq)/NO_3^-$

Here we note that redox reactions for the two gases are disabled. The N_2 gas is functionally inert, so we wish to prevent it from reacting with O_2 to form nitrate. We further need to disallow reaction between H_2 and O_2.

On the **Basis** pane, O_2(g) and N_2(g) are swapped into the Basis and constrained to their partial pressures in the atmosphere. The program defaults to a confining pressure of 1 atm, the value given at 25°C by the thermodynamic dataset.

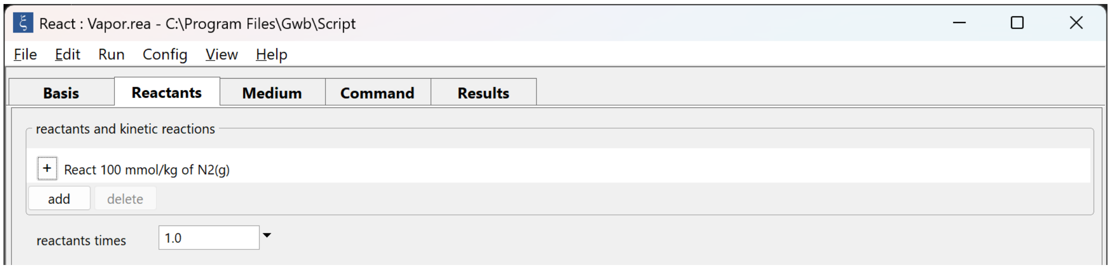

React : Vapor.rea - C:\Program Files\Gwb\Script
File Edit Run Config View Help
Basis
Reactants
Medium
Command
Results
add
delete
+ advanced
Ready

constraints on initial system

H2O	1.0	free kg	solvent
N2(g) ⇄ N2(aq)	0.78	atm	
O2(g) ⇄ O2(aq)	0.2	atm	
temperature	25.0	C	

□ time start 0.0 day end 1.0 day

On the **Reactants** pane, we titrate N_2(g) into the system

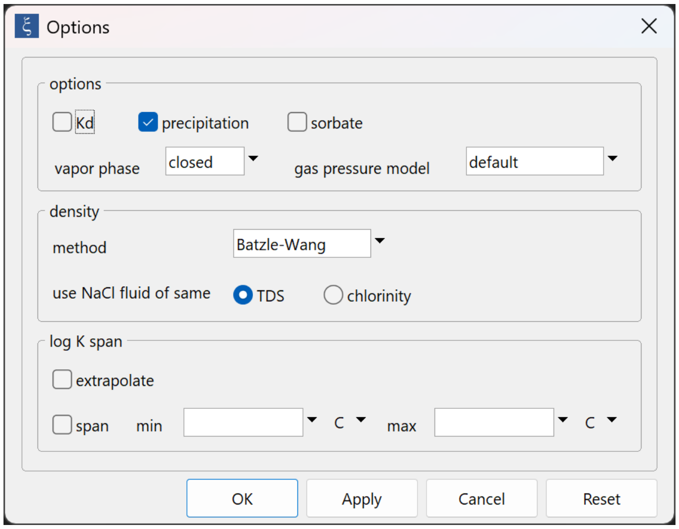

React : Vapor.rea - C:\Program Files\Gwb\Script
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete

reactants and kinetic reactions

+ React 100 mmol/kg of N2(g)

reactants times 1.0

and on the **Config → Options...** dialog, we enable the vapor phase feature, assuming a closed system



options

□ Kd ☑ precipitation □ sorbate

vapor phase closed gas pressure model default

density

method Batzle-Wang

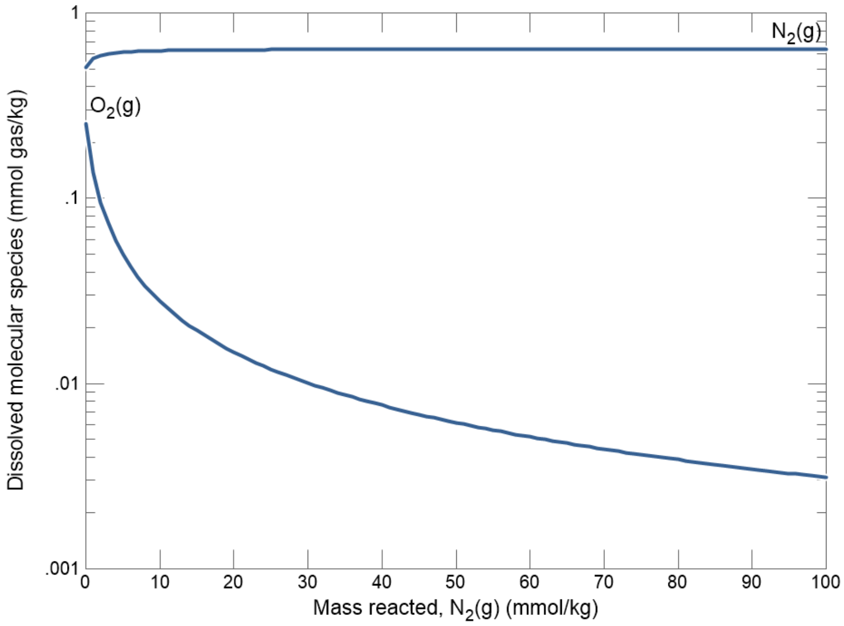
use NaCl fluid of same ● TDS ○ chlorinity

log K span

□ extrapolate

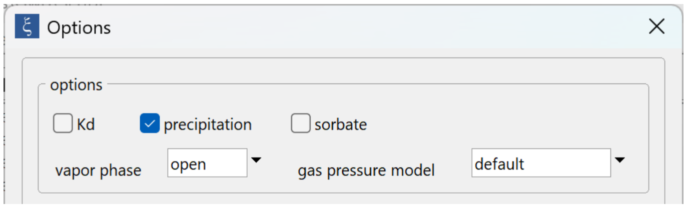
□ span min C max C

Click **Run → Go** to trace the simulation. The figure below shows how the concentrations of dissolved gases vary as N_2 gas passes through the fluid.



You may use **Gtplot** to diagram other results of note: the vapor pressure, the masses of the free gas species that evolve, their partial pressures, and their mole fractions.

Repeating the procedure assuming an open system,



Options

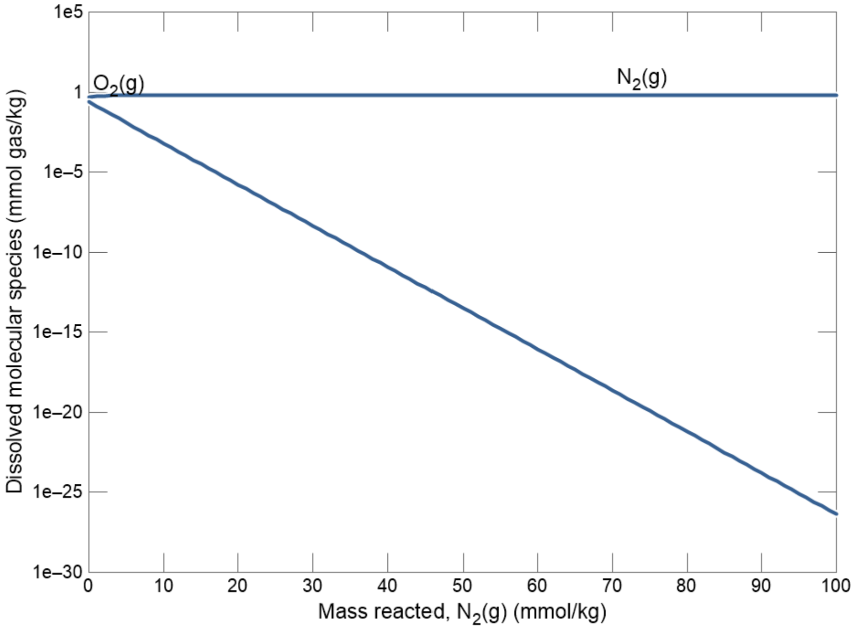
options

☐ Kd ☑ precipitation ☐ sorbate

vapor phase: open

gas pressure model: default

as might represent a benchtop experiment, we see the oxygen is sparged more efficiently when the vapor phase does not maintain contact with the fluid.



3.10 Boiling and condensation

React can use the vapor phase feature to model boiling due to the addition of heat energy to the system. Begin by consulting the previous **Vapor phase** section of this guide, which sets out the underpinnings for this class of simulation.

To configure a boiling model, you add heat to drive the temperature of the fluid and any coexisting minerals to the boiling point. The heating rate reflects the heat capacities set for the fluid and solids. Once a vapor phase evolves, the program accounts also for the energy required to vaporize water, considerably slowing the heating. A model of condensation works in the contrary sense, by removing heat energy.

To trace a boiling or condensation path, **React** enforces an enthalpy balance accounting for the heat of vaporization as well as the specific heat capacities of each phase in the system. The program calculates the molal heat of vaporization of water automatically using an internal function, specifically a correlation over the range 0°C-370°C at saturation pressure from Affandi et. al. You can, however, set the variable yourself using the `hvapor` command, either as a constant value

```
hvapor = 9700 cal/mol
```

or as a transient field variable. The specific heat capacity of water vapor is calculated from a fit of NIST's steam table data over the range 0°C-300°C at saturation pressure. You can set the parameter to a value of your choosing

```
cpv = 0.45 cal/g/C
```

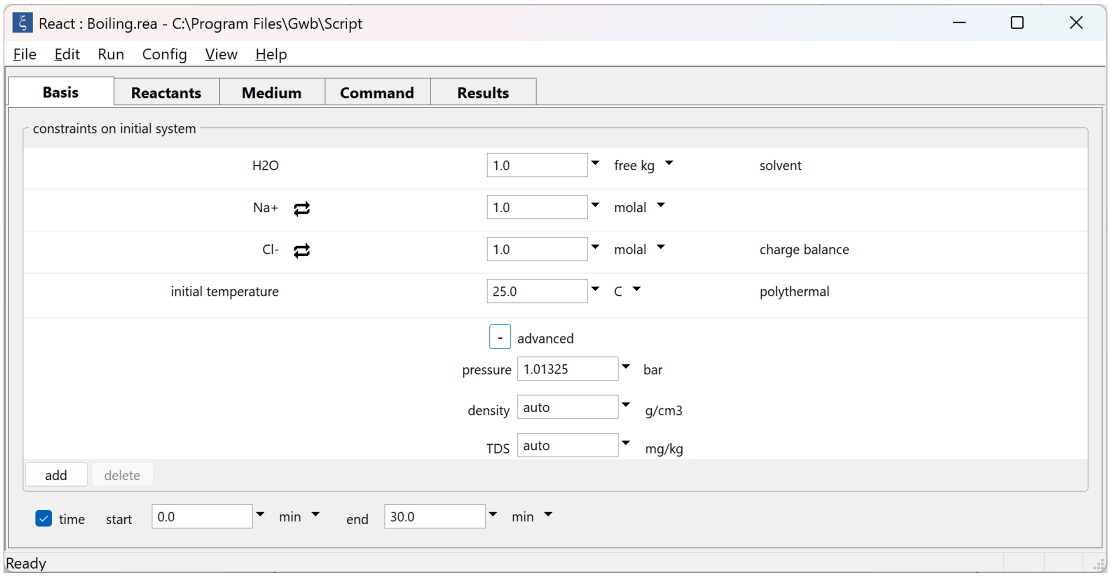
or constrain it as a transient field variable. The specific heat capacities of water and rock are set by default to 1.0 and 0.2 cal/g/°C, respectively, but you can redefine them using the `cpw` and `cpr` commands.

The thermodynamic dataset in use provides a pressure profile as a function of temperature. In many cases the profile holds steady at one atm pressure below 100°C and then follows the vapor pressure of water with increasing temperature. The program by default uses the profile to set confining pressure. You can instead specify a specific confining pressure by using the `pressure` command.

To construct a model of boiling in **React**, begin by providing an internal heat source, as described in Polythermal reaction paths in this guide and in the **Internal heat production** section of the **GWB Reactive Transport Modeling Guide**. A positive source adds heat to the system over time, facilitating boiling, whereas a negative value removes it, favoring condensation. System volume tends to change dramatically in a boiling model, so you should specify an absolute rate of heat production, in units such as J/s, rather than a rate per unit volume, such as J/m3/s. Set also an initial temperature, confining pressure, and the duration over which the heat source is applied. Finally, enable the vapor phase feature, as described in the previous section.

As an example, we calculate isobaric boiling of a NaCl brine at atmospheric pressure. Double-click the "Boiling.rea" input file to open **React**. We envision concentrating the fluid to high salinity, so we seek to choose a thermo dataset suited for high ionic strengths and valid over a range of temperatures, specifically the THEREDA project's 2023a release, which uses the Harvie-Møller-Weare activity model. If not already loaded, you would go to **File → Open → Thermo Data…** and select "THEREDA_2023a_GWB.tdat". You can view the contents of the dataset by going to **File → View → .\THEREDA_2023a_GWB.tdat**.

On the **Basis** pane, we specify the composition of the brine.


React : Boiling.rea - C:\Program Files\Gwb\Script
File Edit Run Config View Help
Basis
Reactants
Medium
Command
Results
add
delete
Ready

constraints on initial system

H2O	1.0	free kg	solvent
Na+	1.0	molal	
Cl-	1.0	molal	charge balance
initial temperature	25.0	C	polythermal

- advanced

pressure 1.01325 bar

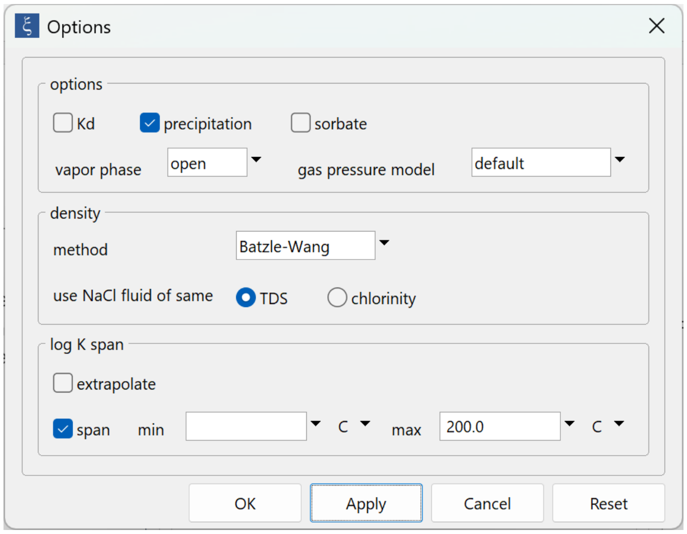
density auto g/cm3

TDS auto mg/kg

☑ time start 0.0 min end 30.0 min

Because we need to depart from the pressure profile set in the dataset, which follows the vapor saturation curve above 100°C, we specify a constant value, in bars, which you can see by clicking the "+" button next to "advanced". Finally, we set a period of 30 minutes over which heat will be added to the system.

On the **Config → Options…** dialog, we enable the vapor phase



options

□ Kd ☑ precipitation □ sorbate

vapor phase open gas pressure model default

density

method Batzle-Wang

use NaCl fluid of same ● TDS ○ chlorinity

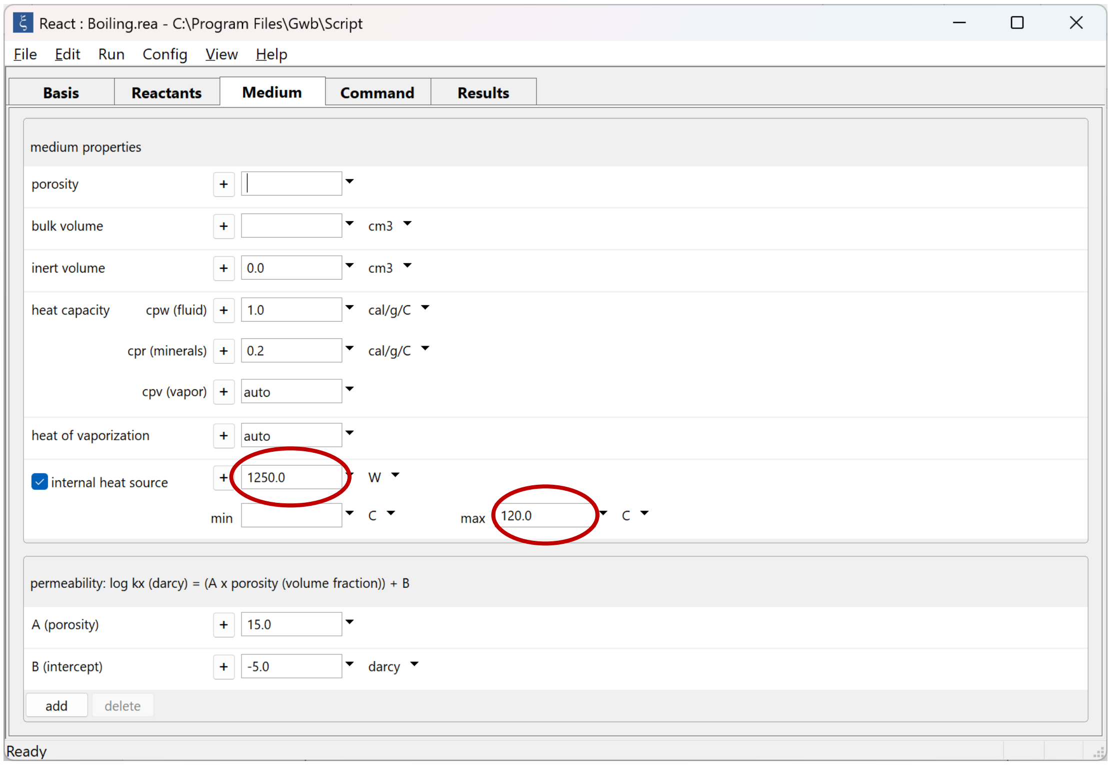
log K span

□ extrapolate

☑ span min C max 200.0 C

In this case, we specify an open system in which the water vapor once exsolved is isolated from contact with the fluid.

On the **Medium** pane,


React : Boiling.rea - C:\Program Files\Gwb\Script
File Edit Run Config View Help
Basis
Reactants
Medium
Command
Results
add
delete
Ready

medium properties

porosity

bulk volume cm3

inert volume 0.0 cm3

heat capacity cpw (fluid) 1.0 cal/g/C

cpr (minerals) 0.2 cal/g/C

cpv (vapor) auto

heat of vaporization auto

☑ internal heat source 1250.0 W

min C max 120.0 C

permeability: log kx (darcy) = (A x porosity (volume fraction)) + B

A (porosity) 15.0

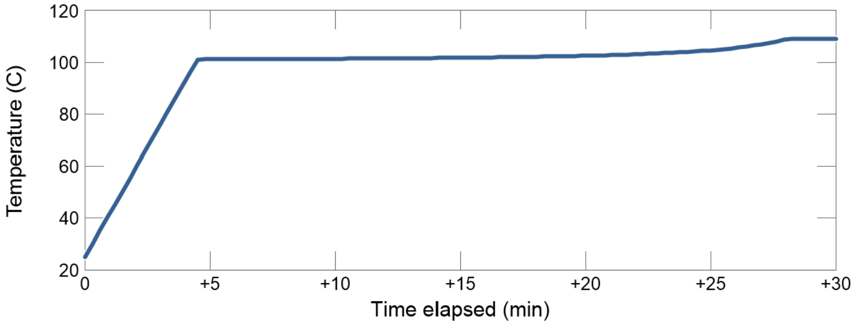
B (intercept) -5.0 darcy

we specify a 1250 W heat source for the system (1 W = 1 J/s). As previously noted, we set the rate directly, rather than per unit volume.

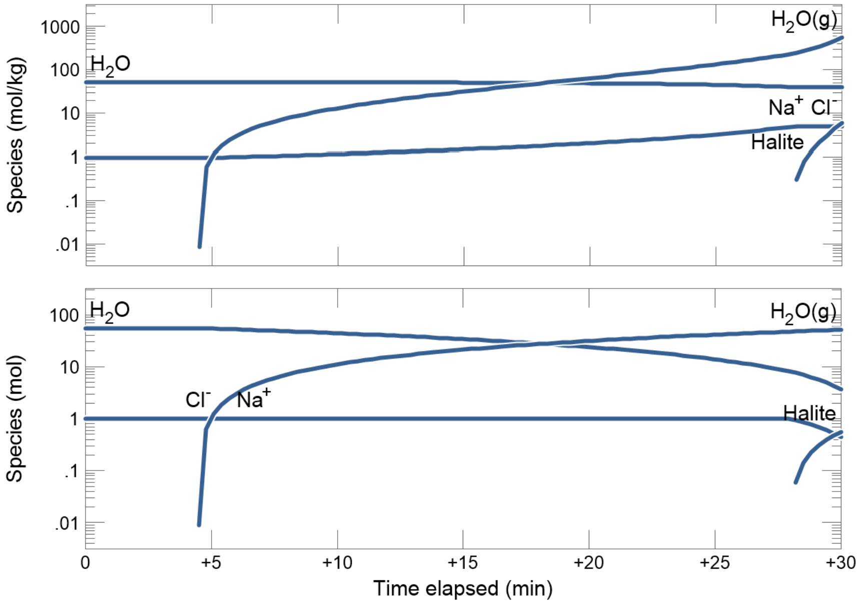
The maximum boiling temperature for a salt-saturated solution at atmospheric pressure is ~109°C. We choose a somewhat higher upper limit of 120°C for the temperature expected in the simulation, a value that is not too high to preclude loading the relevant reactions and virial parameters from the thermo dataset.

Click **Run → Go** to trace the simulation, then plot the results in **Gtplot**.

Temperature rises at a linear rate initially, given the constant value carried for the heat capacity of water. Once a vapor phase forms, however, warming proceeds more slowly, since heat is consumed vaporizing water. A gradual rise in temperature is this second leg of the calculation shows how the fluid's boiling point drifts upward with increasing ionic strength.



Eventually, halite reaches saturation and precipitates. At this point, the solution composition and boiling point become invariant.



We could continue to add heat until nearly all of the solvent water is boiled off, as we do in the "Evaporate.rea" input file described in the **Flow-through model** section of this guide, but the program will not continue past that point of complete desiccation.

In the presence of minerals, more heat would need to be added to raise temperature to the boiling point. We could investigate the effect by including within the initial system minerals or, more simply, inert volume, which has a density of 2.65 g/cm^3. For example, you could issue the command

```
inert = 50 vol%
```

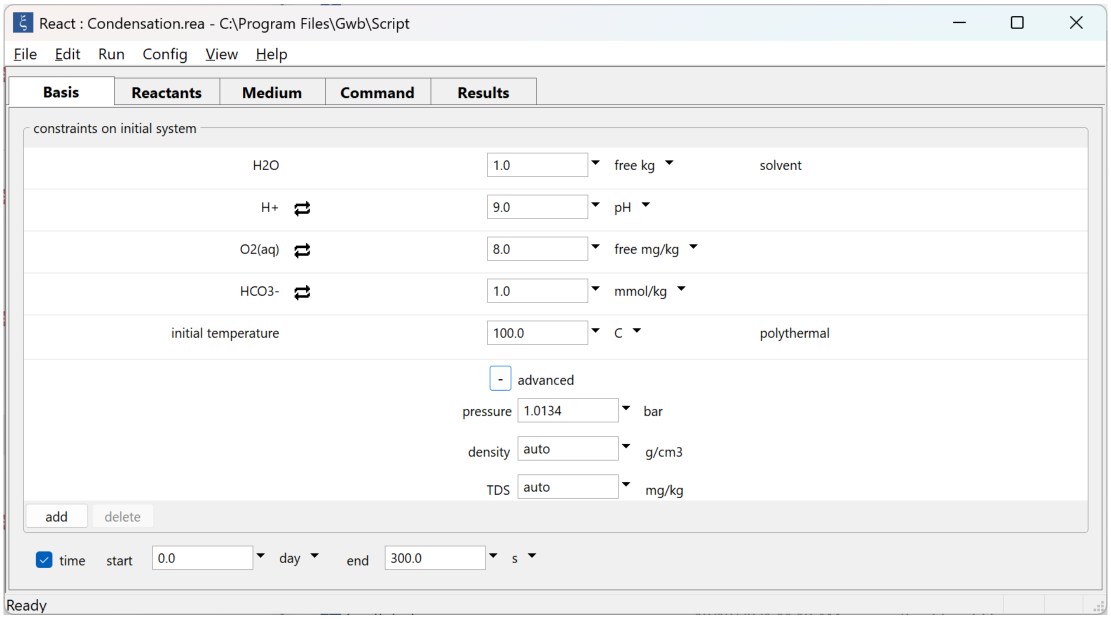
and rerun. We will need more time, or a more rapid heat source, in this case to replicate the previous simulation.

React can further consider condensation of a vapor phase, either one produced from an initially charged solution, or one that forms over the course of a boiling path. Simulations to create and subsequently condense a vapor phase need to be traced as a closed system, since lost gases are not available to condense.

The `pickup` command (see **Picking up the results of a run**) can be useful for taking the results of a boiling path, including evolved free gases, as the starting point for a new calculation. If you were to trace a boiling model as a closed system and then pick up the entire set of results as a new initial system, you could set the heat source negative to trace the original model in reverse. Or, as a variation, you could pick up the entire vapor phase and a fraction of the fluid (as a seed) to show the evolution of the condensate itself. The retained fraction of fluid phase is adjustable, as described in the referenced section of this guide. As a third option, you can pick up evolved vapor to use as reactants for a distinct simulation.

As an example, we seek to condense the vapor phase flashed from an initially charged fluid. Open **React** by double-clicking on the “Condensation.rea” input file.

On the **Basis** pane, we specify the composition of the 100°C charged fluid.


React : Condensation.rea - C:\Program Files\Gwb\Script
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete
Ready

constraints on initial system

H2O	1.0	free kg	solvent
H+	9.0	pH	
O2(aq)	8.0	free mg/kg	
HCO3-	1.0	mmol/kg	
initial temperature	100.0	C	polythermal

- advanced

pressure 1.0134 bar

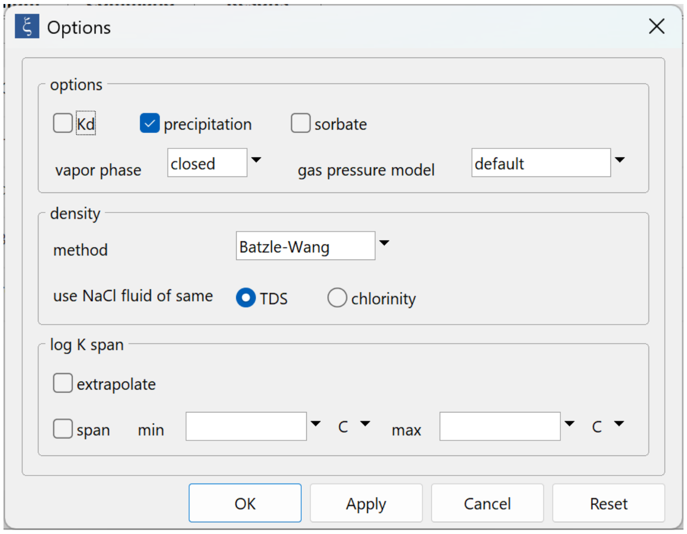
density auto g/cm3

TDS auto mg/kg

☑ time start 0.0 day end 300.0 s

and set confining pressure to atmospheric pressure, in bars. We further prescribe a period of 300 seconds (five minutes) over which heat will be removed from the system.

On the **Config → Options…** dialog, we enable the vapor phase option



options

□ Kd ☑ precipitation □ sorbate

vapor phase closed gas pressure model default

density

method Batzle-Wang

use NaCl fluid of same ● TDS ○ chlorinity

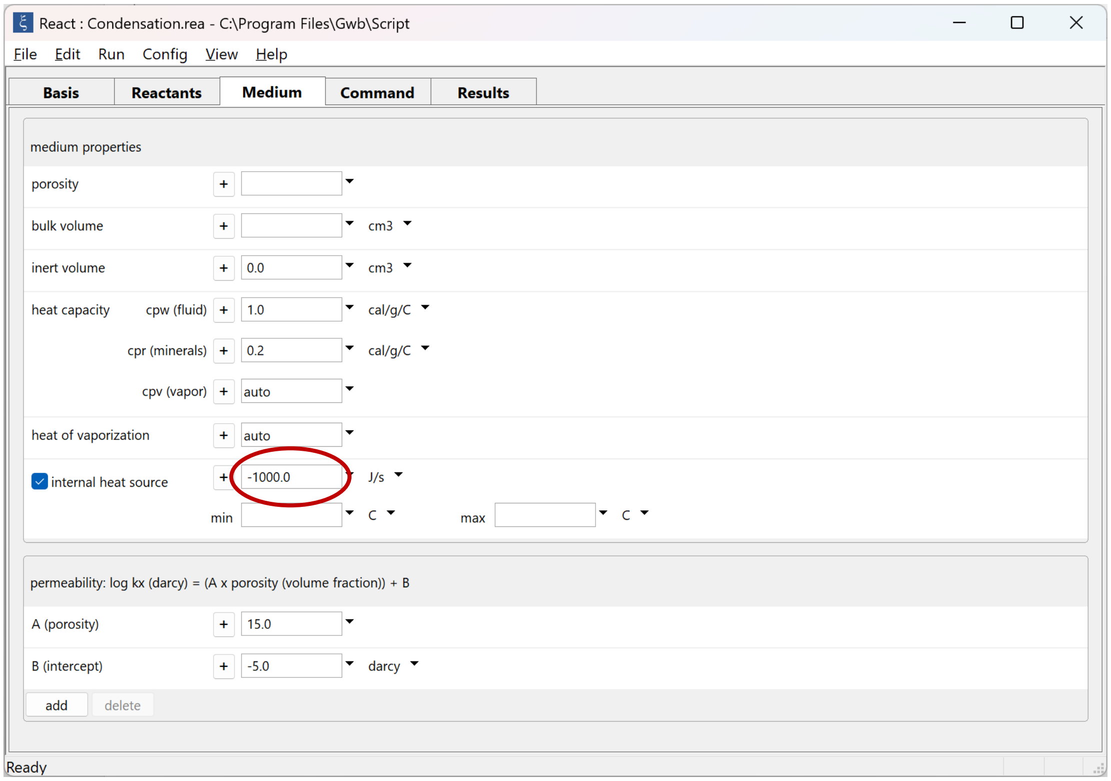
log K span

□ extrapolate

□ span min C max C

for a closed system, as required to allow condensation.

On the **Medium** pane,


React : Condensation.rea - C:\Program Files\Gwb\Script
File Edit Run Config View Help
Basis
Reactants
Medium
Command
Results
add
delete
Ready

medium properties

porosity +

bulk volume + cm3

inert volume + 0.0 cm3

heat capacity cpw (fluid) + 1.0 cal/g/C

cpr (minerals) + 0.2 cal/g/C

cpv (vapor) + auto

heat of vaporization + auto

☑ internal heat source + -1000.0 J/s

min C max C

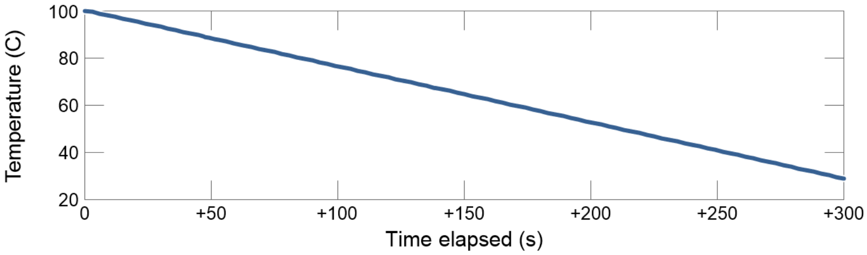
permeability: log kx (darcy) = (A x porosity (volume fraction)) + B

A (porosity) + 15.0

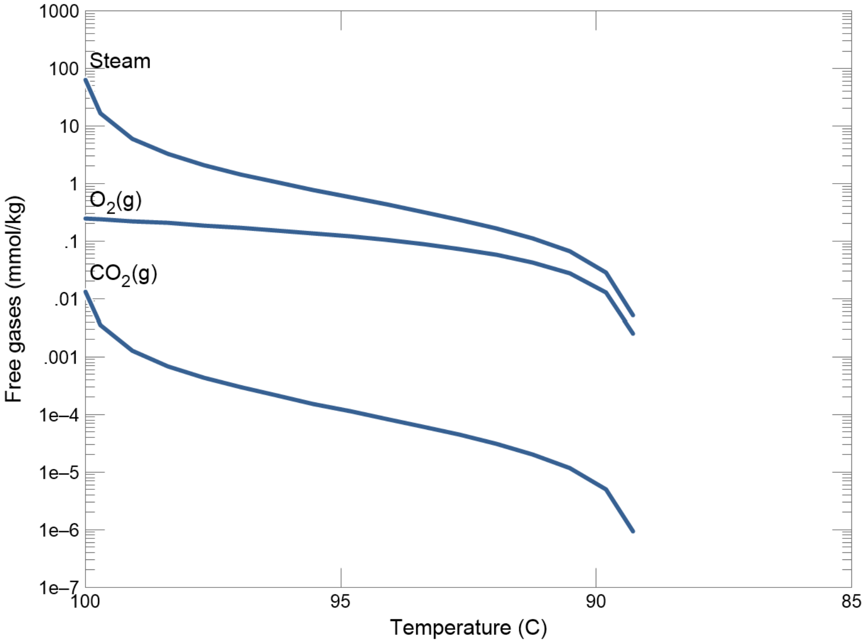
B (intercept) + -5.0 darcy

we specify a negative internal heat source of −1000 J/s.

Clicking **Run → Go** to trace the simulation and then plotting the results in **Gtplot**, we see that temperature decreases steadily from 100°C to just under 30°C.



Plotting at the masses of the most abundant free gases vs. temperature shows how the vapor phase evolves as the system cools. By about 89°C, the entire vapor phase has collapsed.

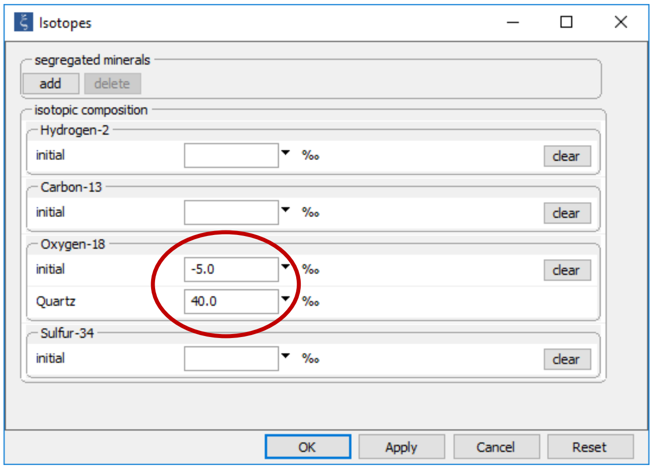


3.11 Fractionation of stable isotopes

As described in the **Fractionation of stable isotopes** section of the ***GWB Essentials Guide***, the reaction modeling programs in the GWB can account for the equilibrium fractionation of stable isotopes. The GWB is distributed with a database “isotope.dat” containing fractionation factor correlations for the isotopes ^{2}H, ^{13}C, ^{18}O, and ^{34}S. You can augment the database by adding isotopes such as ^{17}O of those elements, or isotopes like ^{57}Fe of other elements.

Program **React** can trace the equilibrium fractionation of stable isotopes over the course of a reaction path. To define an isotopic path, set first the fluid’s initial composition and then the isotopic composition of whatever reactants are to be added to the system. You set the isotopic compositions from either the **Config → Isotopes...** dialog or the **Command** pane.

To trace oxygen fractionation as a system reacts with quartz, for example, begin by adding “Quartz” as a reactant, then enter values for δ^{18}O of the initial fluid as well as the reactant mineral



initial -5.0 ‰

Quartz 40.0 ‰

Alternatively, type the command

```
oxygen initial = -5, Quartz = +40
```

You may specify compositions relative to any isotopic standard, including SMOW, PDB, or CDT. The program predicts over the reaction path isotopic compositions for species, minerals, end members, and gases relative to the same standard, and writes the results to the dataset “React_isotope.txt”, as well as to “React_plot.gtp”, where it is available for plotting by **Gtplot**.

The program begins by calculating the compositions of solvent and dissolved species to honor the fluid’s bulk composition, as specified, assuming isotopic equilibrium among species. The program then sets minerals and end members in the initial system and any buffered gases to compositions in equilibrium with the fluid.

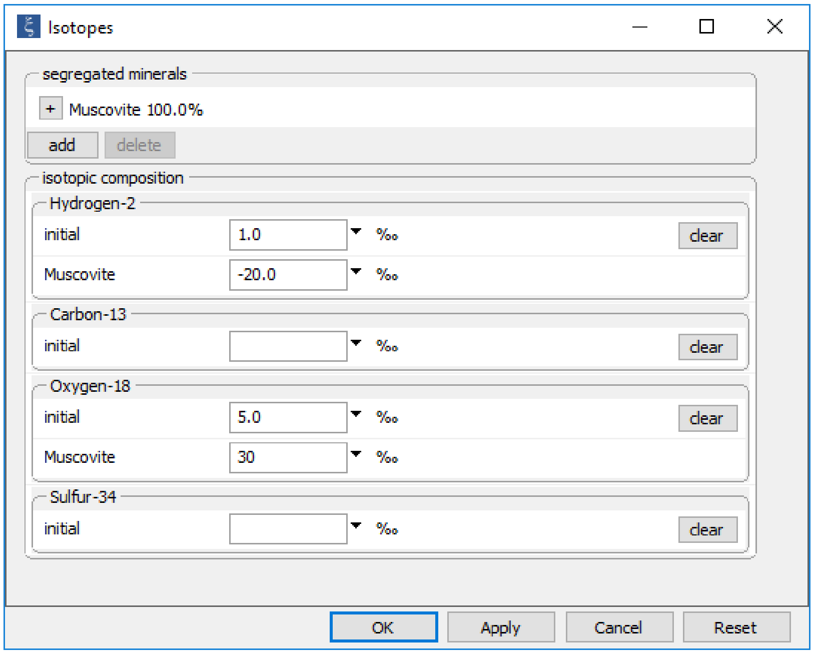
Adding or removing reactants over the course of a reaction path changes the system’s isotopic composition. The program responds by redistributing isotopic mass to maintain equilibrium among solvent, aqueous and surface species, and minerals and end members. You specify the compositions of reactants that are added to the system. When reactants are removed, however, the program assumes the composition of the reactant species as it exists at that point in the reaction path.

The calculation as described treats minerals and end members as if they exchange isotopes rapidly enough to remain in equilibrium with the changing system. This assumption, which can be unrealistic, may be relaxed by specifying that a mineral or end member, or some fraction of its mass, remain segregated from isotopic exchange.

A segregated mineral changes composition over a reaction step when it precipitates from solution; the newly precipitated mass forms in isotopic equilibrium with the system, but any of the mineral or end member present at the start of the step is unaffected. Conversely, a segregated mineral affects the system's composition when it dissolves. Note that a kinetic mineral or end member exists outside the equilibrium system and hence behaves isotopically as if it were fully segregated.

To hold a mineral segregated, move to the **Config → Isotopes...** dialog and select the mineral by clicking add. For an end member, segregate the corresponding mineral. Segregating both minerals making up a solid solution, then, segregates the solution.

For example, you could segregate muscovite and assign its composition at the start of the path as shown below (Note you must set the initial bulk fluid composition for an isotope system before you can use the GUI to set the composition of reactants or segregated minerals in that system).



segregated minerals
+ Muscovite 100.0%

isotopic composition

Hydrogen-2
initial 1.0 ‰
Muscovite -20.0 ‰

Carbon-13
initial ‰

Oxygen-18
initial 5.0 ‰
Muscovite 30 ‰

Sulfur-34
initial ‰

Alternatively, type the commands

```
segregate Muscovite
oxygen Muscovite = +30
hydrogen Muscovite = -20
```

If you do not specify a segregated mineral's composition, and it is present in the initial system, the program will set its composition to be in equilibrium with the initial fluid.

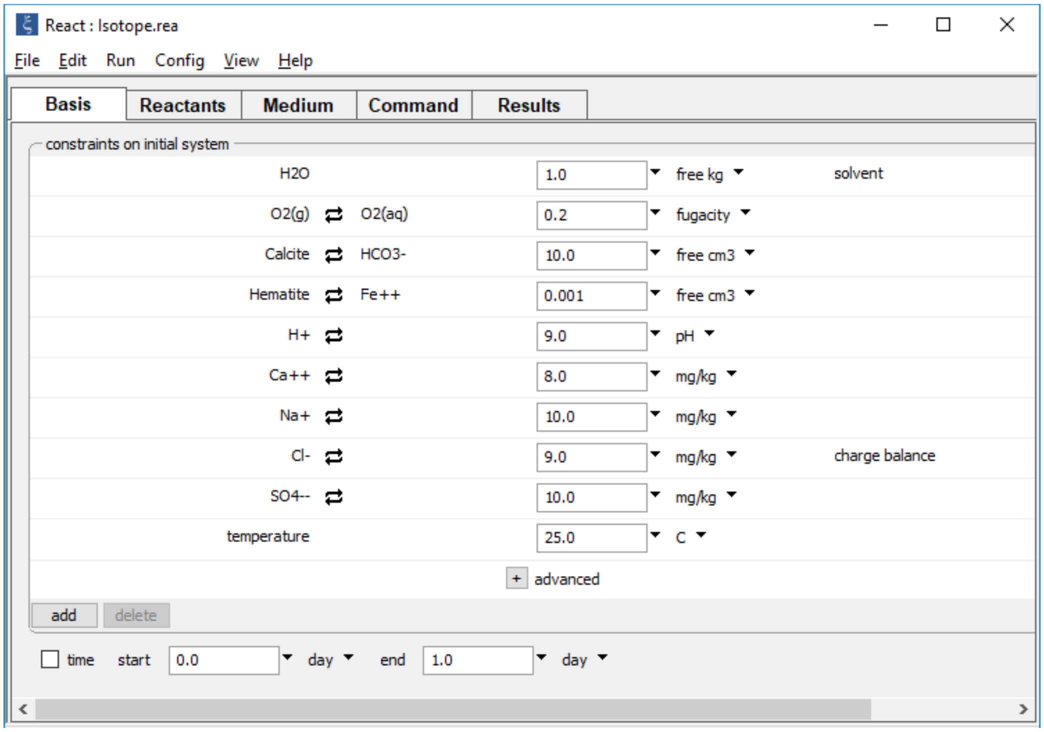
Table 3.2 Isotopic composition of reactants

Reactant	*Mass is being added*	*Mass is being removed*
Simple reactant	Set by modeler	For existing mineral or end member, its segregated plus unsegregated composition; else, composition in equilibrium with current fluid
Fixed or sliding buffer	Set by modeler; defaults to equilibrium with initial system ($\xi = 0$)	Isotopic equilibrium with fluid at current point in reaction path
Kinetic mineral or end member	Set by modeler, unless dissolving mineral is completely within the equilibrium system at $\xi = 0$, in which case the mineral's segregated plus unsegregated composition	Increment in mass of precipitating mineral or end member is in isotopic equilibrium with current system; bulk reactant composition reflects cumulative mass to current point in reaction path
Kinetic gas transfer	Set by modeler; defaults to equilibrium with initial system	Isotopic equilibrium with fluid at current point in reaction path
Charge rebalancing	Set by modeler; defaults to equilibrium with initial system	Isotopic equilibrium with fluid at current point in reaction path

React treats buffered reactants in a manner similar to simple reactants. When added, a buffered species or gas assumes the composition specified by the modeler, or, by default, the composition in equilibrium with the initial system. Upon removal, the buffer takes on the composition in equilibrium with the system at the current point in the reaction path. Table 3.2 summarizes how **React** assigns isotopic compositions to various types of reactants as it traces a reaction path.

React treats flush and flash models in the same way: you specify the isotopic composition of each component in the flushing or flashing fluid. In a flow-through model, the program reports the isotopic composition of the increment in mineral or end member mass precipitated over the current step, rather than the composition of the cumulative mass.

Consider, as an example, the effects of pyrite dissolving into a groundwater that remains oxidized by contact with the atmosphere. Double-click on the “Isotope.rea” input file to launch **React**. The **Basis** pane


React : Isotope.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete
advanced

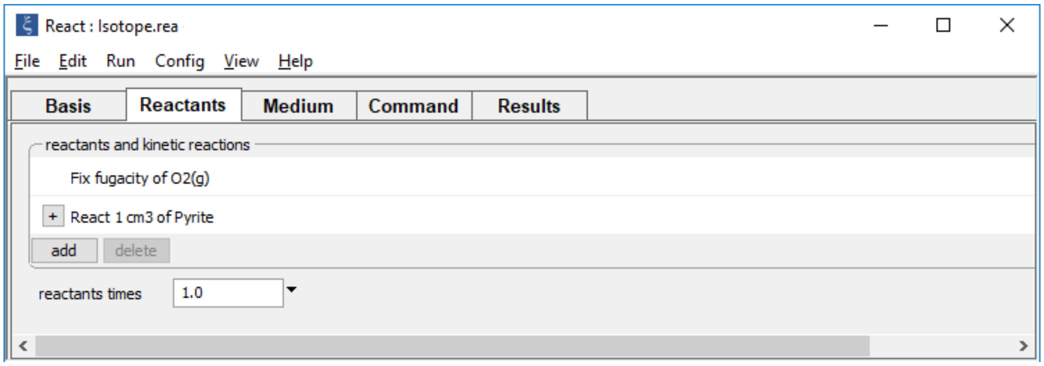
constraints on initial system

H2O			1.0	free kg	solvent
O2(g)	⇄	O2(aq)	0.2	fugacity	
Calcite	⇄	HCO3-	10.0	free cm3	
Hematite	⇄	Fe++	0.001	free cm3	
H+	⇄		9.0	pH	
Ca++	⇄		8.0	mg/kg	
Na+	⇄		10.0	mg/kg	
Cl-	⇄		9.0	mg/kg	charge balance
SO4--	⇄		10.0	mg/kg	
temperature			25.0	C	

□ time start 0.0 day end 1.0 day

describes the initial groundwater, an oxygenated fluid in equilibrium with calcite and hematite.

On the **Reactants** pane,


React : Isotope.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete

reactants and kinetic reactions

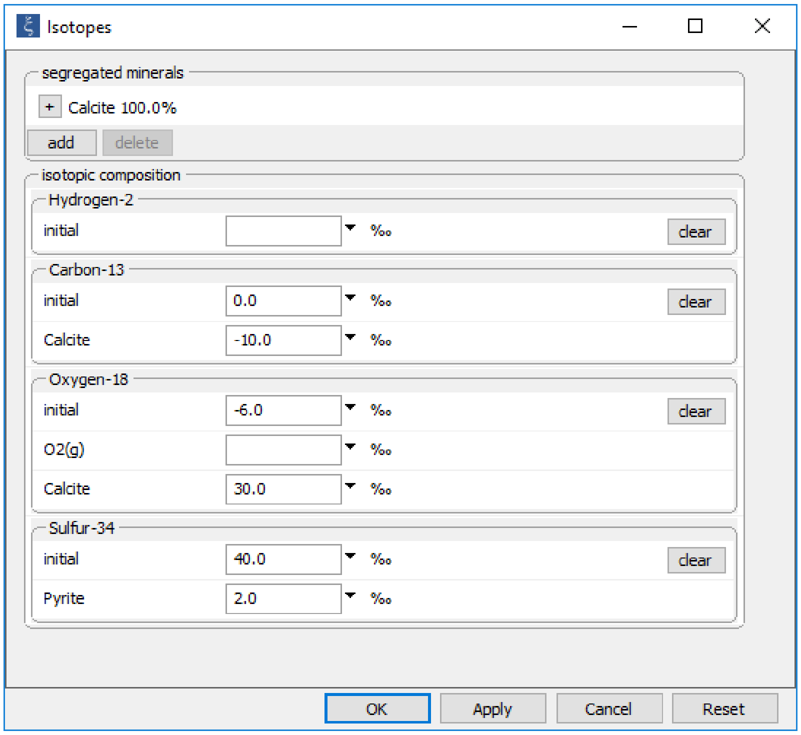
Fix fugacity of O2(g)

React 1 cm3 of Pyrite

reactants times 1.0

pyrite is reacted into the oxygenated system. As a result, calcite will dissolve and gypsum will precipitate.

On the **Config → Isotopes…** dialog,



segregated minerals

Calcite 100.0%

isotopic composition

Hydrogen-2

initial ‰

Carbon-13

initial 0.0 ‰

Calcite -10.0 ‰

Oxygen-18

initial -6.0 ‰

O2(g) ‰

Calcite 30.0 ‰

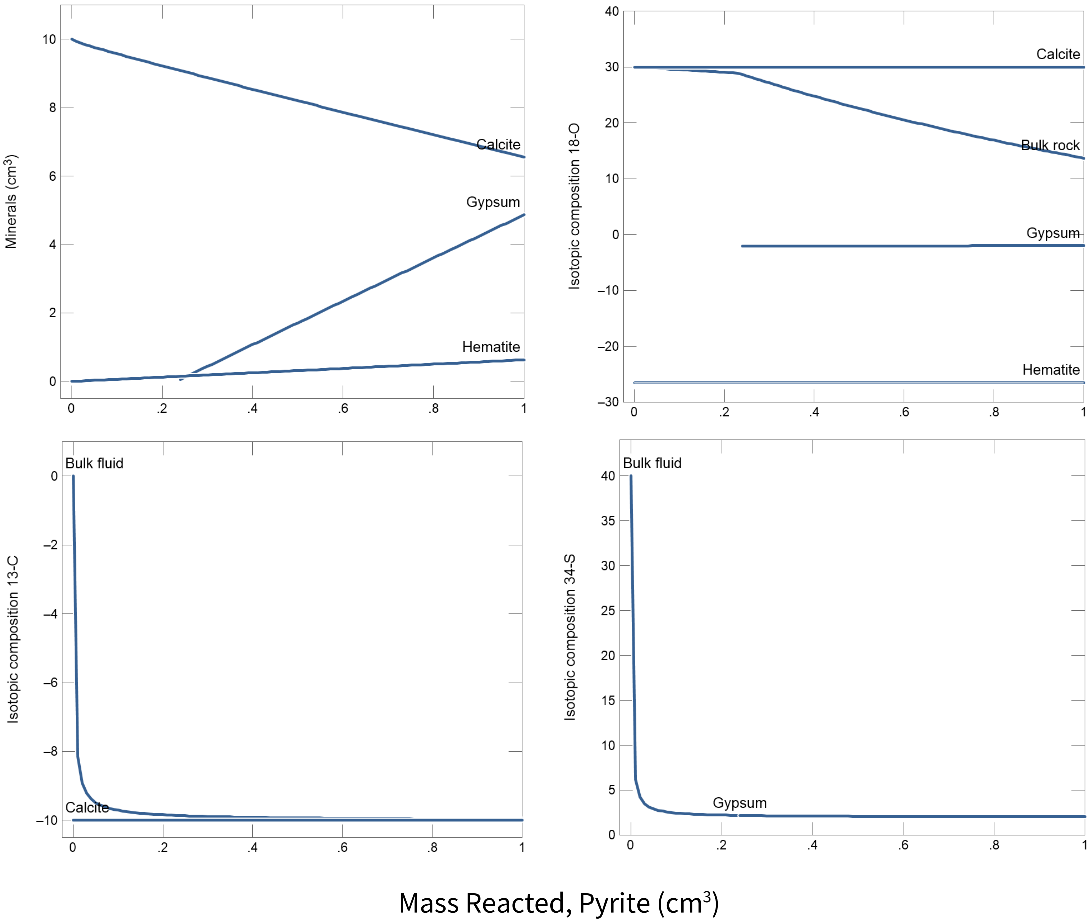
Sulfur-34

initial 40.0 ‰

Pyrite 2.0 ‰

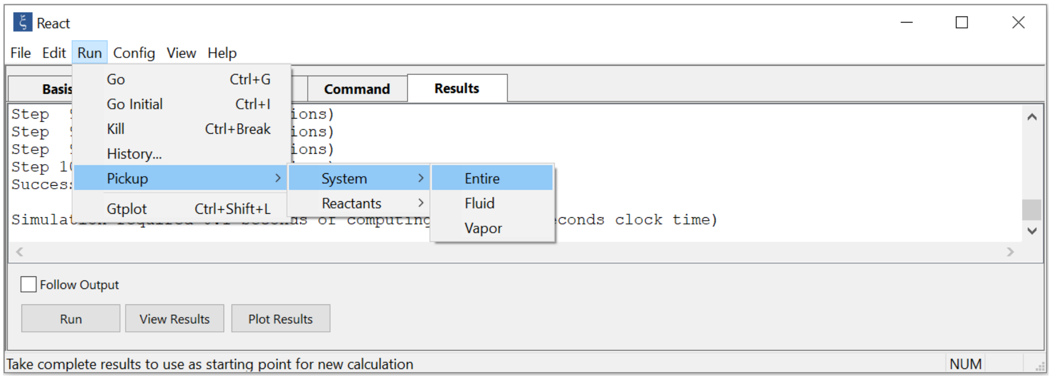
specify the isotopic composition of the initial fluid, the reactant mineral pyrite, and calcite. Calcite, which is in chemical equilibrium with the water, is segregated isotopically.

The ^{18}O and ^{34}S composition of the resulting gypsum reflects the influence, respectively, of the calcite and pyrite compositions. The fluid's ^{13}C composition varies from its initial value toward that of the calcite, and its ^{34}S value decreases toward the pyrite composition.


Minerals (cm³)
10
8
6
4
2
0
Calcite
Gypsum
Hematite
0
.2
.4
.6
.8
1
Isotopic composition 18-O
40
30
20
10
0
−10
−20
−30
Calcite
Bulk rock
Gypsum
Hematite
0
.2
.4
.6
.8
1
Isotopic composition 13-C
0
−2
−4
−6
−8
−10
Bulk fluid
Calcite
0
.2
.4
.6
.8
1
Isotopic composition 34-S
40
35
30
25
20
15
10
5
0
Bulk fluid
Gypsum
0
.2
.4
.6
.8
1
Mass Reacted, Pyrite (cm³)

3.12 Picking up the results of a run

React has the ability to “pick up” the results of a calculation and use these results as the starting point for a new calculation. You can pick up an entire system or just the fluid, minerals, or vapor phase and use these results as a new equilibrium system or a new set of reactants. Choosing from the menubar, **Run → Pickup → System → Entire**,



```
Step                                  ions)
Step                                  ions)
Step                                  ions)
Step 1(
Succes:

Simulat                         s of computing                      conds clock time)
```

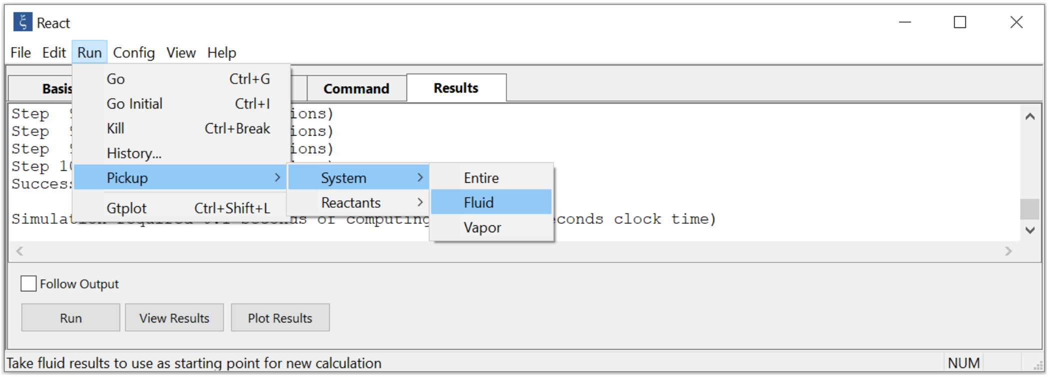
Take complete results to use as starting point for new calculation

dragging the **Results** pane onto the **Basis** pane, or issuing the command

```
pickup
```

for example, picks up fluid and any coexisting minerals, end members, and vapor as a new equilibrium system.

Run → Pickup → System → Fluid



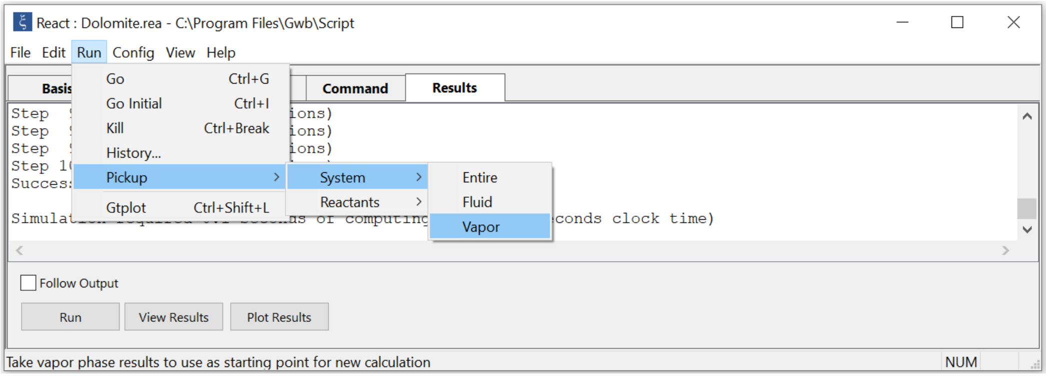
```
Step                                  ions)
Step                                  ions)
Step                                  ions)
Step 1(
Succes:

Simulat                         s of computing                      conds clock time)
```

Take fluid results to use as starting point for new calculation

or the command

```
pickup fluid
```

picks up only the fluid, whereas **Run → Pickup → System → Vapor**



```
Step  ions)
Step  ions)
Step  ions)
Step 1(
Succes:
Simulat  conds clock time)
```

Take vapor phase results to use as starting point for new calculation

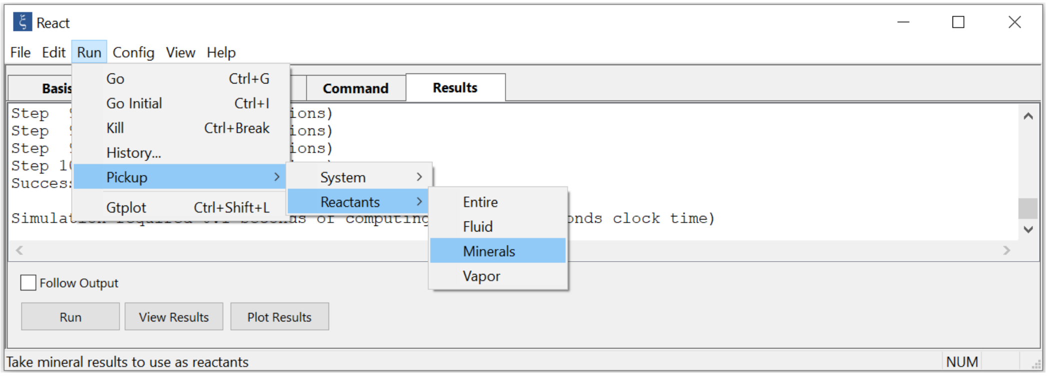
or the command

```
pickup vapor
```

picks up the entire vapor phase as well as a portion of the fluid phase, 1% by default, as the starting point for a subsequent calculation. To carry forward an arbitrary portion of the fluid phase, include the desired fraction or percentage on the command as an additional argument. For example,

```
pickup vapor 0.1%
```

As an alternative to picking up results to use as the starting point for a calculation, you can use them as a reactant list. For example, **Run → Pickup → Reactants → Minerals**,



```
Step  ions)
Step  ions)
Step  ions)
Step 1(
Succes:
Simulat  onds clock time)
```

Take mineral results to use as reactants

or the command

```
pickup reactants = minerals
```

picks up the resulting minerals and end members as a set of reactants.

You can use the pickup feature, as an example, to simulate the mixing of two fluids. To mix fluid 1 (whose composition, pH, etc. are listed in dataset “Fluid1”) into fluid 2 (in dataset “Fluid2”), follow the steps

```
read Fluid1
go
pickup reactants = fluid

read Fluid2
reactants times 10
go
```

React equilibrates fluid 1 and takes the fluid’s bulk composition as a reactant. The program then equilibrates fluid 2 and reacts about 10 kg of fluid 1 into the second fluid over the course of a reaction path.

The `pickup` command also allows you to pick up kinetic reactions at their endpoint states, and thus chain together kinetic reaction paths. This feature is discussed in Chaining together kinetic paths in the next chapter of this guide.

Kinetic Reaction Paths

In the reaction models discussed to this point, reactants are added to (or removed from) a system at arbitrary rates. **React** can also trace reactions in which kinetic rate laws control the time rates at which reactions proceed.

React can trace the progress through time of several types of kinetic reactions, including

- The dissolution and precipitation of minerals, or solid solution end members
- The dissociation and association of aqueous complexes, or surface complexes
- The transfer of gases from an external reservoir, or into one
- Oxidation and reduction (redox) reactions, in solutions held in redox disequilibrium
- The rate of microbial metabolism and the growth of microbial communities

You can construct a model considering a single kinetic reaction, or one combining several reactions or types of reactions. Kinetic modeling is discussed in detail in the remainder of this section.

4.1 Setting kinetic reactions

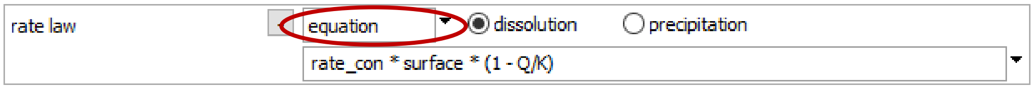
To construct a kinetic reaction path, specify the time span of the simulation on the **Basis** pane, or with the `time` command. Then set one or more kinetic reactions, either on the **Reactants** pane (click add and the appropriate option) or using the `kinetic` command. In this way, you define the specifics of the rate law, such as the rate constant to be used, reaction orders, and so on. Table 4.1 summarizes the keywords that can be used to define kinetic reactions.

Table 4.1 Summary of keywords for the **kinetic** command. Symbol ✓ indicates that for the applicable reaction types, a value for the keyword must be set. For symbol *, note the following: (1) an activation energy and a pre-exponential constant can be specified in place of a rate constant; (2) a value for **surface** must be set for redox reactions considering a heterogeneous catalyst; (3) a value for **KA** is required for enzymatically promoted reactions; and (4) a value for **mE** or **aE** is required when modeling enzyme kinetics. For symbol †, note: (1) units shown for **rate_con** and **pre-exp** are in the absence of promoting and inhibiting species, as set with **apower** or **mpower** keyword; (2) units for **KA** and **KD** depend on settings for **mpowerA** and **mpowerD**. Use keywords **apower**, **apowerA**, and **apowerD** to set a species' activity in rate law, and **mpower**, **mpowerA**, and **mpowerD** to set its molality.

	reaction type						
keyword	*mineral*	*complex*	*gas*	*redox*	*microbe*	*default*	*unit*
rate_con	•		•			✓	mol/cm^2 sec†
		•		•		✓	molal/s *or* molal/cm^2 sec†
					•	✓	mol/mg sec†
pre-exp	•	•	•	•	•	*	*same as* rate_con
act_en	•	•	•	•	•	*	J/mol
surface, contact	•		•			✓	cm^2/g
				•		*	cm^2/g *or* cm^2
apower, mpower	•	•	•	•	•	none set	
order1	•	•	•	•	•	1	
order2	•	•	•	•	•	1	
xaffin	•					not set	
nucleus	•					0	cm$^{2/}$ cm^3
critSI	•					0	
f_ext			•			equilib'm	fugacity
p_ext			•			not set	bar

rxn	•	•	✓	
catalyst	•		off	
mE, aE	•		*	molality *or* activity
KA	•		*	molality *or* activity
KP	•		not set	molality *or* activity
KA		•	0	†
KD		•	0	†
ATP_energy		•	0	kJ/mol
ATP_number		•	0	
biomass		•	✓	mg/kg
PKA		•	1	
PKD		•	1	
apowerA, mpowerA		•	none set	
apowerD, mpowerD		•	none set	
growth_yield		•	0	mg biomass/mol
decay_con		•	0	sec^{-1}

The rate law for each type of kinetic reaction is coded within **React** in a form sufficiently general to support most applications in geochemistry. This form is the “built in” rate law. You may, furthermore, specify the form of a rate law directly. **React** evaluates the rate law you set using an internal interpreter. For example, after adding the kinetic mineral quartz from the **Reactants** pane, you could change the rate law pulldown from “built in” to “equation” and then set the form directly, as in the box below.



rate law equation ● dissolution ○ precipitation

rate_con * surface * (1 - Q/K)

The equation prescribes a rate law of the form shown, for the dissolution and precipitation of quartz. On the **Command** pane, you would type

```
kinetic Quartz rate_law = "rate_con * surface * (1 – Q/K)"
```

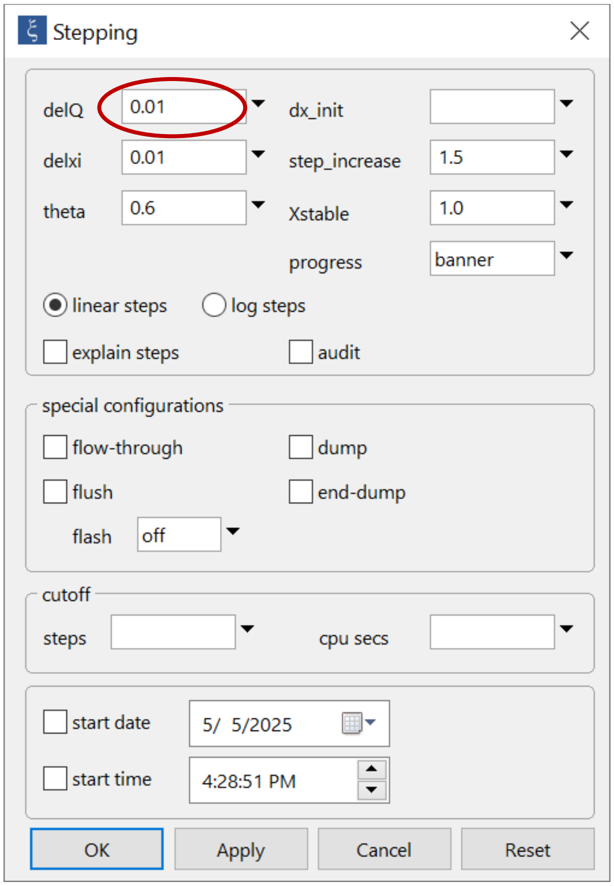
You can also write basic scripts to calculate reaction rates, and even write and compile C++ functions that **React** links to and executes at run time. Methods for setting custom rate laws are described in detail in the next section of this guide, **Custom Rate Laws**.

At each point in a simulation, **React** selects a time step short enough to accurately trace each kinetic reaction considered, and to ensure numerical stability. Introducing kinetic rate laws can sometimes lead to lengthy runs, for this reason, if the time span of the simulation is long relative to the characteristic times of the kinetic reactions considered. Calculating a simulation spanning millions of years, for example, might require a lot of computing time if the run includes kinetic reactions that approach equilibrium within a few days.

To constrain a time step, the program allows only a small change in the relative value of the ion activity product (i.e., in $\Delta Q/Q$) for each kinetic reaction. The limiting projected change in this quantity is by default 0.1, but you can set your own criterion, as described below. Setting a value larger than the default can lead to longer time steps, whereas a smaller value might improve stability by forcing smaller steps. You can also constrain the time step directly by adjusting the variable "delxi".

A time weighting variable θ, which can range from 0 to 1, controls the method **React** uses to integrate rate laws. When θ is 0, the integration proceeds by evaluating reaction rates only at the start of each time step, whereas a θ value of 1 causes the rates to be figured at the step's end, as part of the Newton-Raphson iteration. Prescribing an intermediate value sets a weighted average of the rates, before and after. Values close to one-half give runs that are more stable numerically than when $\theta = 0$, and lead to less truncation error than when θ is either 0 or 1. In **React**, the default value is 0.6.

Values "delQ", "delxi", and "theta" are examples of "settable variables," as described under **Settable variables** in the **Getting Started with React** section of this guide. You can alter these values directly from the command line or using a script, or on the **Config → Stepping…** dialog. To constrain the time stepping tightly, for example, you might reduce the value for "delQ" by a factor of 10 by changing the value to 0.01



Stepping

delQ 0.01

delxi 0.01

theta 0.6

dx_init

step_increase 1.5

Xstable 1.0

progress banner

● linear steps ○ log steps

□ explain steps □ audit

special configurations

□ flow-through □ dump

□ flush □ end-dump

flash off

cutoff

steps

cpu secs

□ start date 5/ 5/2025

□ start time 4:28:51 PM

or type `delQ = 0.01`. The command `show variables` lists the settable variables and their current values.

4.2 Kinetics of precipitation and dissolution

A simple rate law for mineral dissolution and precipitation takes the form

$$r_{\vec{k}} = A_S k_+ \left(1 - \frac{Q}{K}\right) \tag{4.1}$$

Here $r_{\vec{k}}$ is reaction rate (mol s^{-1}, positive for dissolution), A_S is the mineral's surface area (cm^2), k_+ is the rate constant (in mol cm^{-2} s^{-1}), and Q and K are the activity product and equilibrium constant for the dissolution reaction.

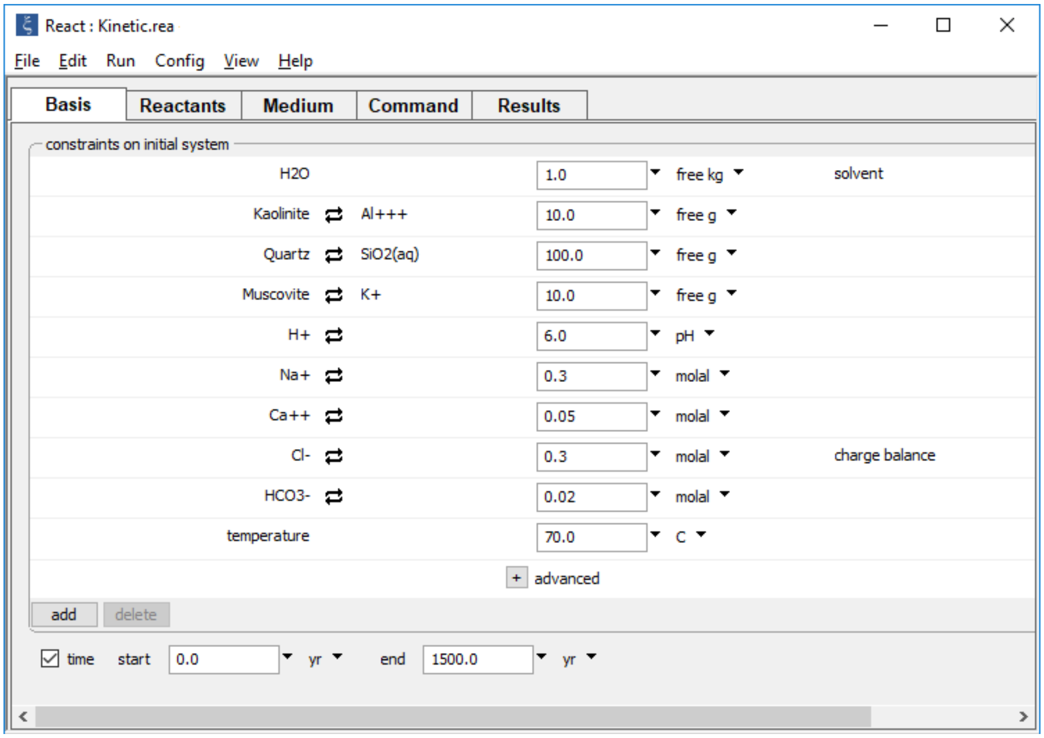
By this equation, a mineral precipitates when it is supersaturated and dissolves when it is undersaturated at a rate that depends on its rate constant and surface area. **React** calculates surface area from a specific surface area in $cm^2\ g^{-1}$ entered by the user for each kinetic mineral. The user also supplies the value of the rate constant, or an activation energy E_A and pre-exponential factor A for the Arrhenius equation

$$k_+ = A\, e^{-E_A/\,RT_K} \tag{4.2}$$

so that the program can determine k_+ as temperature varies. Here, R is the gas constant and T_K is the absolute temperature in Kelvins. There is provision in **React** for more complicated rate expressions than the rate law above, such as those in which the rate law accounts for promoting and inhibiting species in solution, as discussed later.

You can set kinetic rate laws for minerals swapped into the initial system and thus initially in equilibrium with the fluid, as well as for minerals set as reactants. However, a kinetic mineral must exist in some amount (or a nucleation term must be set, as described below); otherwise, its surface area will be 0 and it will never precipitate.

As an example, trace the kinetic dissolution of albite at 70°C into a water in equilibrium with kaolinite, muscovite, and quartz, assuming that the reaction of quartz is also controlled kinetically. Open “Kinetic.rea” and move to the **Basis** pane.



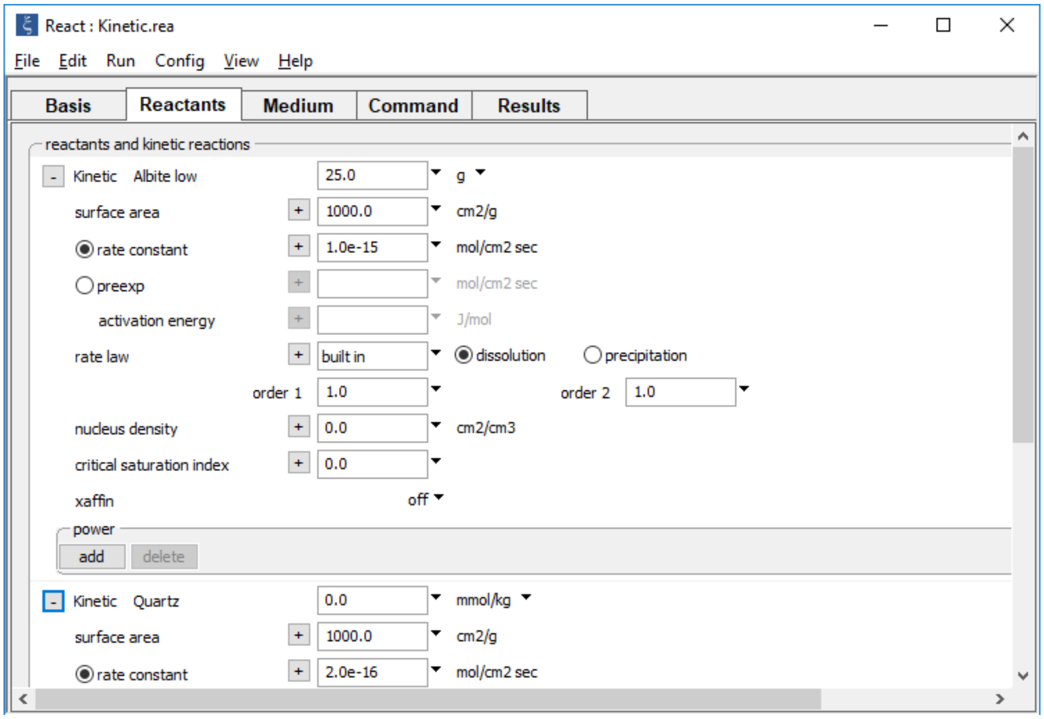
constraints on initial system

H2O 1.0 free kg solvent
Kaolinite ⇄ Al+++ 10.0 free g
Quartz ⇄ SiO2(aq) 100.0 free g
Muscovite ⇄ K+ 10.0 free g
H+ ⇄ 6.0 pH
Na+ ⇄ 0.3 molal
Ca++ ⇄ 0.05 molal
Cl- ⇄ 0.3 molal charge balance
HCO3- ⇄ 0.02 molal
temperature 70.0 C

☑ time start 0.0 yr end 1500.0 yr

Swapping kaolinite, muscovite, and quartz into the basis sets the 70°C fluid in equilibrium with the minerals. Set a time span of 1500 years.

On the **Reactants** pane, expand the “Kinetic” fields for the two minerals.



reactants and kinetic reactions

- Kinetic Albite low 25.0 g
surface area 1000.0 cm2/g
● rate constant 1.0e-15 mol/cm2 sec
○ preexp mol/cm2 sec
activation energy J/mol
rate law built in ● dissolution ○ precipitation
order 1 1.0 order 2 1.0
nucleus density 0.0 cm2/cm3
critical saturation index 0.0
xaffin off
power

- Kinetic Quartz 0.0 mmol/kg
surface area 1000.0 cm2/g
● rate constant 2.0e-16 mol/cm2 sec

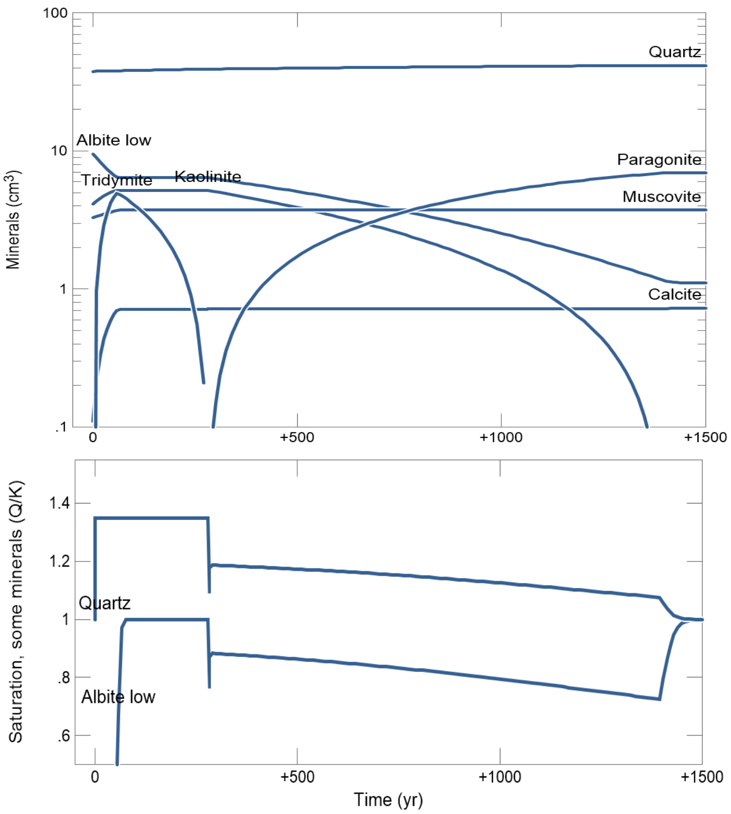
Rate constants of 10^{-15} and 2×10^{-16} mol cm^{-2} s^{-1} are set for albite and quartz, and a specific surface area of 10^3 cm^2 g^{-1} is set for each mineral.

If working from the **Command** pane, type

```
T = 70
swap Kaolinite for Al+++
swap Quartz for SiO2(aq)
swap Muscovite for K+

pH = 6
Na+ = .3 molal
Ca++ = .05 molal
Cl- = .3 molal
HCO3- = .02 molal
10 free g Kaolinite
100 free g Quartz
10 free g Muscovite
```

```
(cont'd)
react 25 g "Albite low"
kinetic "Albite low" rate_con = 1e-15 surf = 1e3
kinetic Quartz rate_con = 2e-16 surf = 1e3

time begin 0 yr, end 1500 yr
go
```

Run → Go traces the simulation. The results



show that albite reaches equilibrium twice in the calculation, once in the presence of tridymite and again in the presence of quartz.

Each end member in a continuous solid solution can be similarly set to react according to a kinetic rate law. The activity product *Q* carried in the rate law in this case accounts for the end member's activity, rather than the unity activity of the corresponding mineral.

4.2.1 Promoting and inhibiting species

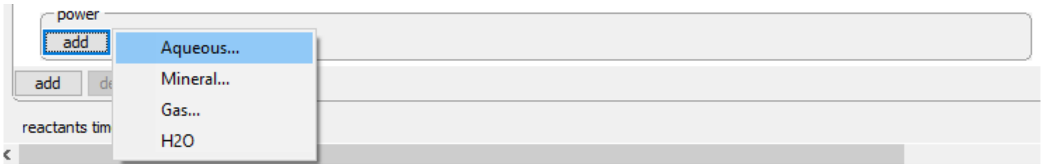
The kinetic rate law carried to this point can be generalized to account for the effects of promoting and inhibiting species. In this case, the law takes the form

$$r_{\vec{k}} = A_S\, k_+ \prod_j (a_j|m_j)^{p_j} \left(1 - \frac{Q}{K}\right) \tag{4.3}$$

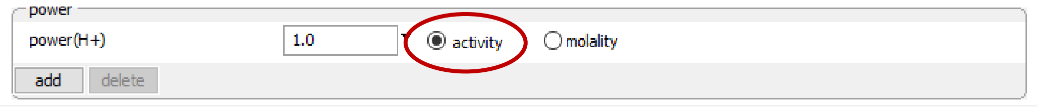
Here $\prod$ is the product function (the multiplicative equivalent of the summation function $\sum$), a_j and m_j are the activity and molality of each promoting or inhibiting species (the notation "$a_j|m_j$" signifies that either the species activity or molality is carried), and p_j is that species' "power" in the rate law.

A promoting species has a positive value for p_j, and an inhibiting species has a negative value. When there are no promoting or inhibiting species, this rate law reduces to the previous form. Aqueous species, minerals (represented in the rate law by unit activity or by molality), end members (represented by activity or mole fraction), gas species (represented by fugacity or partial pressure), surface complexes (represented by the species' molality), and solvent water (represented by activity) can serve as promoting and inhibiting species.

To set the promoting and inhibiting species in a rate law, click the add button within the "power" section of the kinetic reactant and choose a species from the pulldown list.



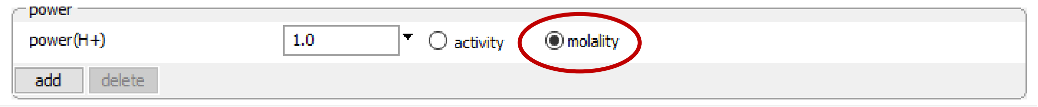
Set a value for the "power" (the default is 1), then choose whether the species activity



power

power(H+) 1.0 ● activity ○ molality

or molality should be evaluated in the rate law by selecting the appropriate button.

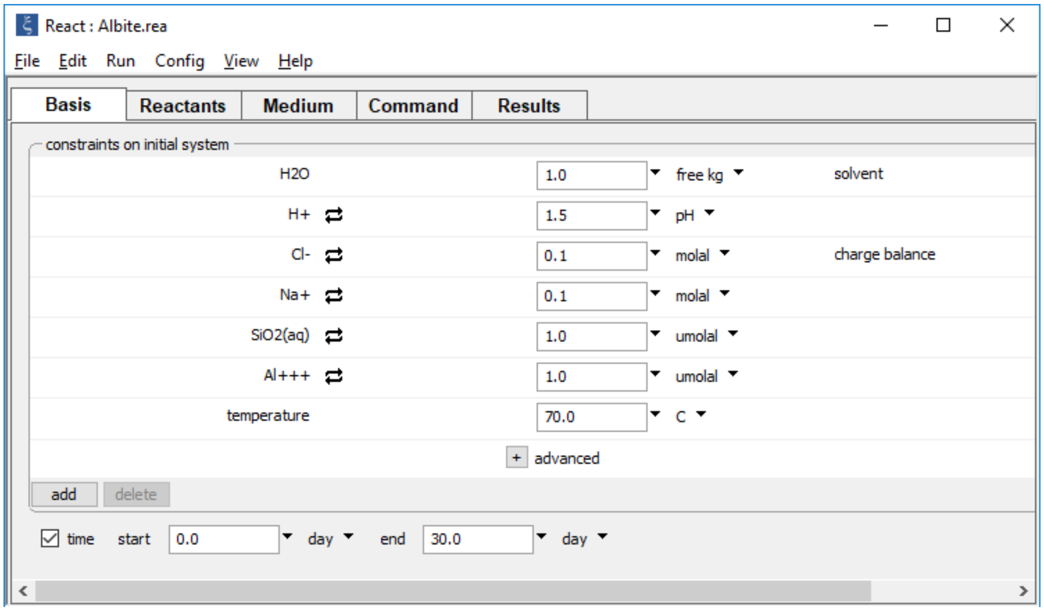


power

power(H+) 1.0 ○ activity ● molality

From the **Command** pane, use the `apower` or `mpower` (also `apow` or `mpow`) keyword within the `kinetic` command to set the species activity or molality, respectively, in the rate law. For an end member, keyword `apower` sets the term as activity and `xpower` sets it as mole fraction. For a gas species, keywords `fpower` and `ppower` set the term as fugacity or partial pressure, in bar. The generic keyword `power` sets the solvent or an aqueous species in terms of activity, an end member as activity, a gas as fugacity, and a surface species in terms of molality.

As an example, open “Albite.rea” and move to the **Basis** pane,



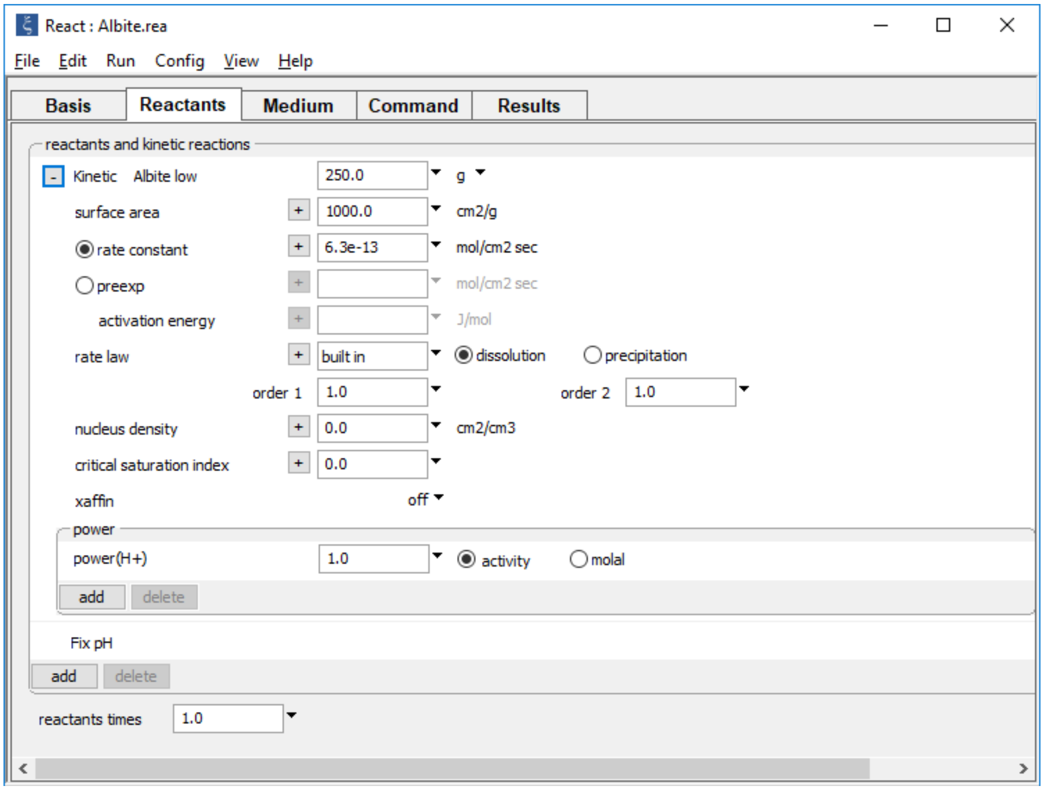
constraints on initial system

H2O	1.0	free kg	solvent
H+	1.5	pH	
Cl-	0.1	molal	charge balance
Na+	0.1	molal	
SiO2(aq)	1.0	umolal	
Al+++	1.0	umolal	
temperature	70.0	C	

☑ time start 0.0 day end 30.0 day

which defines an acidic fluid at 70°C. The simulation will span 30 days.

The kinetic rate law for albite dissolution is set on the **Reactants** pane,


React : Albite.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete

reactants and kinetic reactions

Kinetic Albite low 250.0 g

surface area 1000.0 cm2/g

● rate constant 6.3e-13 mol/cm2 sec

○ preexp mol/cm2 sec

activation energy J/mol

rate law built in ● dissolution ○ precipitation

order 1 1.0 order 2 1.0

nucleus density 0.0 cm2/cm3

critical saturation index 0.0

xaffin off

power

power(H+) 1.0 ● activity ○ molal

Fix pH

reactants times 1.0

Or, type

```
time begin = 0 day, end = 30 day
T = 70

pH = 1.5
0.1 molal Cl-
0.1 molal Na+
1 umolal SiO2(aq)
1 umolal Al+++
react 250 g of "Albite low"
kinetic "Albite low" rate_con = 6.3e-13, apower(H+) = 1
kinetic "Albite low" surface = 1000
fix pH
go
```

The rate law assumes that H^+ acts as a promoter, the activity of which is raised to a power of one. The example includes a single promoting species, but you can set for each rate law as many promoting and inhibiting species as you desire.

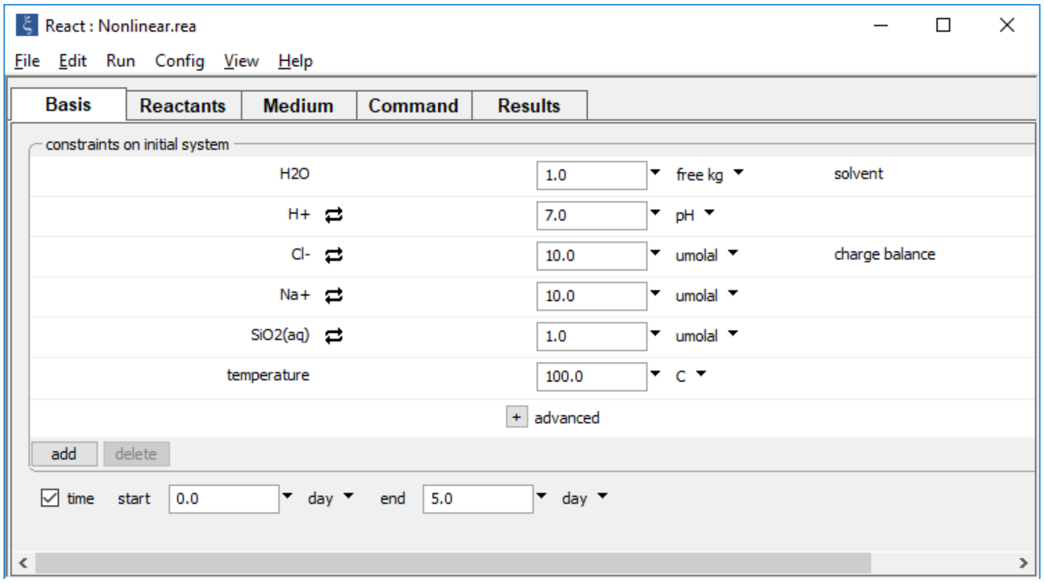
4.2.2 Nonlinear rate laws

React can also treat rate laws of the nonlinear form

$$r_{\vec{k}} = \operatorname{sgn}\left(1 - \frac{Q}{K}\right) A_S \, k_+ \prod_j \left(a_j | m_j\right)^{p_j} \left|1 - \left(\frac{Q}{K}\right)^{\omega}\right|^{\Omega} \tag{4.4}$$

Here, sgn is a function returning −1 or +1, depending on the sign of its argument, and variables ω and Ω are the “orders” of the rate law, which define the nonlinearity. When both orders are set to 1, the form of the rate law reduces to the previous form.

Set ω and Ω with keywords “order1” and “order2”. As an example, open “Nonlinear.rea”. The **Basis** pane


React : Nonlinear.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results

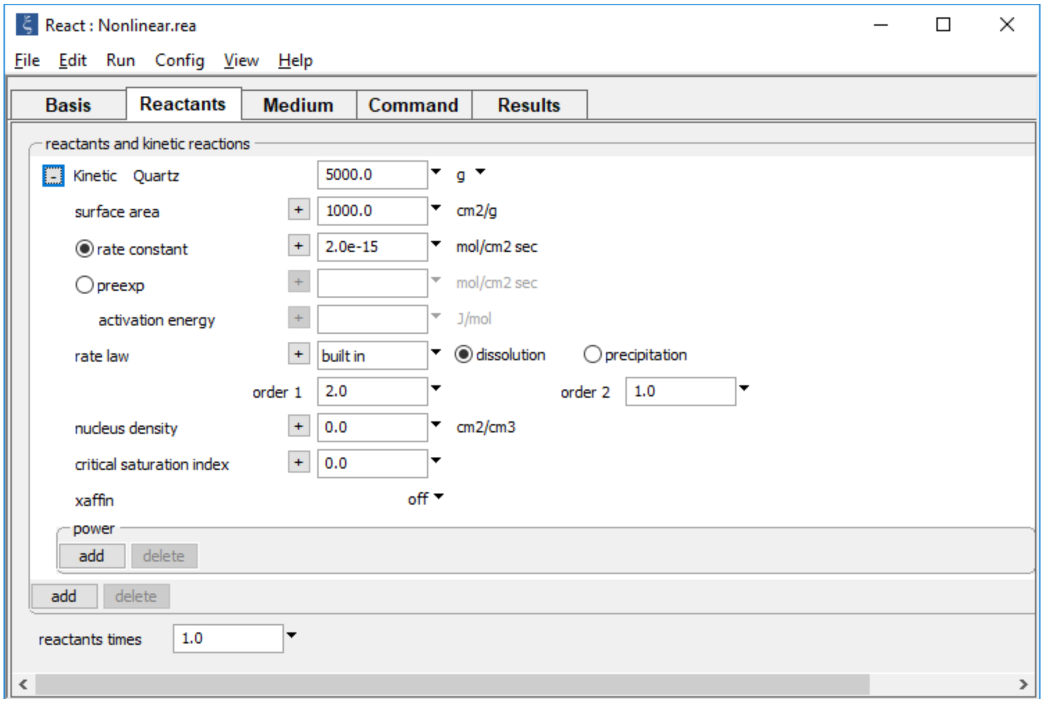
constraints on initial system

H2O	1.0	free kg	solvent
H+ ⇄	7.0	pH	
Cl- ⇄	10.0	umolal	charge balance
Na+ ⇄	10.0	umolal	
SiO2(aq) ⇄	1.0	umolal	
temperature	100.0	C	

☑ time start 0.0 day end 5.0 day

sets the initial fluid composition and the time span of the simulation.

The **Reactants** pane



reactants and kinetic reactions

Kinetic Quartz 5000.0 g

surface area 1000.0 cm2/g

● rate constant 2.0e-15 mol/cm2 sec

○ preexp mol/cm2 sec

activation energy J/mol

rate law built in ● dissolution ○ precipitation

order 1 2.0 order 2 1.0

nucleus density 0.0 cm2/cm3

critical saturation index 0.0

xaffin off

power

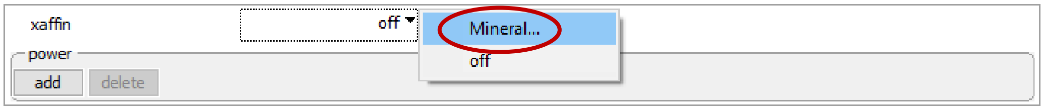
reactants times 1.0

describes the dissolution of quartz at 100°C, taking a value of 2 for ω and 1 for Ω.

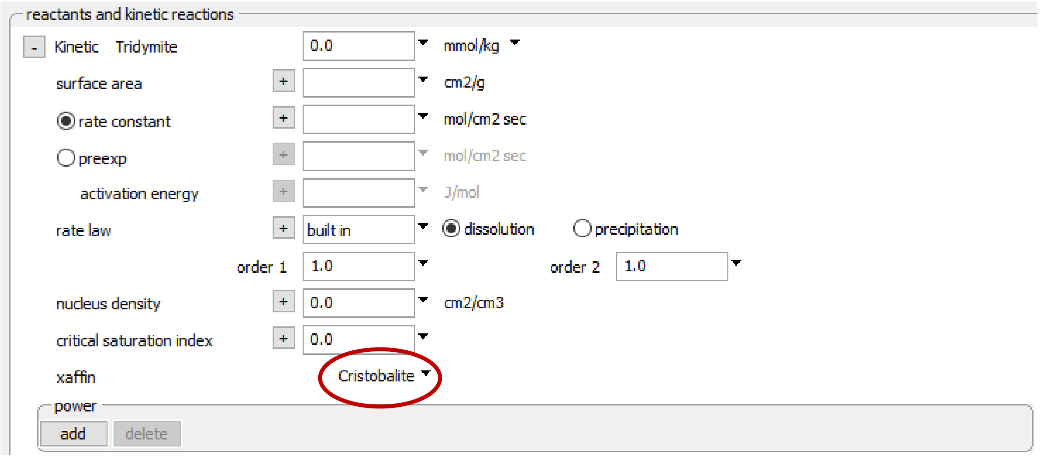
4.2.3 Cross-affinity rate laws

React supports the use of “cross-affinity” rate laws, in which the saturation state (i.e., *Q/K*) of one mineral is used to calculate the reaction rate of another mineral (or of an end member). Rate laws of this form are useful for studying problems such as glass dissolution, where reaction of an altered surface layer appears to control the overall dissolution rate.

To set the cross-affinity option, choose a mineral from the pulldown next to “xaffin”.



For a purely hypothetical example, the cross-affinity setting in tridymite's rate law below


reactants and kinetic reactions

Kinetic Tridymite 0.0 mmol/kg

surface area cm2/g

● rate constant mol/cm2 sec

○ preexp mol/cm2 sec

activation energy J/mol

rate law built in ● dissolution ○ precipitation

order 1 1.0 order 2 1.0

nucleus density 0.0 cm2/cm3

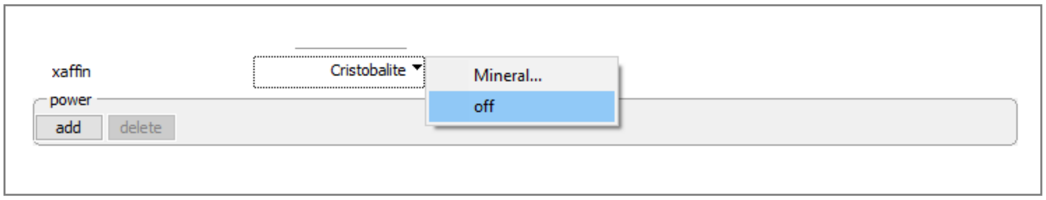
critical saturation index 0.0

xaffin Cristobalite

causes the program to calculate the reaction rate of tridymite according to the fluid's saturation state with respect to the mineral cristobalite. Alternatively, type

```
kinetic Tridymite xaffin = Cristobalite
```

Selecting "off"


xaffin Cristobalite

or typing

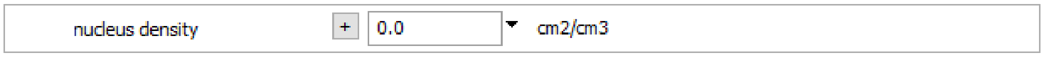
```
kinetic Tridymite xaffin = OFF
```

turns off the option.

4.2.4 Nucleation

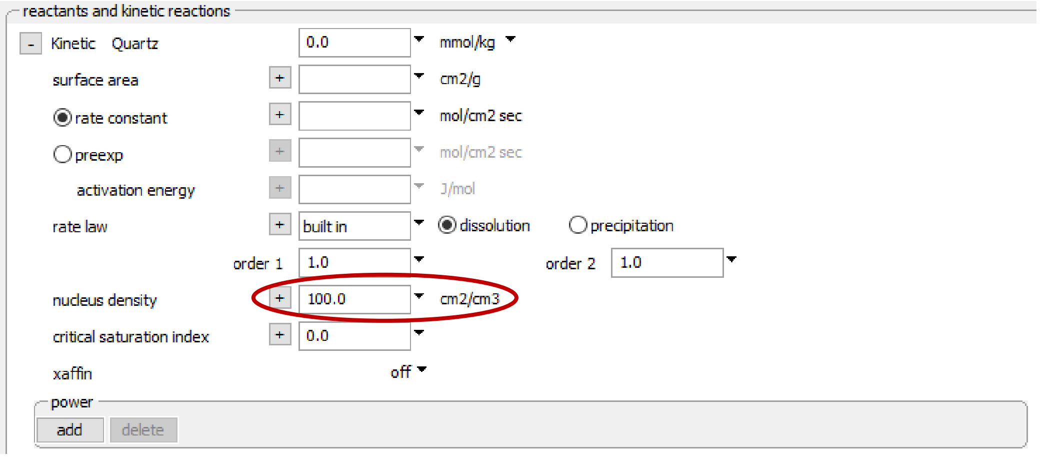
You can set for any kinetic mineral or end member a simple description of nucleation. This feature allows minerals not present in the chemical system to form according to kinetic rate laws. In the absence of a treatment for nucleation, a new kinetic mineral would not be able to form because its surface area A_S would be 0, leading to null reaction rates according to the rate laws.

To set the surface area (in cm^2 cm^{-3} fluid volume) available in the chemical system for nucleation whenever a mineral is supersaturated, enter a value for "nucleus density" in the kinetic rate law for that mineral.



nucleus density + 0.0 cm2/cm3

This option prescribes a minimum value for a supersaturated mineral's A_S over the calculation. For example, the setting below



reactants and kinetic reactions

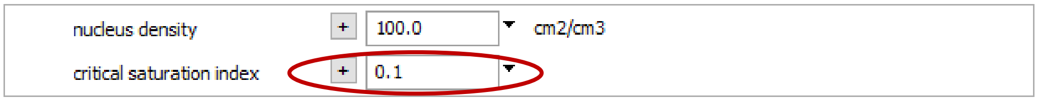
- Kinetic Quartz 0.0 mmol/kg
surface area + cm2/g
● rate constant + mol/cm2 sec
○ preexp + mol/cm2 sec
activation energy + J/mol
rate law + built in ● dissolution ○ precipitation
order 1 1.0 order 2 1.0
nucleus density + 100.0 cm2/cm3
critical saturation index + 0.0
xaffin off
power
add delete

or the command

```
kinetic Quartz nucleus = 100
```

prescribes that, per cm^3 of fluid, 100 cm^2 of surface area is available for the growth of quartz. By default, the nucleation area is 0.

In addition, you can set a critical saturation index (log Q/K) for a mineral or end member. In this case, the mineral's nucleation area is available only when its saturation index exceeds the prescribed value. Set this option using "critical saturation index"



nucleus density + 100.0 cm2/cm3
critical saturation index + 0.1

or the `CritSI` keyword.

```
kinetic Quartz nucleus = 100, CritSI = 0.1
```

The example above makes the nucleation area for quartz available only when the mineral has a saturation state ≥ 0.1. The default value for this option is 0, which means that the prescribed nucleation area is available whenever the mineral is supersaturated.

4.3 Forward and reverse reactions

In **React**, you most commonly set a kinetic law describing the forward rate of a reaction. In this case, a positive $r_{\vec{k}}$ indicates a mineral or end member is dissolving, a complex dissociating, a gas dissolving into the fluid, or a redox or microbial reaction proceeding from left to right. You can alternatively specify that evaluating the rate law gives a positive result for the reverse reaction: precipitation, association, gas exsolution, or right-to-left progress.

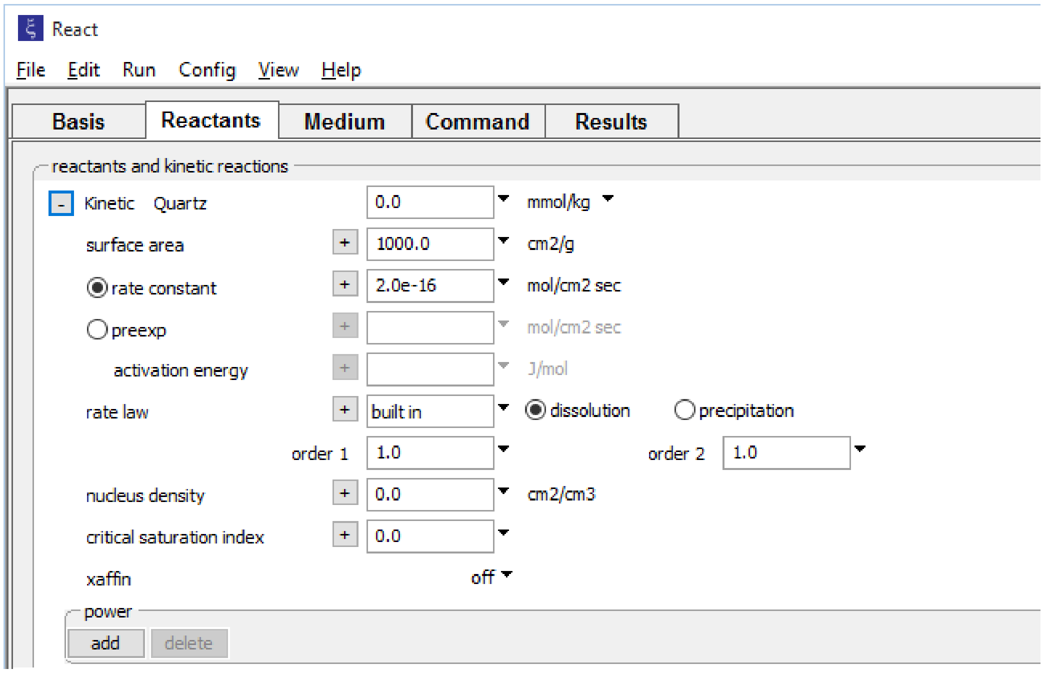
The dissolution of quartz

$$\mathrm{SiO_2(qtz)} \rightarrow \mathrm{SiO_2(aq)} \tag{4.5}$$

for example, might be described by the rate law

$$r_{\vec{k}} = A_S\, k_+ \left(1 - \frac{Q}{K}\right) \tag{4.6}$$

where $r_{\vec{k}}$ is dissolution rate, A_S is surface area, k_+ is the rate constant, and Q and K are the reaction's activity product and equilibrium constant. Taking k_+ at 70°C to be 2×10^{-16} mol $\mathrm{cm^{-2}}$ s and A_S as 10^3 $\mathrm{cm^2\ g^{-1}}$, you can set this rate law from the **Reactants** pane as shown below



reactants and kinetic reactions

Kinetic Quartz 0.0 mmol/kg

surface area 1000.0 cm2/g

● rate constant 2.0e-16 mol/cm2 sec

○ preexp mol/cm2 sec

activation energy J/mol

rate law built in ● dissolution ○ precipitation

order 1 1.0 order 2 1.0

nucleus density 0.0 cm2/cm3

critical saturation index 0.0

xaffin off

power

or by typing

```
kinetic Quartz forward rate_con = 2e-16 surf = 1000
```

Selection of the "dissolution" option specifies that the law describes the dissolution (forward) reaction, which is the default behavior.

If we write the reverse reaction, for precipitation,

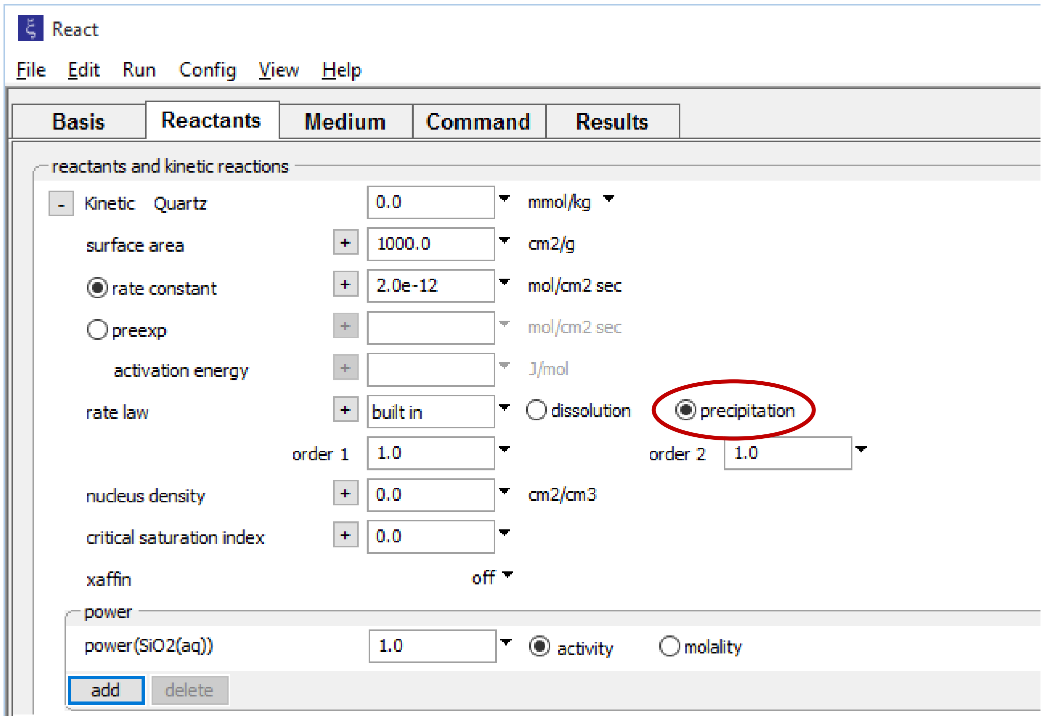
$$\mathrm{SiO_2(qtz)} \ \leftarrow\ \mathrm{SiO_2(aq)} \tag{4.7}$$

the activity product $\overleftarrow{Q} = 1/Q$ and the equilibrium constant $\overleftarrow{K} = 1/K$. Noting that $K = k_+/\overleftarrow{k}_+$ is the ratio of the rate constants for the forward and reverse reactions, and that $Q = a_{\mathrm{SiO_2\,(aq)}}$, the rate law for this reaction is

$$\overleftarrow{r}_k \ = \ -\overrightarrow{r}_k \ = \ A_S\,\overleftarrow{k}_+\,a_{\mathrm{SiO_2\,(aq)}}\left(1 - \frac{\overleftarrow{Q}}{\overleftarrow{K}}\right) \tag{4.8}$$

K at 70°C is $10^{-3:39}$, so $\overleftarrow{k}_+$ is 4.9×10^{-13}.

The input below



reactants and kinetic reactions

Kinetic Quartz 0.0 mmol/kg

surface area 1000.0 cm2/g

● rate constant 2.0e-12 mol/cm2 sec

○ preexp mol/cm2 sec

activation energy J/mol

rate law built in ○ dissolution ● precipitation

order 1 1.0 order 2 1.0

nucleus density 0.0 cm2/cm3

critical saturation index 0.0

xaffin off

power

power(SiO2(aq)) 1.0 ● activity ○ molality

or the command

```
kinetic Quartz reverse \
   rate_con = 2e-12 surf = 1000 apower(SiO2(aq)) = 1
```

sets this law, which is equivalent in function to the forward law described previously.

4.4 Kinetics of complexation and sorption

React can trace the association and dissociation of any number of aqueous complexes, as well as surface complexes. The program uses a built-in rate law of the form

$$r_{\vec{k}} = n_w \, k_+ \prod_j \left(a_j / m_j\right)^{p_j} \left(1 - \frac{Q}{K}\right) \tag{4.9}$$

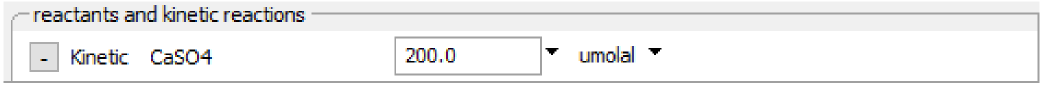
to calculate the reaction rate $r_{\vec{k}}$, in mol s^{-1}. Here, n_w is kg of solvent water, k_+ is the intrinsic rate constant (molal s^{-1}, in the absence of promoting and inhibiting species), p_j are powers applied to the activities a_j or molalities m_j of the **promoting and inhibiting species**, and Q and K are the reaction's ion activity product and equilibrium constant.

Reactions in **React** are written for species dissociation, so if you set a rate law for a forward reaction (see previous section, **Forward and reverse reactions**), $r_{\vec{k}}$ is positive when the complex decomposes. The built-in law can assume the **nonlinear form**,

$$r_{\vec{k}} = \operatorname{sgn}\left(1 - \frac{Q}{K}\right) n_w \, k_+ \prod_j \left(a_j / m_j\right)^{p_j} \left|1 - \left(\frac{Q}{K}\right)^{\omega}\right|^{\Omega} \tag{4.10}$$

in which ω and Ω are the nonlinearity coefficients, or reaction orders. When you set either nonlinearity coefficient to a value other than 1, the program uses the nonlinear law. You can, of course, set your own rate equation, rather than using the built-in law.

Set the initial concentration of a kinetic complex directly within its entry on the **Reactants** pane



reactants and kinetic reactions

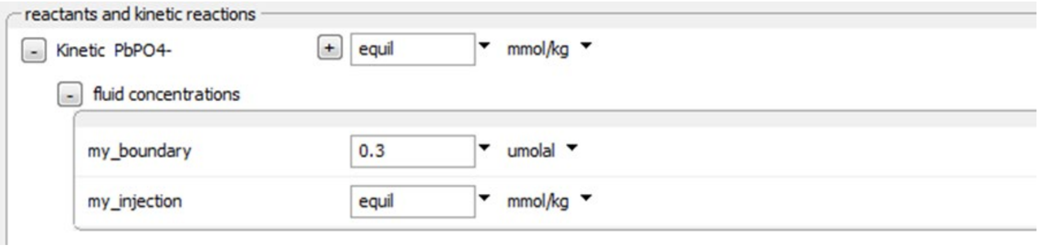
- Kinetic CaSO4 200.0 umolal

or by typing

```
kinetic CaSO4 200 umolal
```

The input specifies a 200 μmolal concentration for the $CaSO_4$ ion pair, at the beginning of the run. Note that the basis must be set to contain at least this much total calcium and sulfate, or the program will not be able to distribute mass. If you do not set an initial concentration, or set it to `?`, the program will take the complex at the onset of the run to be in equilibrium with the initial fluid.

In **X1t** and **X2t**, click + to expand the "fluid concentrations" box, then set the concentration of the kinetic complex in each boundary fluid:



reactants and kinetic reactions

- Kinetic PbPO4- + equil mmol/kg

- fluid concentrations

my_boundary 0.3 umolal

my_injection equil mmol/kg

Alternatively, type

```
kinetic PbPO4- my_boundary = 0.3 umolal, my_injection = ?
```

As with the initial fluid, if you omit this setting, or specify an argument of ?, the complex will be set to its equilibrium concentration.

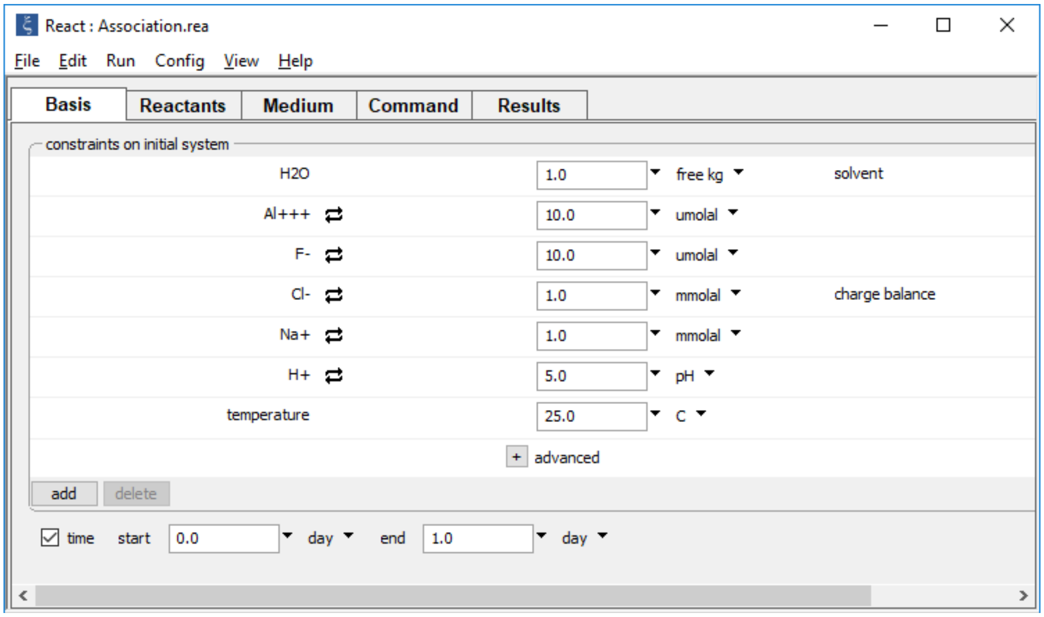
As an example, the complexation of aluminum and fluoride ions

$$AlF^{++} \leftarrow Al^{+++} + F^{-} \tag{4.11}$$

can be described by the rate law

$$\frac{dm_{AlF^{++}}}{dt} = \frac{\overleftarrow{r}_k}{n_w} = \overleftarrow{k}_+ \, m_{Al^{+++}} \, m_{F^-} \left(1 - \frac{\overleftarrow{Q}}{\overleftarrow{K}}\right) \tag{4.12}$$

Open the “Association.rea” input file and move to the **Basis** pane,



constraints on initial system

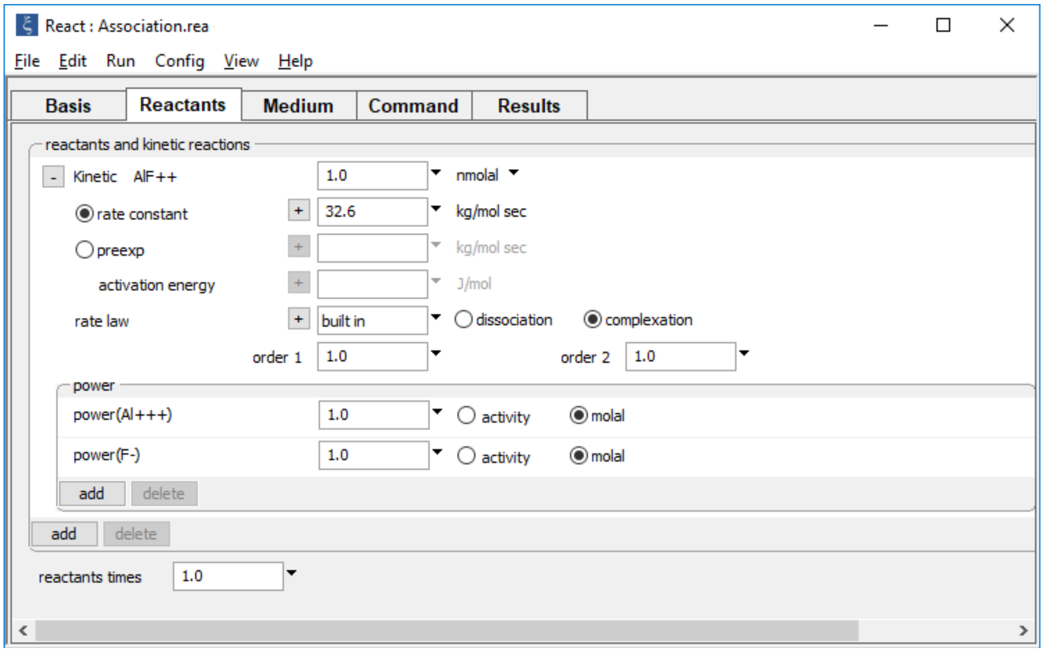
H2O	1.0	free kg	solvent
Al+++	10.0	umolal	
F-	10.0	umolal	
Cl-	1.0	mmolal	charge balance
Na+	1.0	mmolal	
H+	5.0	pH	
temperature	25.0	C	

+ advanced

☑ time start 0.0 day end 1.0 day

which defines the initial system, including the total amount of aluminum and fluoride dissolved in the water.

On the **Reactants** pane, the kinetic rate law



reactants and kinetic reactions

Kinetic AlF++ 1.0 nmolal

● rate constant 32.6 kg/mol sec

○ preexp kg/mol sec

activation energy J/mol

rate law built in ○ dissociation ● complexation

order 1 1.0 order 2 1.0

power

power(Al+++) 1.0 ○ activity ● molal

power(F-) 1.0 ○ activity ● molal

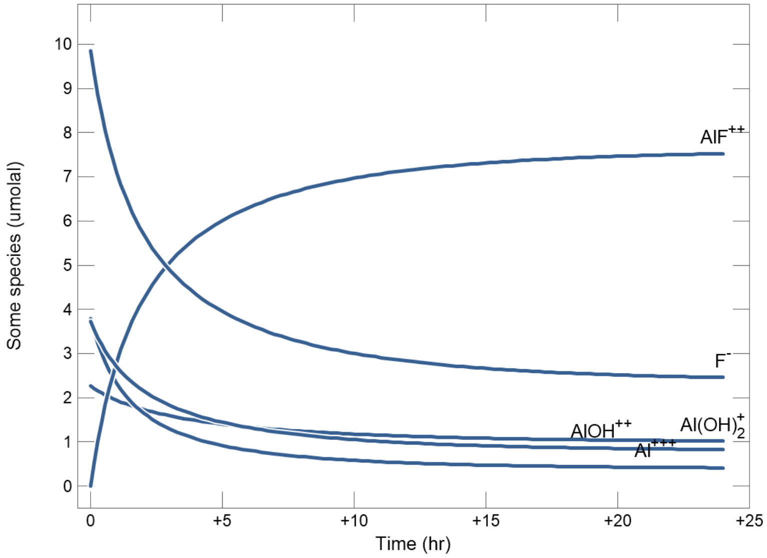
reactants times 1.0

causes the program to trace the complexation of alumina and fluoride over the course of a day, assuming a rate constant of 32.6 molal^{-1} s^{-1}

From the **Command** pane, type

```
Al+++ = 10 umolal
F- = 10 umolal
Cl- = 1 mmolal
Na+ = 1 mmolal
pH = 5

kinetic AlF++ 1 nmolal, reverse, rate_con = 32.6, \
   mpower(Al+++) = 1, mpower(F-) = 1
suppress AlF2+, AlF3, AlF4-, AlF5--
precip off
time end 1 day
go
```

The calculation result shows the association of the AlF^{++} ion pair, calculated according to a kinetic rate law.



Take as a second example the desorption of calcium

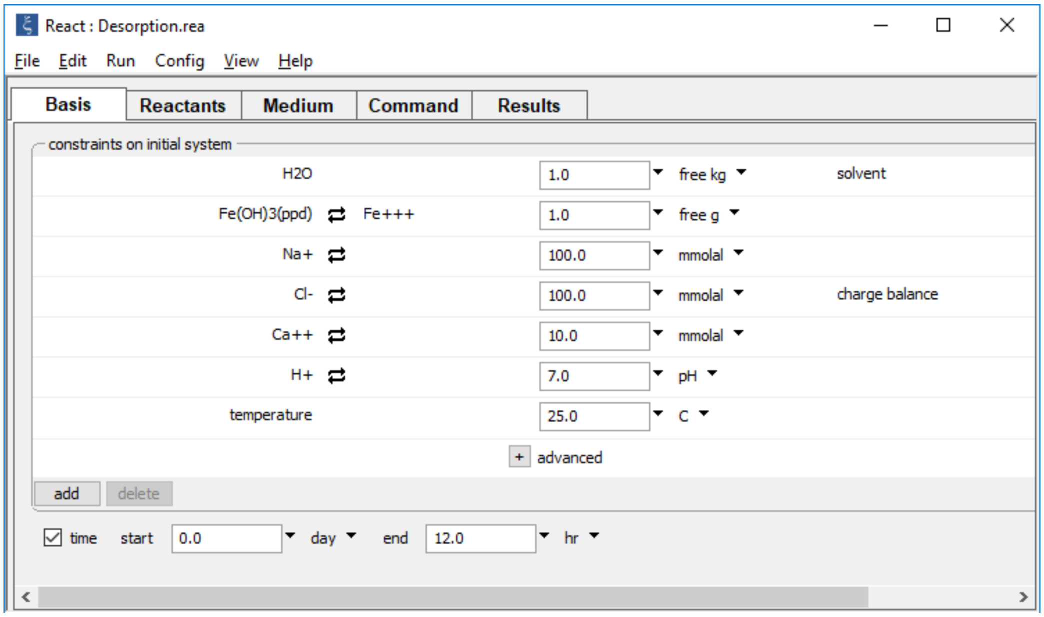
$$>(w)FeOCa^+ + H^+ \rightarrow >(w)FeOH + Ca^{++} \tag{4.13}$$

in a pH-buffered fluid, due to the decomposition of the $>(w)FeOCa^+$ surface species. Assume the reaction is described by the rate equation

$$-\frac{dm_{>(w)FeOCa^+}}{dt} = \frac{r_{\vec{k}}}{n_w} = k_+ \, m_{>(w)FeOCa^+} \left(1 - \frac{Q}{K}\right) \tag{4.14}$$

Take a rate constant k_+ of 2×10^{-4} s^{-1}, and the initial concentration of the surface complex to be 1500 μmolal.

Open “Desorption.rea” and go to the **Basis** pane,



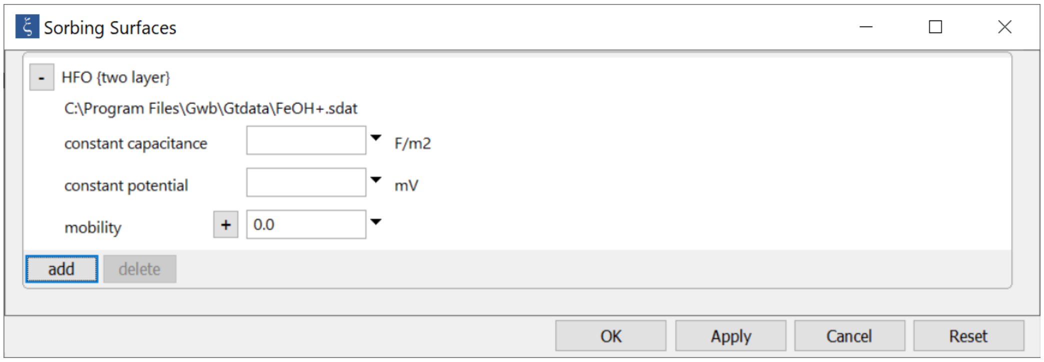
constraints on initial system

H2O		1.0	free kg	solvent
Fe(OH)3(ppd) ⇄	Fe+++	1.0	free g	
Na+ ⇄		100.0	mmolal	
Cl- ⇄		100.0	mmolal	charge balance
Ca++ ⇄		10.0	mmolal	
H+ ⇄		7.0	pH	
temperature		25.0	C	

☑ time start 0.0 day end 12.0 hr

where the initial fluid in equilibrium with a ferric hydroxide is set up. The simulation is set to span 12 hours.

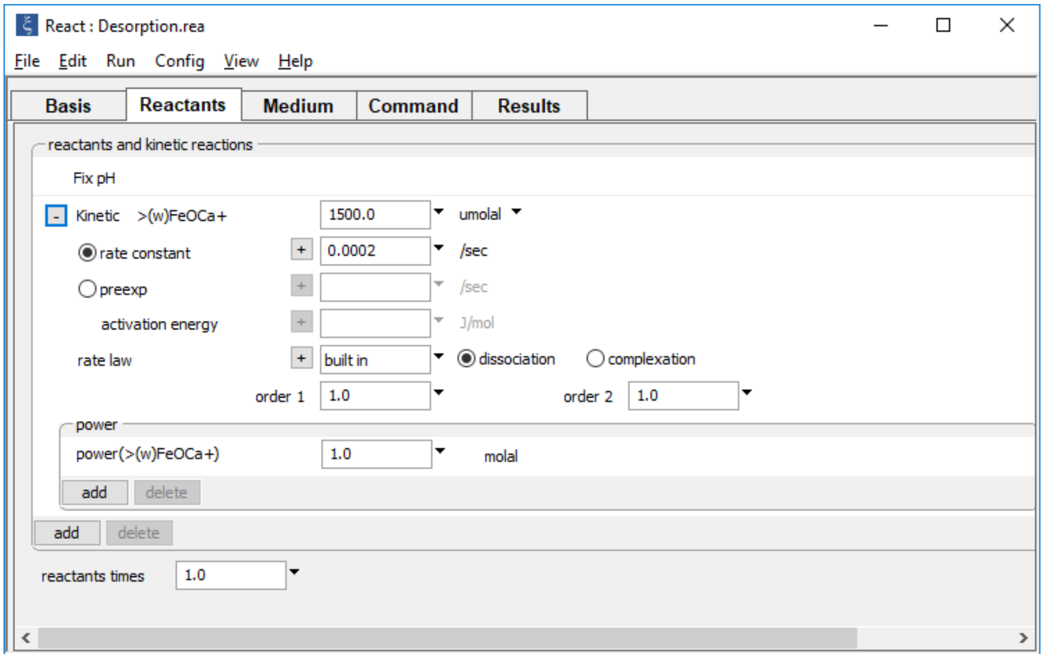
Go to **Config → Sorbing Surfaces…**



HFO (two layer)
C:\Program Files\Gwb\Gtdata\FeOH+.sdat
constant capacitance F/m2
constant potential mV
mobility 0.0

to see that the Dzombak and Morel dataset for ion complexation with the hydrous ferric oxide is loaded.

Move to the **Reactants** pane.



React : Desorption.rea

reactants and kinetic reactions

Fix pH

Kinetic >(w)FeOCa+ 1500.0 umolal

● rate constant 0.0002 /sec

○ preexp /sec

activation energy J/mol

rate law built in ● dissociation ○ complexation

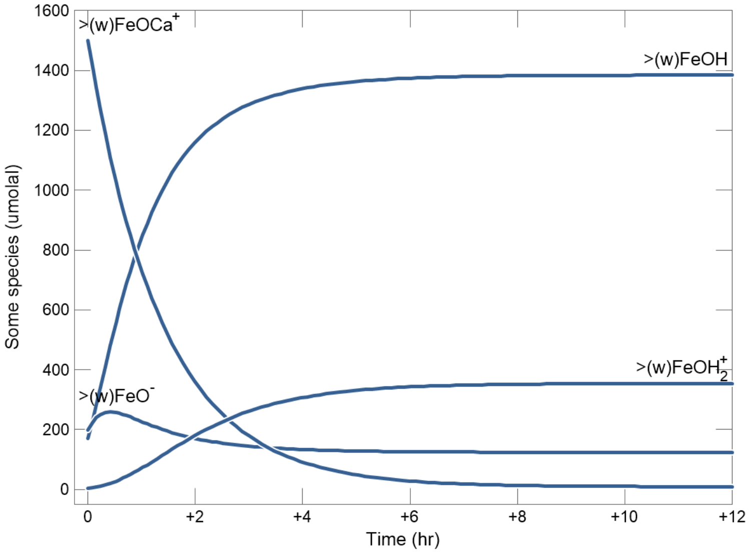
order 1 1.0 order 2 1.0

power

power(>(w)FeOCa+) 1.0 molal

reactants times 1.0

Set a pH buffer and the rate law for calcium desorption. Take a rate constant k_+ of 2×10^{-4} s^{-1}, and the initial concentration of the surface complex to be 1500 μmolal. The results are show here.



4.5 Gas transfer kinetics

React can account for the kinetics of gas transfer between a fluid and an external gas reservoir, such as the atmosphere. The program calculates the rate $r_{\vec{k}}$ (mol s^{-1}) at which a kinetic gas $A_{\vec{k}}$ dissolves into a fluid using the built-in equation

$$r_{\vec{k}} = n_w A_{sp} k_+ \left(f_{ext} - f_{\vec{k}}\right) \tag{4.15}$$

Here, n_w is solvent mass (kg), k_+ is the rate constant (mol cm^{-2} s^{-1}), A_{sp} is the specific contact area (cm^2 [kg solvent]$^{-1}$) between reservoir and fluid, and f_{ext} and $f_{\vec{k}}$ are the fugacities of $A_{\vec{k}}$ in the external reservoir and the fluid. Where gas exsolves from the fluid, $r_{\vec{k}}$ is negative.

Parallel to the other kinetic features, you can specify **promoting and inhibiting species**, in which case, the law takes the form

$$r_{\vec{k}} = n_w A_{sp} k_+ \prod_j \left(a_j / m_j\right)^{p_j} \left(f_{ext} - f_{\vec{k}}\right) \tag{4.16}$$

You can set a **nonlinear law**

$$r_{\vec{k}} = \operatorname{sgn}\left(f_{ext} - f_{\vec{k}}\right) n_w A_{sp} k_+ \prod_j \left(a_j / m_j\right)^{p_j} \left|f_{ext} - f_{\vec{k}}\right|^{\Omega} \tag{4.17}$$

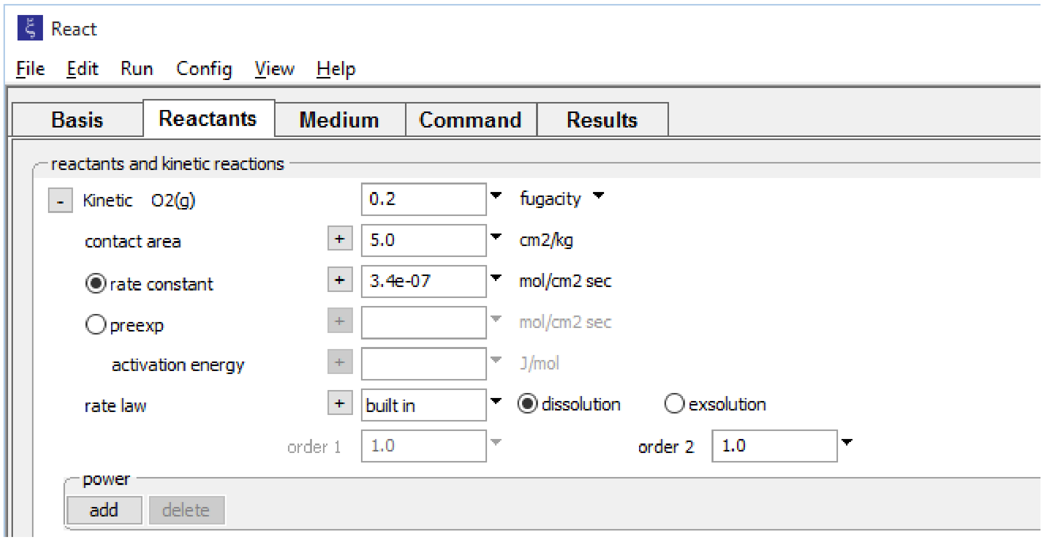
in which Ω is the nonlinearity coefficient (there is no ω in the gas transfer model). And, as before, you can specify your own rate equation, instead of using the built-in law.

To set up kinetic gas transfer, you generally set the gas' fugacity f_{ext} in the external reservoir, or its partial pressure P_{ext} there; the model converts a pressure value to fugacity internally. You furthermore set the contact area between the fluid and external reservoir, in units of cm^2 (kg solvent)$^{-1}$.

On the **Reactants** pane, at the top of the field for a gas transfer reaction, you constrain the external reservoir by entering a value and choosing either "fugacity" or a unit of pressure from the pulldown. Working from the command line, use the `f_ext` keyword to set external fugacity, and `p_ext` plus an optional unit to set partial pressure; the default unit is bar.

If you do not specify a value for external fugacity, the program takes it as the gas' fugacity in the initial system, at the onset of reaction. Finally, you set the contact area on the next line in the **Reactants** pane, or with the `contact` keyword on the command line.

Having added a kinetic gas reaction, the **Reactants** pane might look like



reactants and kinetic reactions

- Kinetic O2(g) 0.2 fugacity

contact area + 5.0 cm2/kg

● rate constant + 3.4e-07 mol/cm2 sec

○ preexp + mol/cm2 sec

activation energy + J/mol

rate law + built in ● dissolution ○ exsolution

order 1 1.0 order 2 1.0

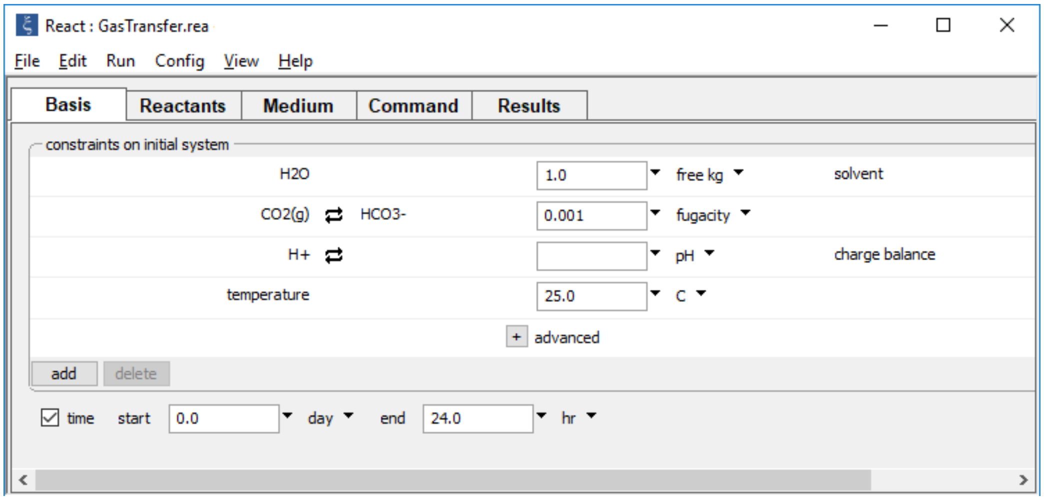
power

Alternatively, we might type the command

```
kinetic O2(g) f_ext = 0.2, contact = 5, rate_con = 3.4e-7
```

These cases set up a kinetic reaction for the transfer of O_2(g) between the fluid and an external reservoir in which f_{O_2} is 0.2. The contact area A_{sp} in this case is 5 cm^2 (kg solvent)$^{-1}$ and the rate constant k_+ is 3.4×10^{-7} mol cm^{-2} s^{-1}.

As a worked example, we trace the degassing of CO_2 from a well-mixed surface water. Open “GasTransfer.rea” and go to the **Basis** pane



constraints on initial system

H2O 1.0 free kg solvent

CO2(g) ⇄ HCO3- 0.001 fugacity

H+ ⇄ pH charge balance

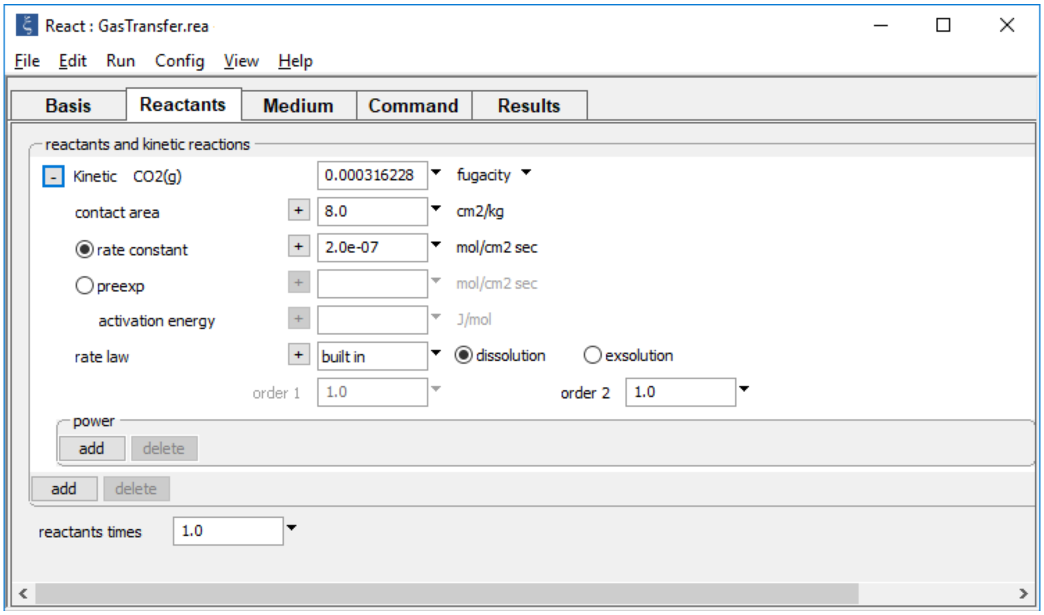
temperature 25.0 C

+ advanced

☑ time start 0.0 day end 24.0 hr

to see the fugacity in the water initially is 10^{-3}.

Move to the **Reactants** pane,



React : GasTransfer.rea

File Edit Run Config View Help

Basis | Reactants | Medium | Command | Results

reactants and kinetic reactions

Kinetic CO2(g) 0.000316228 fugacity

contact area 8.0 cm2/kg

● rate constant 2.0e-07 mol/cm2 sec

○ preexp mol/cm2 sec

activation energy J/mol

rate law built in ● dissolution ○ exsolution

order 1 1.0 order 2 1.0

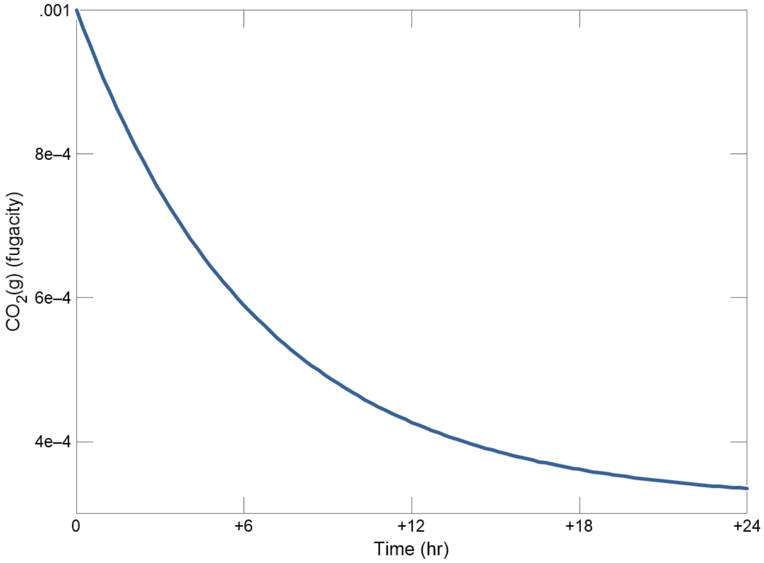
power

add delete

add delete

reactants times 1.0

and set the details of the kinetic rate law. The f_{CO_2} in the atmosphere is $10^{-3.5}$. The water is 125 cm deep, so A_{sp} is 8 cm^2 (kg solvent)$^{-1}$, and assume a rate constant k_+ of 2×10^{-7} mol cm^{-2} s^{-1}. The results are shown here.



4.6 Kinetics of redox reactions

You can use **React** to trace the progress over time of redox reactions, as described by kinetic rate laws. There is no limit to the number of kinetic redox reactions you can consider in a single simulation.

The kinetic rate law for a redox reaction is often presented in the form,

$$r_{\vec{k}} = n_w \, k_+ \prod_j (a_j | m_j)^{p_j} \tag{4.18}$$

where n_w is the mass of solvent water (kg), and k_+ is the intrinsic rate constant (expressed in molal s^{-1}). The product function $\prod$ in this equation accounts for the **promoting and inhibiting species** j in the rate law; the activity or molality of each such species appears raised to its power p_j. Each of these species may be an aqueous species, mineral, end member (in which case it is represented by its activity or mole fraction), surface species (by molality), gas (by fugacity or partial pressure), or solvent (by activity).

If this equation is generalized to account for reverse as well as forward reaction, it takes the form,

$$r_{\vec{k}} = n_w \, k_+ \prod_j (a_j | m_j)^{p_j} \left(1 - \frac{Q}{K}\right) \tag{4.19}$$

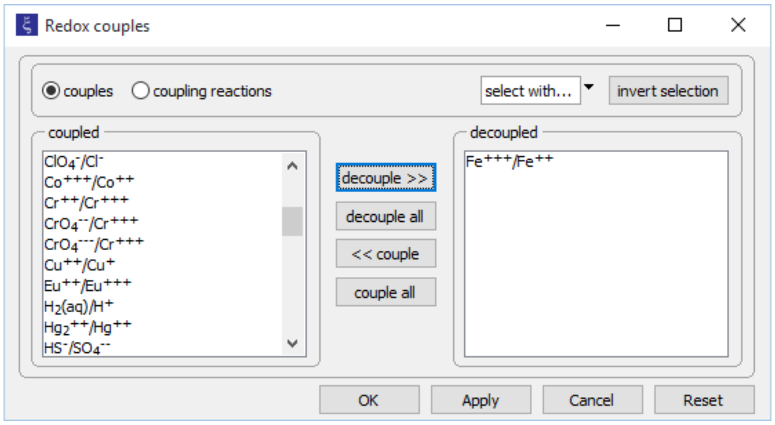
where Q and K are the reaction's activity product and equilibrium constant. This equation may in turn be generalized into a **nonlinear form**,

$$r_{\vec{k}} = \operatorname{sgn}\left(1 - \frac{Q}{K}\right) n_w \, k_+ \prod_j (a_j | m_j)^{p_j} \left|1 - \left(\frac{Q}{K}\right)^{\omega}\right|^{\Omega} \tag{4.20}$$

where ω and Ω are the nonlinearity coefficients (the rate law orders). This is the general form of the rate law that **React** carries for redox reactions.

To set up a kinetic redox reaction, begin by decoupling the redox couple in question, so that the reaction is not in equilibrium initially. Then set up the kinetic reaction and the form of the rate law.

As an example, go to the **Config → Redox couples…** dialog and decouple the Fe^{+++}/Fe^{++} pair,

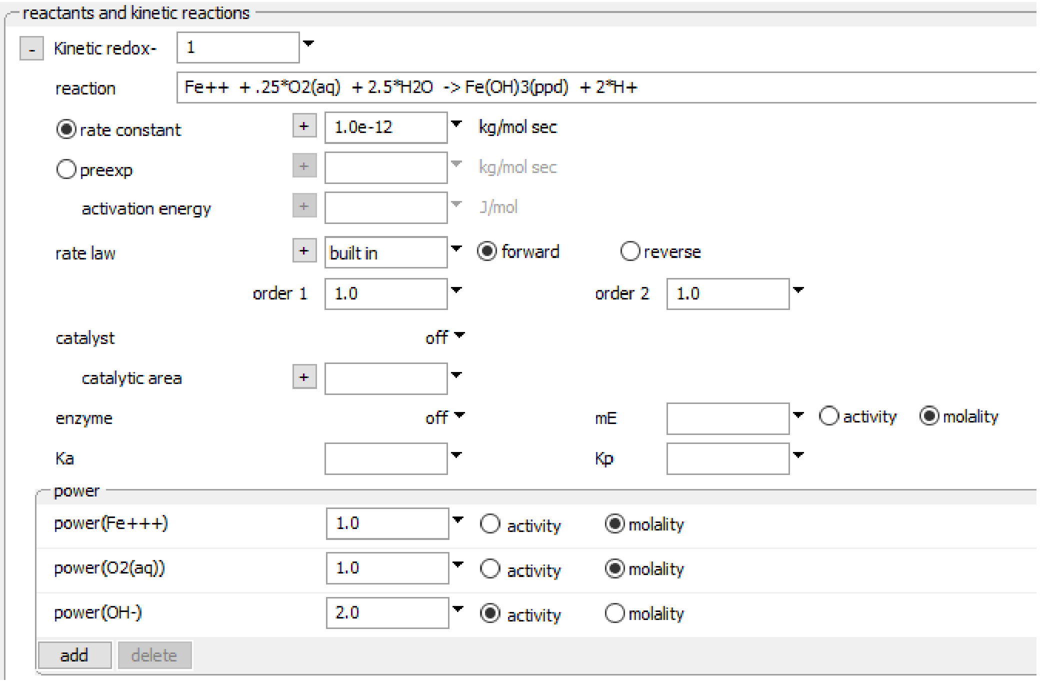


● couples ○ coupling reactions

coupled:
ClO_4^-/Cl^-
Co^{+++}/Co^{++}
Cr^{++}/Cr^{+++}
CrO_4^{--}/Cr^{+++}
CrO_4^{---}/Cr^{+++}
Cu^{++}/Cu^{+}
Eu^{++}/Eu^{+++}
$H_2(aq)/H^{+}$
Hg_2^{++}/Hg^{++}
HS^-/SO_4^{--}

decoupled:
Fe^{+++}/Fe^{++}

then move to the **Reactants** pane and click add → **Kinetic** → **Redox reaction**, supplying the info below.



reactants and kinetic reactions

Kinetic redox- 1

reaction Fe++ + .25*O2(aq) + 2.5*H2O -> Fe(OH)3(ppd) + 2*H+

● rate constant 1.0e-12 kg/mol sec

○ preexp kg/mol sec

activation energy J/mol

rate law built in ● forward ○ reverse

order 1 1.0 order 2 1.0

catalyst off

catalytic area

enzyme off mE ○ activity ● molality

Ka Kp

power

power(Fe+++) 1.0 ○ activity ● molality

power(O2(aq)) 1.0 ○ activity ● molality

power(OH-) 2.0 ● activity ○ molality

Alternatively, type

```
decouple Fe+++
kinetic redox-1 \
   rxn = "Fe++ + 1/4 O2(aq) + 5/2 H2O -> Fe(OH)3(ppd) + 2 H+" \
   mpower(Fe++) = 1 mpower(O2(aq)) = 1 apower(OH-) = 2 \
   rate_con = 1e-12
```

Either way, these steps set a rate law

$$\frac{dM_{\mathrm{Fe}^{+++}}}{dt} = n_w\, k_+\, m_{\mathrm{Fe}^{++}}\, m_{\mathrm{O_2(aq)}}\, a^2_{\mathrm{OH}^-} \left(1 - \frac{Q}{K}\right) \qquad (4.21)$$

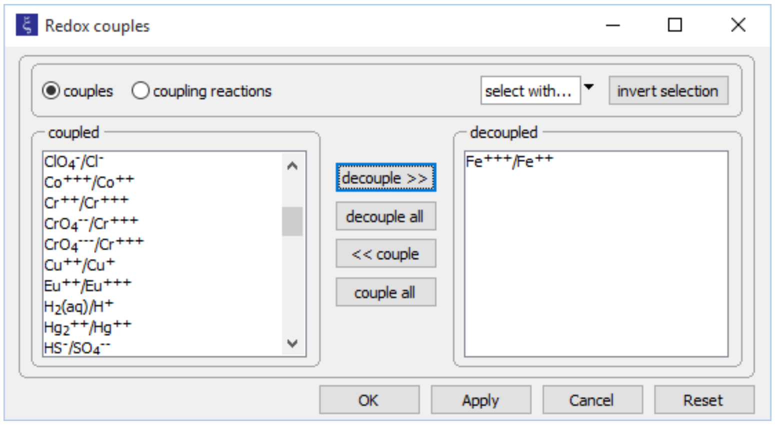
arbitrarily labeled "redox-1", for the redox reaction,

$$\mathrm{Fe^{++}} + {}^1\!/_4\,\mathrm{O_2(aq)} + {}^5\!/_2\,\mathrm{H_2O} \rightarrow \mathrm{Fe(OH)_3} + 2\,\mathrm{H^+} \qquad (4.22)$$

where $M_{\mathrm{Fe}^{+++}}$ is the mass in the system of component Fe^{+++} (mol), and k_+ is the rate constant, 10^{-12} molal^{-1} s^{-1}. Notice that the redox reaction is set explicitly as a character string. The program parses the string to derive the coefficients and species in the reaction.

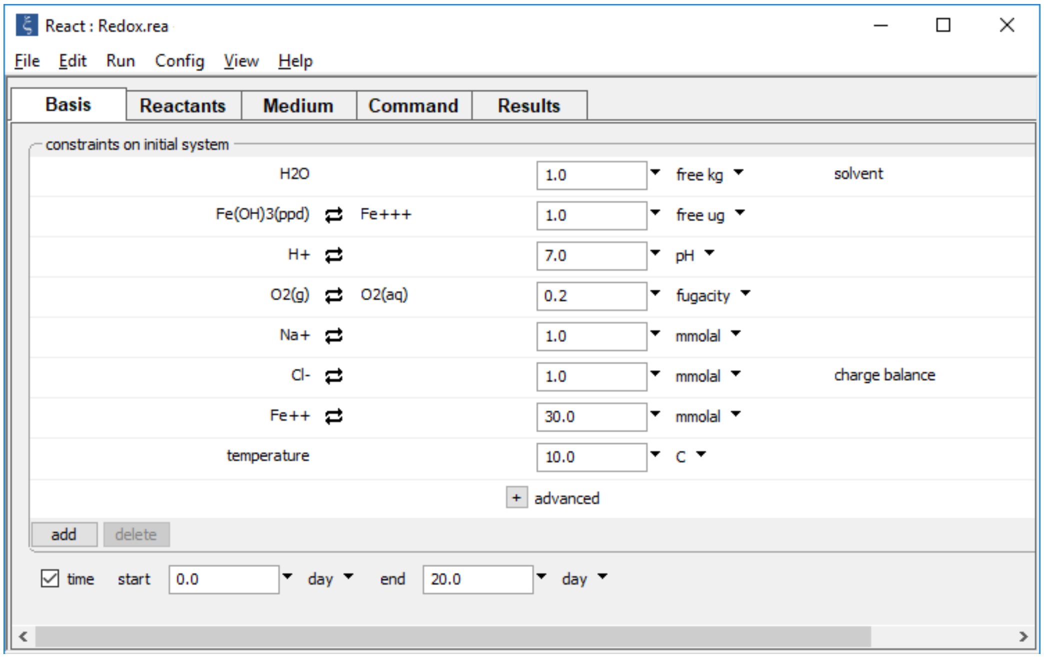
The following procedure models the oxidation of ferrous iron by molecular oxygen, assuming that the reaction produces ferric hydroxide, as might occur in a pH-buffered laboratory experiment left open to oxygen in the atmosphere.

Double-click the "Redox.rea" input file and go to the **Config → Redox couples…** dialog.



Here, the reaction between ferrous and ferric iron has been disabled. Iron in the simulation can therefore change redox state only by kinetic reaction.

The **Basis** pane



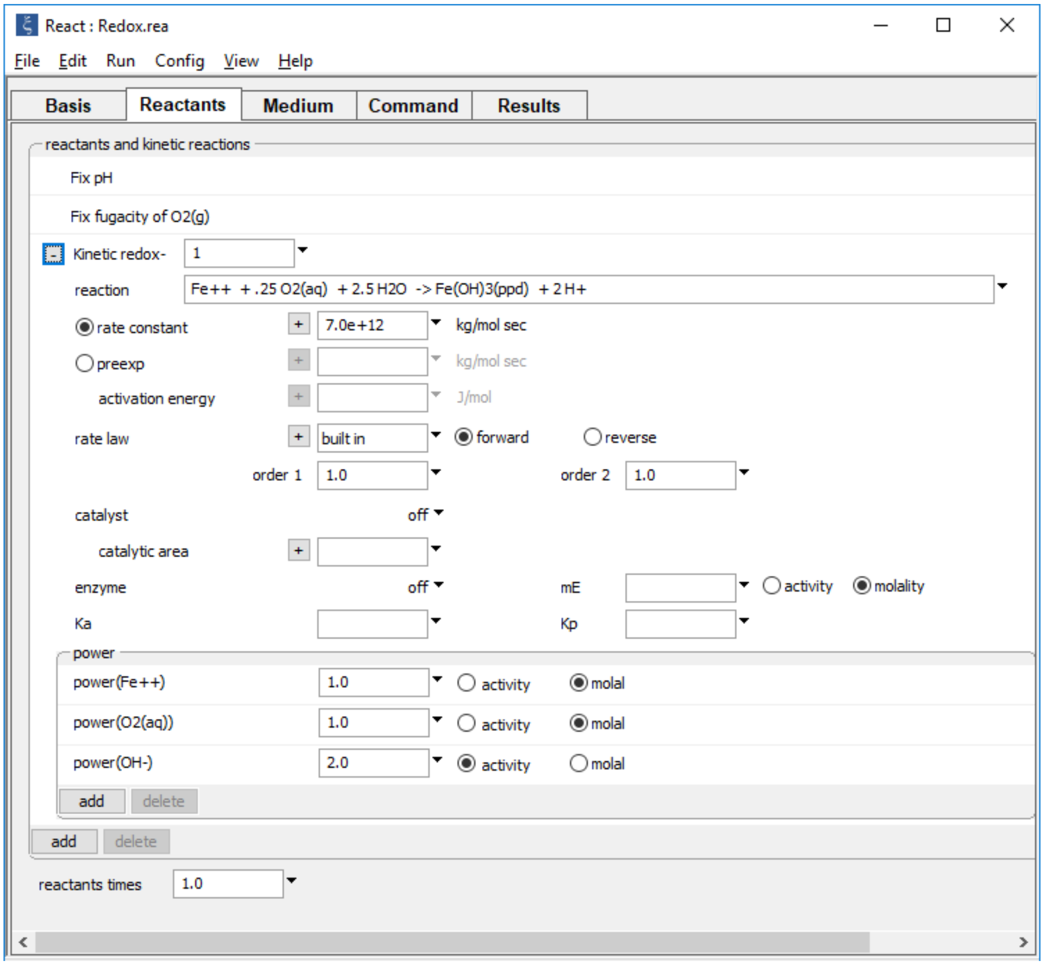
constraints on initial system

H2O			1.0	free kg	solvent
Fe(OH)3(ppd)	⇄	Fe+++	1.0	free ug	
H+	⇄		7.0	pH	
O2(g)	⇄	O2(aq)	0.2	fugacity	
Na+	⇄		1.0	mmolal	
Cl-	⇄		1.0	mmolal	charge balance
Fe++	⇄		30.0	mmolal	
temperature			10.0	C	

☑ time start 0.0 day end 20.0 day

shows the fluid's initial composition. The ferrous and ferric components are constrained separately, since the Fe^{+++}/Fe^{++} redox pair is decoupled.

Move to the **Reactants** pane.



reactants and kinetic reactions

Fix pH

Fix fugacity of O2(g)

Kinetic redox- 1

reaction Fe++ + .25 O2(aq) + 2.5 H2O -> Fe(OH)3(ppd) + 2 H+

● rate constant 7.0e+12 kg/mol sec

○ preexp kg/mol sec

activation energy J/mol

rate law built in ● forward ○ reverse

order 1 1.0 order 2 1.0

catalyst off

catalytic area

enzyme off mE ○ activity ● molality

Ka Kp

power

power(Fe++) 1.0 ○ activity ● molal

power(O2(aq)) 1.0 ○ activity ● molal

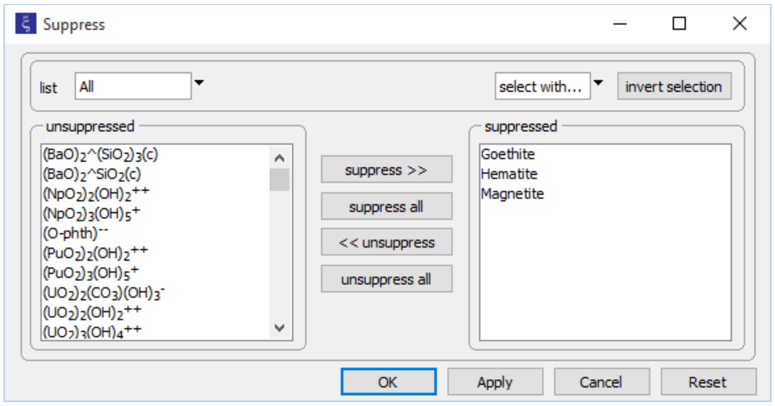
power(OH-) 2.0 ● activity ○ molal

reactants times 1.0

The "fix pH" and "fix fugacity of O2(g)" options simulate reaction in a pH-buffered laboratory experiment left open to oxygen in the atmosphere.

Click on the + button next to reactant "redox-1" to expand its entry. The reaction by which the ferrous iron component oxidizes has been written out explicitly as a character string that the program will read and interpret. If the kinetic reaction hadn't already been created, you would click on add → **Kinetic** → **Redox reaction** to do so.

To keep the discussion simple for the moment, the iron minerals Goethite, Hematite, and Magnetite are suppressed, each of which is more stable and slower to form than ferric hydroxide precipitate.



list: All

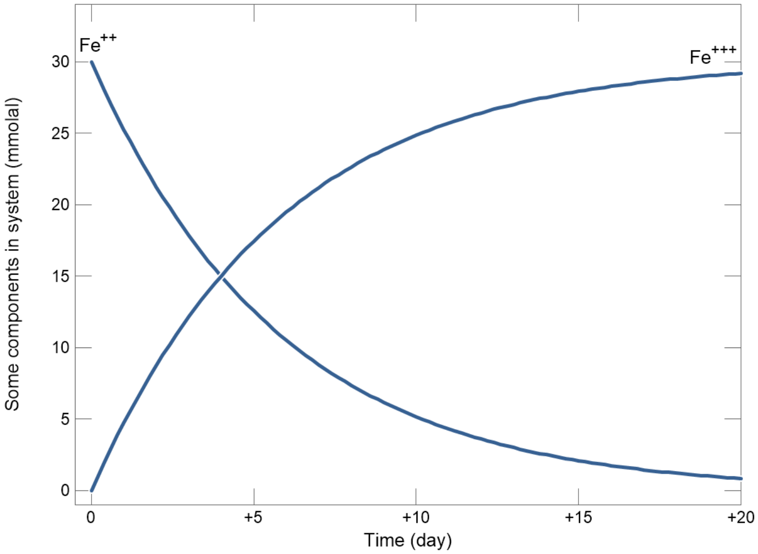
unsuppressed

$(BaO)_2$^$(SiO_2)_3(c)$
$(BaO)_2$^$SiO_2(c)$
$(NpO_2)_2(OH)_2^{++}$
$(NpO_2)_3(OH)_5^{+}$
$(O\text{-}phth)^{--}$
$(PuO_2)_2(OH)_2^{++}$
$(PuO_2)_3(OH)_5^{+}$
$(UO_2)_2(CO_3)(OH)_3^{-}$
$(UO_2)_2(OH)_2^{++}$
$(UO_2)_3(OH)_4^{++}$

suppressed

Goethite
Hematite
Magnetite

Select **Run → Go** to calculate the model. The figure below shows the results of modeling the oxidation of ferrous iron by dioxygen, according to a kinetic rate law.



React can also account for the effects of heterogeneous catalysis, enzymes, and biological activity on the rates of redox reactions; these features are described in the following sections.

4.6.1 Catalysis on mineral surfaces

React can account for the catalytic effects of mineral surfaces on **redox reactions**. The rate law in this case

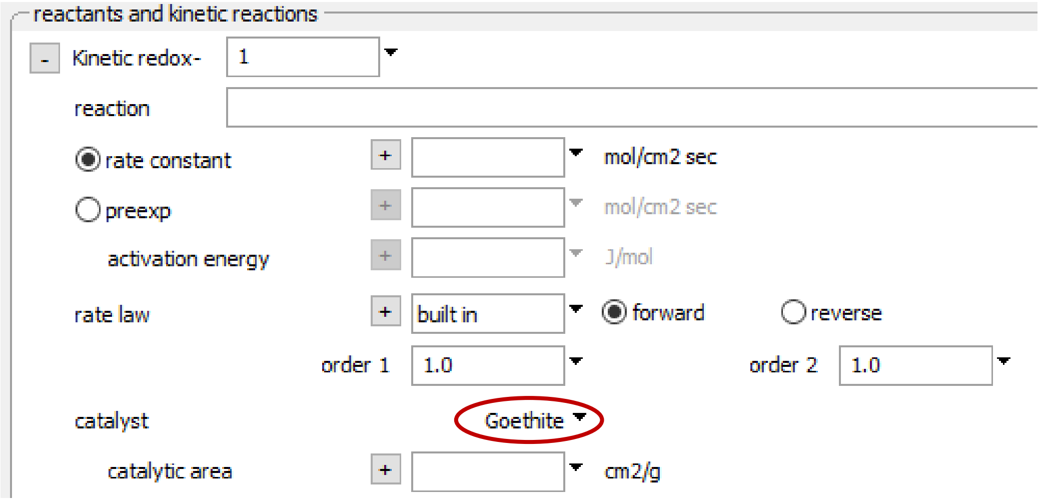
$$r_{\vec{k}} = A_S\, k_+ \prod_j (a_j | m_j)^{p_j} \left(1 - \frac{Q}{K}\right) \tag{4.23}$$

is of the form of the previous equation, with the additional term A_S, which represents the surface area (cm^2) of the catalyst. The rate constant k_+ here, in the absence of **promoting and inhibiting species**, is given in units of mol cm^{-2} s^{-1}. The rate law in **nonlinear form** is

$$r_{\vec{k}} = \operatorname{sgn}\left(1 - \frac{Q}{K}\right) A_S\, k_+ \prod_j (a_j | m_j)^{p_j} \left|1 - \left(\frac{Q}{K}\right)^{\omega}\right|^{\Omega} \tag{4.24}$$

Again, this law is the same as the previous nonlinear form, with the additional term A_S.

You set a catalytic redox reaction in the “catalyst” section of the rate law. Set the “catalyst” to the name of the mineral serving as a catalyst,



reactants and kinetic reactions

Kinetic redox- 1

reaction

● rate constant mol/cm2 sec

○ preexp mol/cm2 sec

activation energy J/mol

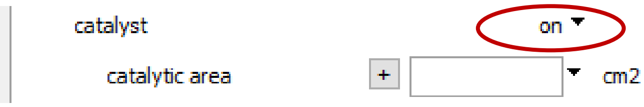
rate law built in ● forward ○ reverse

order 1 1.0 order 2 1.0

catalyst Goethite

catalytic area cm2/g

or simply to “on”.



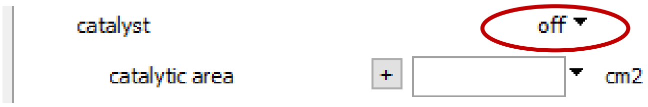
catalyst on

catalytic area cm2

Or, use the `catalyst` keyword in a `kinetic` command

```
kinetic redox-1 catalyst = Goethite
kinetic redox-1 catalyst = on
```

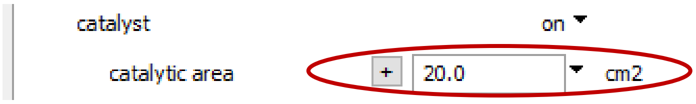
Setting a value of “off” disables the feature.



catalyst off

catalytic area cm2

Set the catalyzing surface area using the “catalytic area” field



catalyst on

catalytic area 20.0 cm2

or by typing

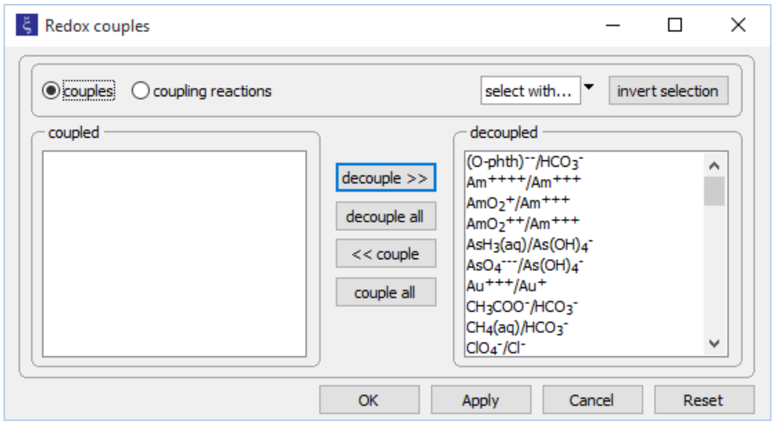
```
kinetic redox-1 surface = 20
```

If you have set a specific mineral as the catalyst, its surface area is set in cm^2 g^{-1}; the program calculates total surface area from this value and the mineral mass, as it varies over the simulation. If you have simply set “catalyst” to “on”, on the other hand, the total catalyzing surface area is set in cm^2.

Note that it is possible to set parallel pathways for the same redox reaction. Reaction “redox-1”, for example, could represent the reaction kinetics in the absence of a catalyst, and “redox-2”, the catalyzed reaction. The total reaction rate would be the sum of the rates for “redox-1” and “redox-2”.

The following procedure shows an example of how to incorporate redox catalysis into a **React** simulation. Here, the oxidation of Mn^{++} to manganite (MnOOH) by molecular oxygen is catalyzed on the surface of ferric hydroxide.

Double-click on file “Catalysis.rea” and go to the **Config → Redox couples…** dialog.



● couples ○ coupling reactions

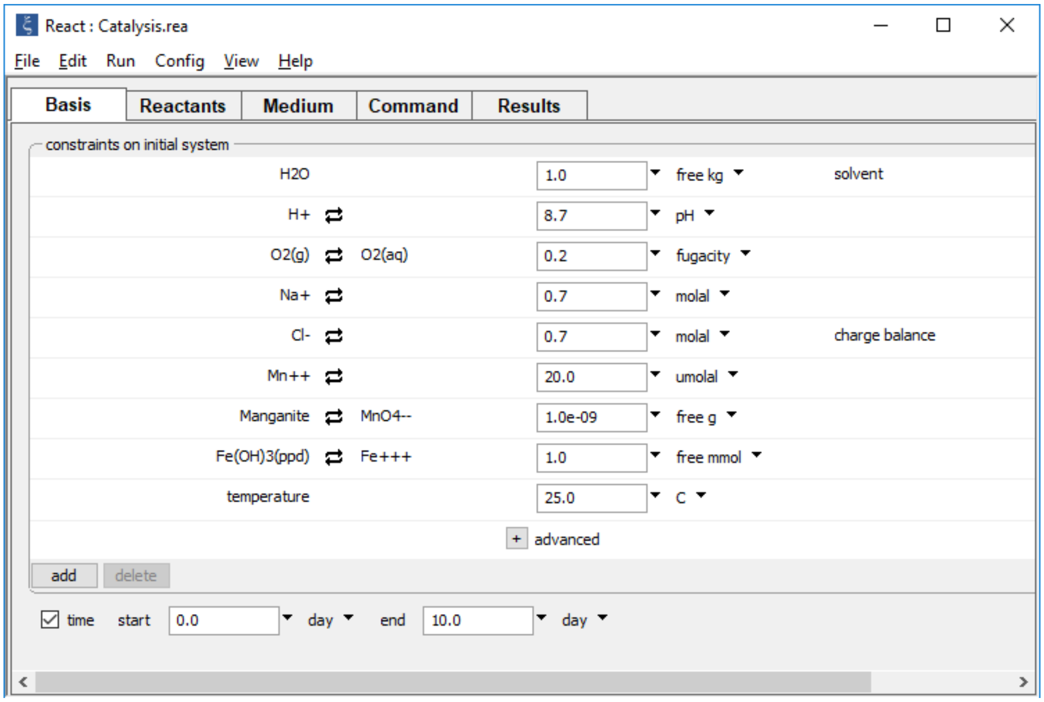
coupled:

decoupled:

- (O-phth)$^{--}$/HCO_3^-
- Am^{++++}/Am^{+++}
- AmO_2^+/Am^{+++}
- AmO_2^{++}/Am^{+++}
- $AsH_3(aq)$/$As(OH)_4^-$
- AsO_4^{---}/$As(OH)_4^-$
- Au^{+++}/Au^+
- CH_3COO^-/HCO_3^-
- $CH_4(aq)$/HCO_3^-
- ClO_4^-/Cl^-

Note how we’ve disabled all redox coupling reactions. Alternatively, individual reactions can be decoupled: those between ferrous and ferric iron, as well as the reactions between several oxidation states of manganese. Either way, iron and manganese in the simulation can change redox state only by kinetic reaction.

Look at the **Basis** pane.



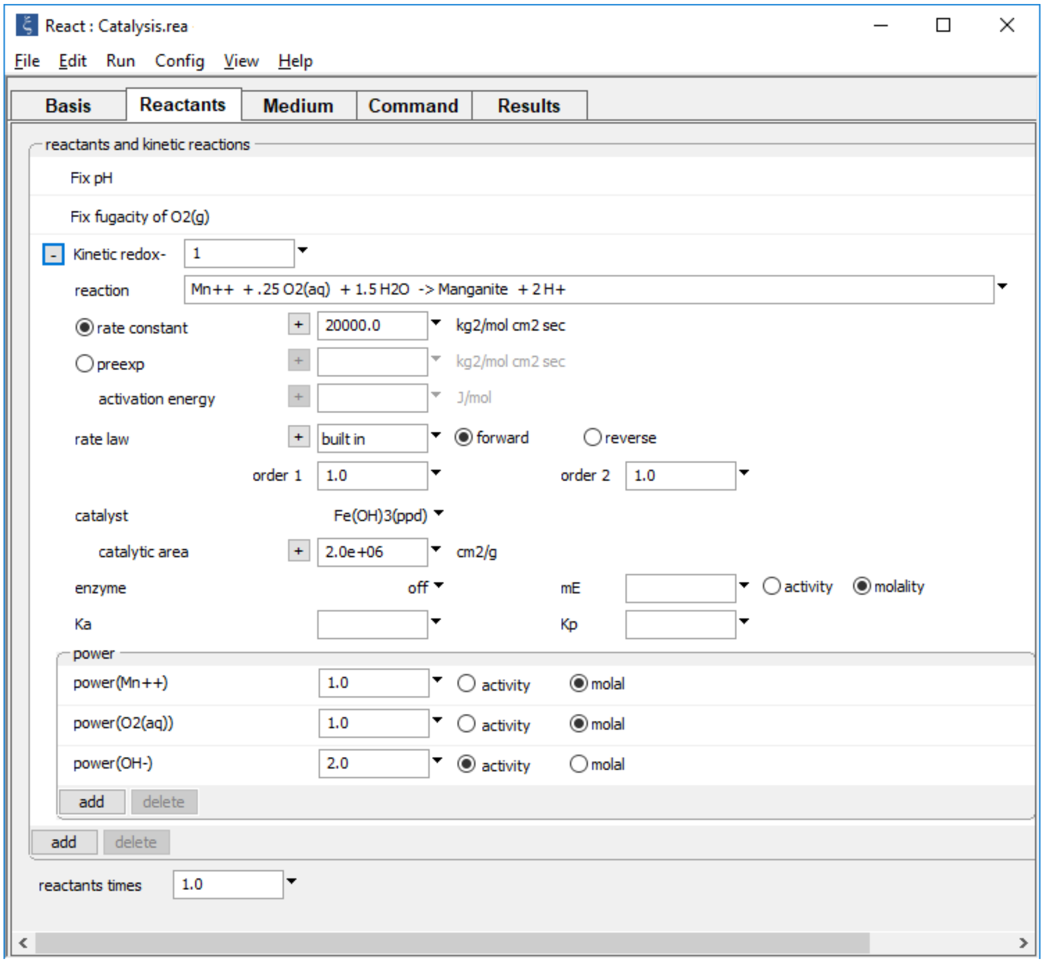
constraints on initial system

H2O		1.0	free kg	solvent
H+ ⇄		8.7	pH	
O2(g) ⇄	O2(aq)	0.2	fugacity	
Na+ ⇄		0.7	molal	
Cl- ⇄		0.7	molal	charge balance
Mn++ ⇄		20.0	umolal	
Manganite ⇄	MnO4--	1.0e-09	free g	
Fe(OH)3(ppd) ⇄	Fe+++	1.0	free mmol	
temperature		25.0	C	

☑ time start 0.0 day end 10.0 day

Consider a system containing reduced and oxidized manganese, as well as ferric hydroxide, which provides the catalyzing surface for manganese oxidation.

Move to the **Reactants** pane and expand the entry for reactant “redox-1”.



reactants and kinetic reactions

Fix pH

Fix fugacity of O2(g)

Kinetic redox- 1

reaction Mn++ + .25 O2(aq) + 1.5 H2O -> Manganite + 2 H+

● rate constant 20000.0 kg2/mol cm2 sec

○ preexp kg2/mol cm2 sec

activation energy J/mol

rate law built in ● forward ○ reverse

order 1 1.0 order 2 1.0

catalyst Fe(OH)3(ppd)

catalytic area 2.0e+06 cm2/g

enzyme off mE ○ activity ● molality

Ka Kp

power

power(Mn++) 1.0 ○ activity ● molal

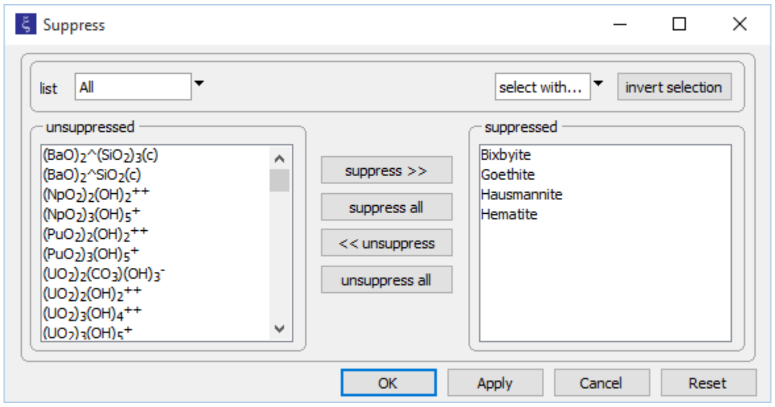
power(O2(aq)) 1.0 ○ activity ● molal

power(OH-) 2.0 ● activity ○ molal

reactants times 1.0

Fix the pH and O_2(g) fugacity to simulate a buffered experiment. Expanding the entry for reactant “redox-1” allows you to view the details of the kinetic rate law specified.

On the **Config → Suppress…** dialog



list: All

unsuppressed

$(BaO)_2$^$(SiO_2)_3(c)$
$(BaO)_2$^$SiO_2(c)$
$(NpO_2)_2(OH)_2^{++}$
$(NpO_2)_3(OH)_5^{+}$
$(PuO_2)_2(OH)_2^{++}$
$(PuO_2)_3(OH)_5^{+}$
$(UO_2)_2(CO_3)(OH)_3^{-}$
$(UO_2)_2(OH)_2^{++}$
$(UO_2)_3(OH)_4^{++}$
$(UO_2)_3(OH)_5^{+}$

suppressed

Bixbyite
Goethite
Hausmannite
Hematite

note that various iron and manganese minerals have been suppressed because they are stable, but unlikely to form over the time span of the simulation.

Select **Run → Go** to calculate the model.

4.6.2 Enzymes and biotransformations

React can trace the progress of enzymatically or biologically promoted redox reactions. In considering the effects of enzymes, the most commonly encountered rate law is the Michaelis-Menten equation. This equation may also be used in simple cases to model biotransformations (microbes are sometimes referred to as “bags of enzymes”).

To construct more sophisticated models of enzymatic and the microbe-mediated reactions, especially for cases in which microbial populations vary with time, see the section Microbial metabolism and growth, below.

The Michaelis-Menten equation represents the reaction of a substrate A with an enzyme E to form an activated complex EA, which decomposes irreversibly to give the reaction product P and return the enzyme:

$$\mathrm{A} + \mathrm{E} \underset{k_2}{\overset{k_1}{\rightleftarrows}} \mathrm{EA} \xrightarrow{k_3} \mathrm{P} + \mathrm{E} \tag{4.25}$$

The corresponding rate equation takes the well-known form,

$$r_{\vec{k}} = n_w \, k_+ \, m_E \, \frac{m_A}{m_A + K_A} \tag{4.26}$$

where k_+ is the intrinsic rate constant ($k_+ = k_3$, in s^{-1}), m_E is the molal concentration of the enzyme, m_A is the substrate concentration, and K_A is the half-saturation constant

$$K_A = \frac{k_2 + \ k_3}{k_1} \tag{4.27}$$

in molal units. The product $k_+ * m_E$ is the maximum reaction rate (molal s^{-1}), commonly denoted r_{max} or v_{max}.

The reaction can be generalized by including the back reaction of products with enzyme to form the activated complex:

$$\text{A} + \text{E} \underset{k_2}{\overset{k_1}{\rightleftarrows}} \text{EA} \underset{k_4}{\overset{k_3}{\rightleftarrows}} \text{P} + \text{E} \tag{4.28}$$

The rate law in this case takes the form,

$$r_{\vec{k}} \ = \ n_w \, k_+ \, m_E \, \frac{\dfrac{m_A}{K_A}}{1 + \dfrac{m_A}{K_A} + \dfrac{m_P}{K_P}} \left(1 - \frac{Q}{K}\right) \tag{4.29}$$

where K_P is the half-saturation constant

$$K_P \ = \frac{k_2 + \ k_3}{k_4} \tag{4.30}$$

for the reverse reaction, in molal, and m_P is the molal concentration of reaction product P. Note that when P is at small concentration, Q becomes small also, and the rate law reduces to the Michaelis-Menten equation.

React carries a generalized rate law to model enzymatic reactions and biotransformations. The rate law in linear and nonlinear forms is given as

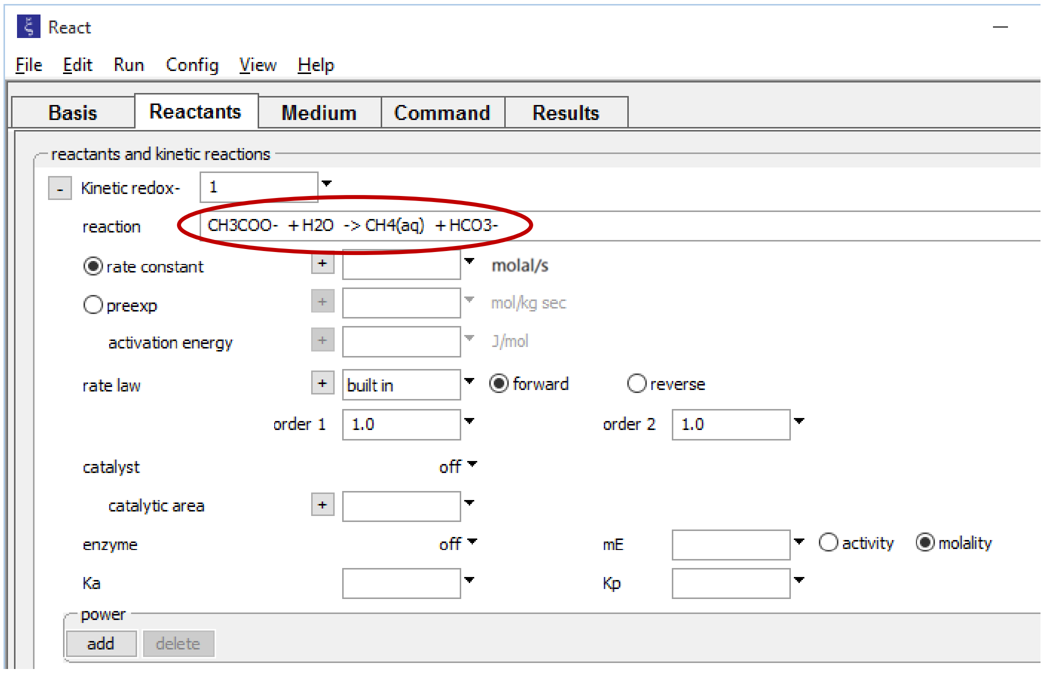
$$r_{\vec{k}} \ = \ n_w \, k_+ \prod_j \left(a_j | m_j\right)^{p_j} m_E \, \frac{\dfrac{m_A}{K_A}}{1 + \dfrac{m_A}{K_A} + \dfrac{m_P}{K_P}} \ \left(1 - \frac{Q}{K}\right) \tag{4.31}$$

and

$$r_{\vec{k}} = \mathrm{sgn}\left(1 - \frac{Q}{K}\right) n_w \, k_+ \prod_j (a_j|m_j)^{p_j} \, m_E \, \frac{\dfrac{m_A}{K_A}}{1 + \dfrac{m_A}{K_A} + \dfrac{m_P}{K_P}} \left|1 - \left(\frac{Q}{K}\right)^{\omega}\right|^{\Omega} \tag{4.32}$$

You set the reaction to be modeled using the `rxn` keyword, as before. It is important to note that the program takes the first species on the left of the reaction to be the limiting substrate, and the first species on the right to be the limiting reaction product (i.e., the limiting substrate for the reverse reaction). The program tracks the concentrations of these species over the course of the simulation and uses them as the values for m_A and m_P.

For example, in the following input


React
File Edit Run Config View Help
Basis Reactants Medium Command Results

reactants and kinetic reactions

Kinetic redox- 1

reaction CH3COO- + H2O -> CH4(aq) + HCO3-

● rate constant molal/s

○ preexp mol/kg sec

activation energy J/mol

rate law built in ● forward ○ reverse

order 1 1.0 order 2 1.0

catalyst off

catalytic area

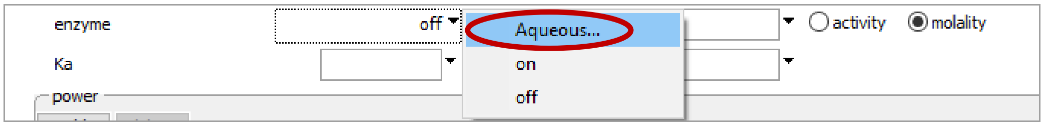
enzyme off mE ○ activity ● molality

Ka Kp

power

the acetate is the limiting substrate, and methane is the limiting product. Throughout the simulation, the program will use $m_{\mathrm{CH3COO-}}$ as m_A, and $m_{\mathrm{CH4(aq)}}$ as m_P. If we were to reverse the order of the species on the right of the reaction, however, the program would take $m_{\mathrm{HCO3-}}$ as m_P. The way you write the redox reaction, therefore, affects the modeling results.

Set the identity of the enzyme using the pulldown next to “enzyme” (or use the `mE` keyword). You can select **Aqueous…** from the pulldown,



enzyme off ▾ ○ activity ● molality

Aqueous…

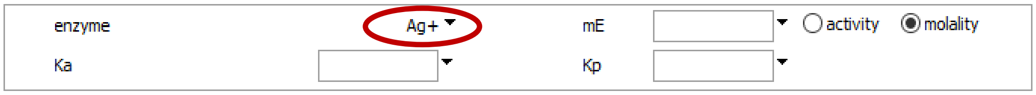
on

off

Ka

power

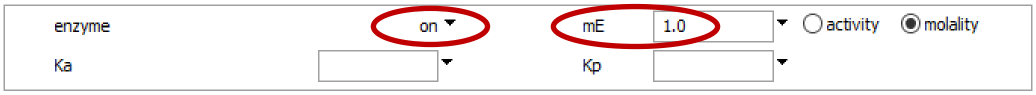
then choose an aqueous species in the simulation from the list. In this case, the program, over the course of the simulation, finds the molality of this species and uses it as m_E.



enzyme Ag+ ▾ mE ▾ ○ activity ● molality

Ka ▾ Kp ▾

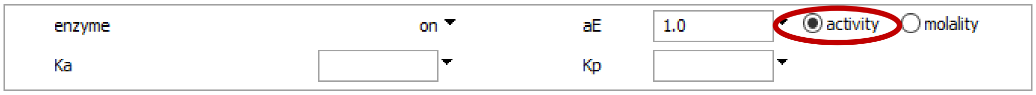
Alternatively, select “on” from the pulldown to set the enzyme molality to a constant value, which you enter in the “mE” field.



enzyme on ▾ mE 1.0 ▾ ○ activity ● molality

Ka ▾ Kp ▾

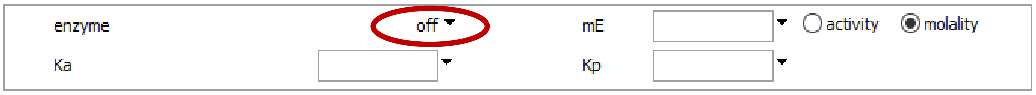
By selecting “activity” instead of “molal” (or using keyword `aE` in place of `mE`), you can specify the activities rather than molalities of the enzyme and the limiting reactant and product species to be used to evaluate the rate law. In this case, the program replaces variables m_E, m_A, and m_P in the rate law shown above with a_E, a_A, and a_P.



enzyme on ▾ aE 1.0 ▾ ● activity ○ molality

Ka ▾ Kp ▾

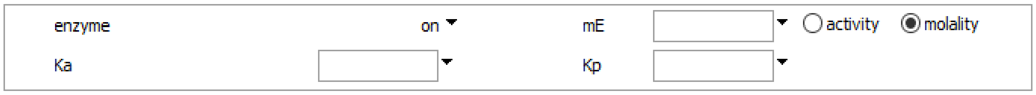
Setting “enzyme” to “off” disables the feature.



enzyme off ▾ mE ▾ ○ activity ● molality

Ka ▾ Kp ▾

You set the values for the half-saturation constants K_A and K_P in molal units using the “Ka” and “Kp” fields

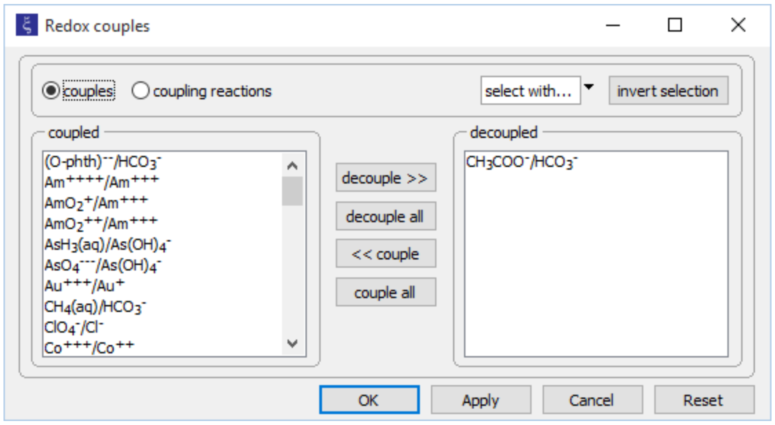


enzyme on ▾ mE ▾ ○ activity ● molality

Ka ▾ Kp ▾

(or the `KA` and `KP` keywords from the command pane), but you can omit a value for K_P, in which case the program will not carry the m_p/K_p term in the rate equation.

The following example models the biotransformation of acetate to bicarbonate by reaction with molecular oxygen, in an experiment open to atmospheric O_2 and CO_2. Double-click on file “Enzyme.rea” and go to the **Config → Redox Couples…** dialog.



coupled

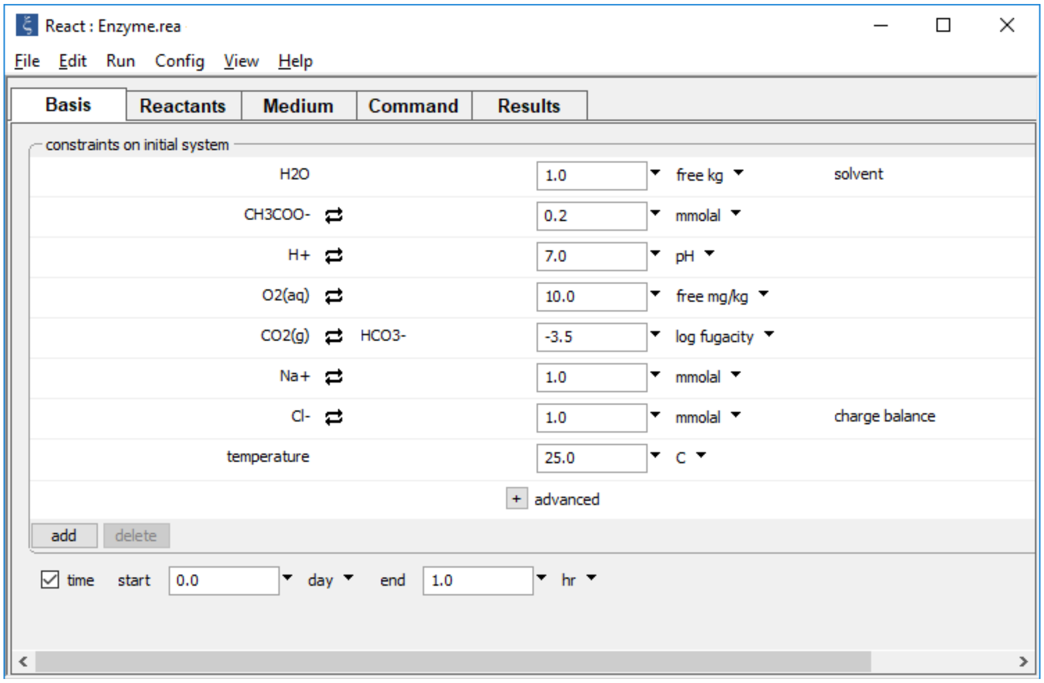
(O-phth)$^{--}$/HCO_3^-
Am^{++++}/Am^{+++}
AmO_2^+/Am^{+++}
AmO_2^{++}/Am^{+++}
$AsH_3(aq)$/$As(OH)_4^-$
AsO_4^{---}/$As(OH)_4^-$
Au^{+++}/Au^+
$CH_4(aq)$/HCO_3^-
ClO_4^-/Cl^-
Co^{+++}/Co^{++}

decoupled

CH_3COO^-/HCO_3^-

Note how the reaction between acetate and inorganic carbon has been disabled. As a result, carbon in the simulation can change redox state only by kinetic reaction.

Look at the **Basis** pane,


React : Enzyme.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
+ advanced
add
delete

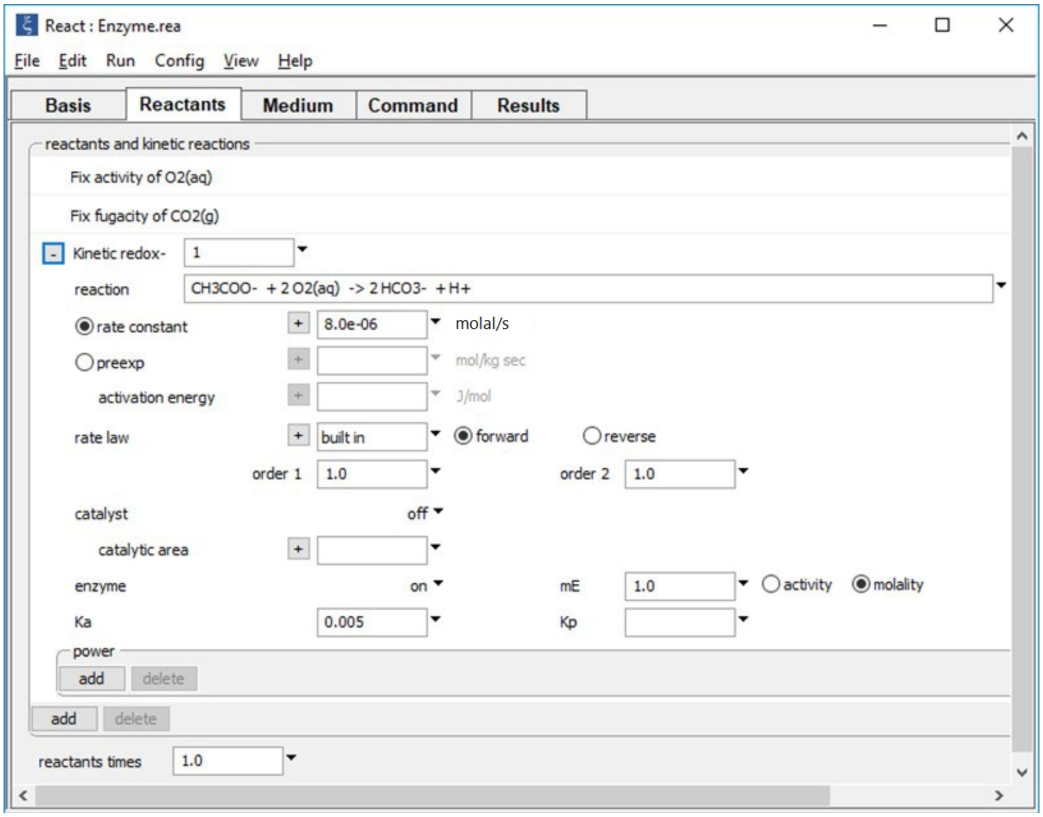
constraints on initial system

H2O		1.0	free kg	solvent
CH3COO-	⇄	0.2	mmolal	
H+	⇄	7.0	pH	
O2(aq)	⇄	10.0	free mg/kg	
CO2(g)	⇄ HCO3-	-3.5	log fugacity	
Na+	⇄	1.0	mmolal	
Cl-	⇄	1.0	mmolal	charge balance
temperature		25.0	C	

☑ time start 0.0 day end 1.0 hr

which describes a fluid with acetate in equilibrium with O_2 and CO_2 in the atmosphere. The simulation is set to span 1 hour.

Move to the **Reactants** pane and expand the entry for reactant “redox-1”.



reactants and kinetic reactions

Fix activity of O2(aq)

Fix fugacity of CO2(g)

Kinetic redox- 1

reaction CH3COO- + 2 O2(aq) -> 2 HCO3- + H+

● rate constant 8.0e-06 molal/s

○ preexp mol/kg sec

activation energy J/mol

rate law built in ● forward ○ reverse

order 1 1.0 order 2 1.0

catalyst off

catalytic area

enzyme on mE 1.0 ○ activity ● molality

Ka 0.005 Kp

power

reactants times 1.0

Fix the pH and O_2(g) fugacity to simulate an experiment open to the atmosphere. Taking a value of 8 × 10^{-6} molal s^{-1} for r_{max}, configure the simulation by setting “mE” to an arbitrary value of 1, and then k_+ to the value of r_{max}, since $r_{max} = k_+ * m_E$, as described earlier.

Alternatively, type

```
time end = 1 hr

decouple CH3COO-
CH3COO- = 0.2 mmolal

pH = 7
10 free mg/kg O2(aq); fix a O2(aq)
swap CO2(g) for HCO3-
log f CO2(g) = -3.5; fix f CO2(g)
Na+ = 1 mmolal
Cl- = 1 mmolal

kinetic redox-1 rxn = "CH3COO- + 2 O2(aq) -> 2 HCO3- + H+" \
           rate_con = 8e-6 mE = 1 KA = 5e-3
```

Select **Run → Go** to calculate the model.

4.7 Microbial metabolism and growth

React can trace the effect of microbial metabolism and growth in a geochemical system using a generalized kinetic rate law. As described below, the generalized rate law simplifies under specific conditions to each of the rate laws commonly used in microbial ecology.

In **React**, a microbe's metabolism is assumed to be based on the transfer of electrons from a donating half reaction

$$\sum_{\mathrm{D}} \nu_{\mathrm{D}}\mathrm{D} \leftrightarrows \sum_{\mathrm{D}^+} \nu_{\mathrm{D}^+}\mathrm{D}^+ + n\,\mathrm{e}^- \qquad (4.33)$$

to an accepting half reaction

$$\sum_{\mathrm{A}^+} \nu_{\mathrm{A}^+}\mathrm{A}^+ + n\,\mathrm{e}^- \leftrightarrows \sum_{\mathrm{A}} \nu_{\mathrm{A}}\mathrm{A} \qquad (4.34)$$

Here, D and D^+ represent species on the reduced and oxidized sides of the donating half reaction, A and A^+ represent the same for the accepting half reaction, and n is the number of electrons transferred. The microbe's overall metabolic reaction is the sum

$$\sum_{\mathrm{D}} \nu_{\mathrm{D}} \mathrm{D} + \sum_{\mathrm{A}^+} \nu_{\mathrm{A}^+} \mathrm{A}^+ \leftrightarrows \sum_{\mathrm{D}^+} \nu_{\mathrm{D}^+} \mathrm{D}^+ + \sum_{\mathrm{A}} \nu_{\mathrm{A}} \mathrm{A} \tag{4.35}$$

of the half reactions.

4.7.1 Metabolic rate

React calculates the metabolic rate (mol/s) for the progress of a microbial reaction according to the generalized rate law

$$r_{\vec{k}} = n_w \, k_+ [X] \times \frac{\prod_j m_j^{p_j}}{\left(K_D \prod_{\mathrm{D}^+} m_{\mathrm{D}^+}^{P_{\mathrm{D}^+}} + \prod_{\mathrm{D}} m_{\mathrm{D}}^{P_{\mathrm{D}}}\right)^{P_{KD}} \left(K_A \prod_{\mathrm{A}} m_{\mathrm{A}}^{P_{\mathrm{A}}} + \prod_{\mathrm{A}^+} m_{\mathrm{A}^+}^{P_{\mathrm{A}^+}}\right)^{P_{KA}}} \times TPF \tag{4.36}$$

where n_w and $[X]$ are, as before, water mass (kg) and biomass concentration (mg kg^{-1}), k_+ is the rate constant (mol mg^{-1} s^{-1}), m_j and p$_j$ are the molalities and the powers in the numerator term of the promoting and inhibiting species (which generally appear in the metabolic reaction, although this is not a requirement), P_{D} and P_{A}, are the powers in the denominator term for various species in the metabolic reaction, K_D and K_A are the half-saturation constants for the donating and accepting reactions (varying units, in mol, kg, and s), and P_{KD} and P_{KA} are the overall powers of the K_D and K_A terms.

Variable TPF is the thermodynamic potential factor, given by

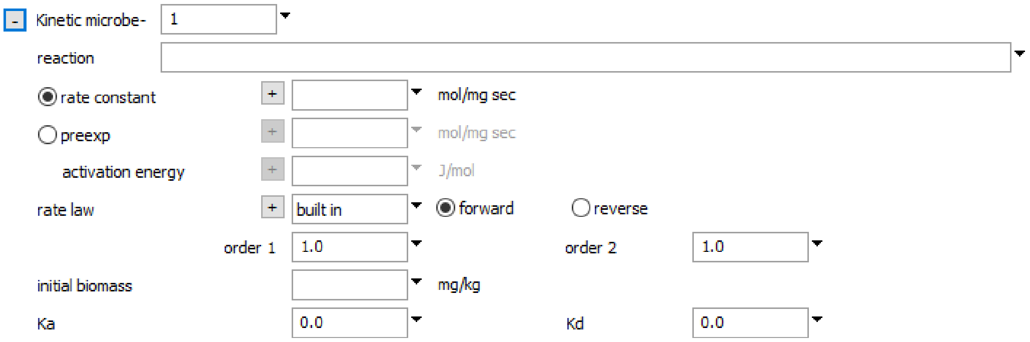
$$TPF = \left[1 - \left(\frac{Q}{K}\right)^{\omega} \exp\left(\frac{-\omega n_{\mathrm{ATP}} \Delta G_{\mathrm{ATP}}}{RT_K}\right)\right]^{\Omega} \tag{4.37}$$

where Q and K are the ion activity product and equilibrium constant for the metabolic reaction, and ω and Ω are the rate law orders, as before (see **Nonlinear rate laws**). Variable n_{ATP} is the number of moles of ATP produced per mole of progress of the metabolic reaction, ΔG_{ATP} is the free energy of ATP hydrolysis (the reaction by which adenosine tri-phosphate, ATP, decomposes to adenosine di-phosphate, ADP), R is the gas constant, and T_K is absolute temperature.

The *TPF* is similar to the (1 − *Q/K*) terms in the other rate laws carried by **React**, except for the exponential term, which accounts for the fact that, to live, the microbe must derive enough energy from its environment to synthesize ATP, the cellular energy store, from ADP.

Note that in contrast to the other types of kinetic reactions, **React** assumes that a microbe's metabolic reaction proceeds only in the forward direction. In other words, the program enforces $r_{\vec{k}} \geq 0$ for all cases.

To construct a microbial model, click add → **Kinetic** → **Microbial Reaction**, set the metabolic reaction in the "reaction" field (or with the `rxn` or `reaction` keyword), as before, and specify a rate constant (keyword `rate_con`) and the reaction orders ω and Ω (`order1` and `order2`). Set the half-saturation constants K_D and K_A for the donating and accepting reactions using the "Kd" and "Ka" fields (or with the `KD` and `KA` keywords); by default, these variables take a value of 0.



□ Kinetic microbe- 1

reaction

● rate constant + mol/mg sec

○ preexp + mol/mg sec

activation energy + J/mol

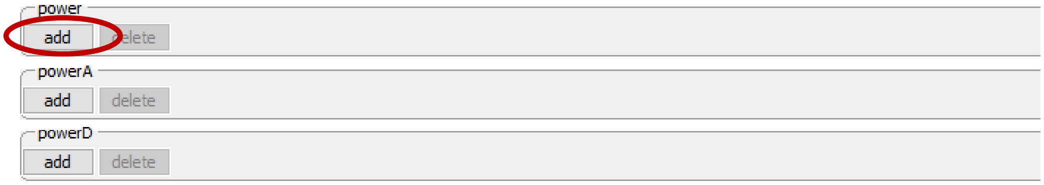
rate law + built in ● forward ○ reverse

order 1 1.0 order 2 1.0

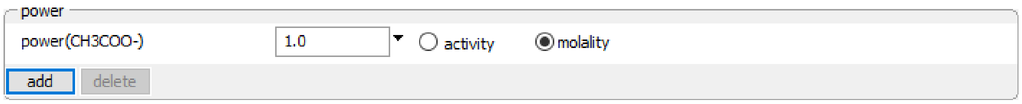
initial biomass mg/kg

Ka 0.0 Kd 0.0

Set the powers p_j, P_D, and so on, of species molalities in the numerator of the rate equation by clicking add under "power"



and selecting a species from the pulldown, or with the `mpower` (or `mpow`) keyword, as shown below.

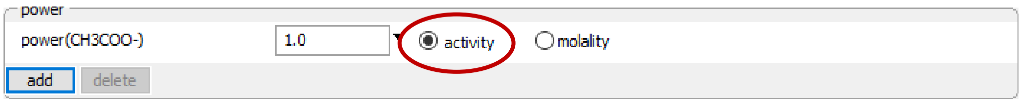


power

power(CH3COO-) 1.0 ○ activity ● molality

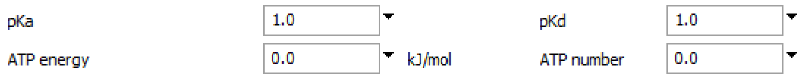
add delete

For the terms in the denominator of the rate law, set the power P_D and P_{D^+} of species from the donating reaction by clicking add under “powerD”, or using the `mpowerD` (or `mpowD`) keyword, and those from the accepting reaction (P_A and P_{A^+}) by clicking add under “powerA”, or with `mpowerA` (or `mpowA`). You can use the “activity” button or keywords `apower`, `apowerD`, or `apowerA` to set in the rate law a species’ activity in place of its molality, as shown below.



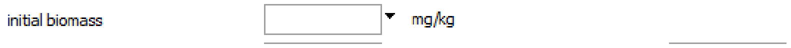
power
power(CH3COO-) 1.0 ● activity ○ molality

Use keywords `PKD` and `PKA` to set the exponents P_{KD} and P_{KA} for the grouped terms in the denominator. Keyword `ATP_number` sets the value of n_{ATP}, and `ATP_energy` sets ΔG_{ATP}, in kJ/mol.



pKa 1.0 pKd 1.0
ATP energy 0.0 kJ/mol ATP number 0.0

You set the initial biomass of the microbial population in mg/kg, using the `biomass` keyword.

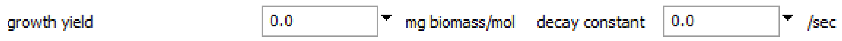


initial biomass mg/kg

Over the simulation, **React** determines the rate of change in the biomass $[X]$ from the calculated metabolic rate $r_{\vec{k}}$ according to the relation

$$\frac{d[X]}{dt} = \frac{Y\,r_{\vec{k}}}{n_w} - D[X] \qquad (4.38)$$

where Y is the growth yield, and D is the decay constant. Specify these parameters in mg mol^{-1} and s^{-1}, respectively, using the `growth_yield` and `decay_con` keywords.



growth yield 0.0 mg biomass/mol decay constant 0.0 /sec

4.7.2 Relationship to other rate laws

React's microbial rate law reduces under specific conditions to each of the variety of empirical and semi-empirical rate laws commonly applied in microbial ecology. These rate laws include the following equations:

- First-order $$\frac{dC}{dt} = k\, m_D$$
- Michaelis-Menten $$\frac{dC}{dt} = k \frac{m_D}{K_D + m_D}$$
- Zero-order $$\frac{dC}{dt} = k$$
- Logistic $$\frac{dC}{dt} = k\,[X]\, m_D$$
- Monod $$\frac{dC}{dt} = k\,[X] \frac{m_D}{K_D + m_D}$$
- Dual-Monod $$\frac{dC}{dt} = k\,[X] \frac{m_D}{K_D + m_D} \frac{m_{A^+}}{K_A + m_{A_+}}$$
- Logarithmic $$\frac{dC}{dt} = k\,[X]$$

Here, C is the concentration of a product species (or the negative concentration of a substrate of electron accepting species), k is a rate constant in appropriate units, $[X]$ is biomass concentration, m_D and m_{A_+} are activities of the limiting reactants in the donating and accepting half reactions, and K_D and K_A are the corresponding half-saturation constants. (The first-order, Michaelis-Menten, logistic, and Monod equations are commonly written in terms of a limiting species from the donating reaction, as shown, but may also be cast in terms of a species from the accepting reaction. In this case, m_{A_+} replaces m_D, and K_A takes the place of K_D in the equations above, as well as in Table 4.2).

To invoke one of these laws in **React**, set the appropriate parameters in your input, as shown in Table 4.2. In each case, set an appropriate value for the rate constant (keyword `rate_con`). You can set the rate law order Ω ("order2") to 0,

order 1 1.0

order 2 0

which negates the thermodynamic potential function and hence casts the laws in the exact form shown above. Alternatively, you can leave Ω at its default value of 1, which will prevent the metabolic reaction in the simulation from proceeding beyond the point of equilibrium, which, of course, is a chemical impossibility.

Table 4.2 Settings for casting microbial rate law in terms of the various empirical equations. Symbol ✓ indicates an appropriate value should be set; x, no value is set. Symbols D and A^+ represent names of limiting reactants in donating and accepting half reactions.

	mpow	mpowD	mpow	mpowA				growth_	decay_
rate law	(D)	(D)	(A^+)	(A^+)	KD	KA	biomass	yield	con
First-order	1	x	x	x	x	x	1	0	0
Michaelis-Menten	1	1	x	x	✓	x	1	0	0
Zero-order	x	x	x	x	x	x	1	0	0
Logistic	1	x	x	x	x	x	✓	✓	✓
Monod	1	1	x	x	✓	x	✓	✓	✓
Dual Monod	1	1	1	1	✓	✓	✓	✓	✓
Logarithmic	x	x	x	x	x	x	✓	✓	✓

4.7.3 Example calculation

As an example, we consider two microbial strains growing by oxidizing acetate (Ac)

$$\mathrm{CH_3COO^-} + 2\,\mathrm{O_2(aq)} \rightarrow \mathrm{HCO_3^-} + \mathrm{H^+} \tag{4.39}$$

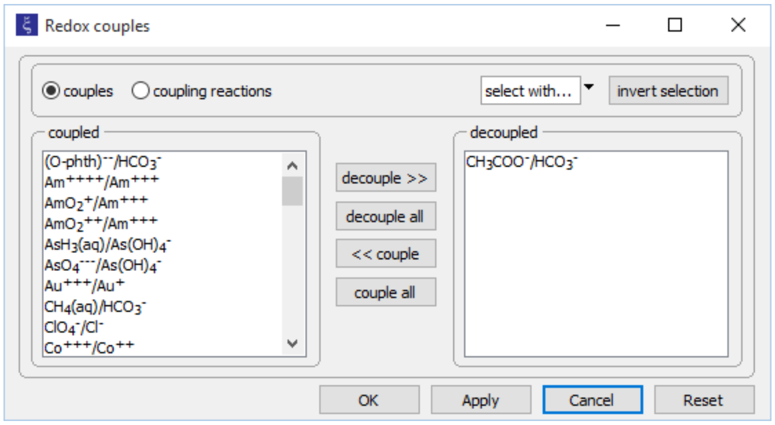
in a chemostat experiment open to the atmosphere. The microbes' metabolic rate is defined by

$$r_{\vec{k}} = k_+ \, [X] \, \frac{m_{\mathrm{Ac}}}{K_D + m_{\mathrm{Ac}}} \times \mathrm{TPF} \tag{4.40}$$

where the thermodynamic potential factor is

$$TPF = 1 - \left(\frac{Q}{K}\right)^{{}^1\!/_4} \exp\left(-\frac{{}^1\!/_4\, n_{\mathrm{ATP}}\, \Delta G_{\mathrm{ATP}}}{RT_K}\right) \tag{4.41}$$

One of the strains is faster at metabolizing than the other (it has a higher k_+), whereas the second is more efficient (lower k_D). Double-click on file “Microbes.rea” and go to the **Config → Redox Couples…** dialog.



● couples ○ coupling reactions

coupled

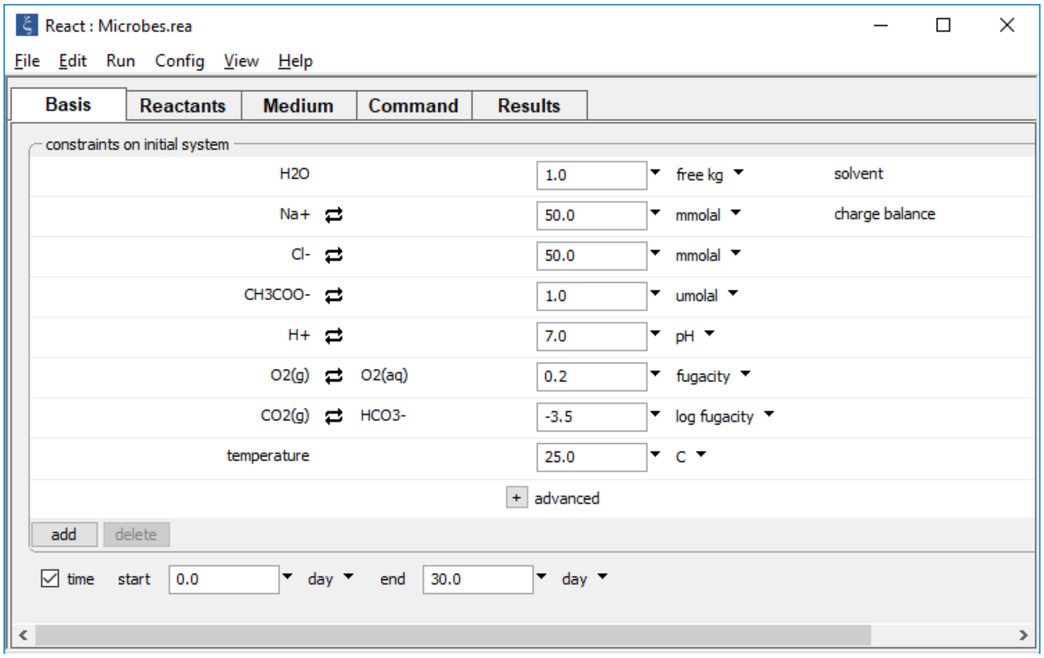
$(O\text{-phth})^{--}/HCO_3^-$
Am^{++++}/Am^{+++}
AmO_2^+/Am^{+++}
AmO_2^{++}/Am^{+++}
$AsH_3(aq)/As(OH)_4^-$
$AsO_4^{---}/As(OH)_4^-$
Au^{+++}/Au^+
$CH_4(aq)/HCO_3^-$
ClO_4^-/Cl^-
Co^{+++}/Co^{++}

decoupled

CH_3COO^-/HCO_3^-

Note how the reaction between acetate and inorganic carbon is disabled. As a result, carbon in the simulation can change redox state only by kinetic reaction.

Look at the **Basis** pane,


React : Microbes.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
+ advanced
add delete

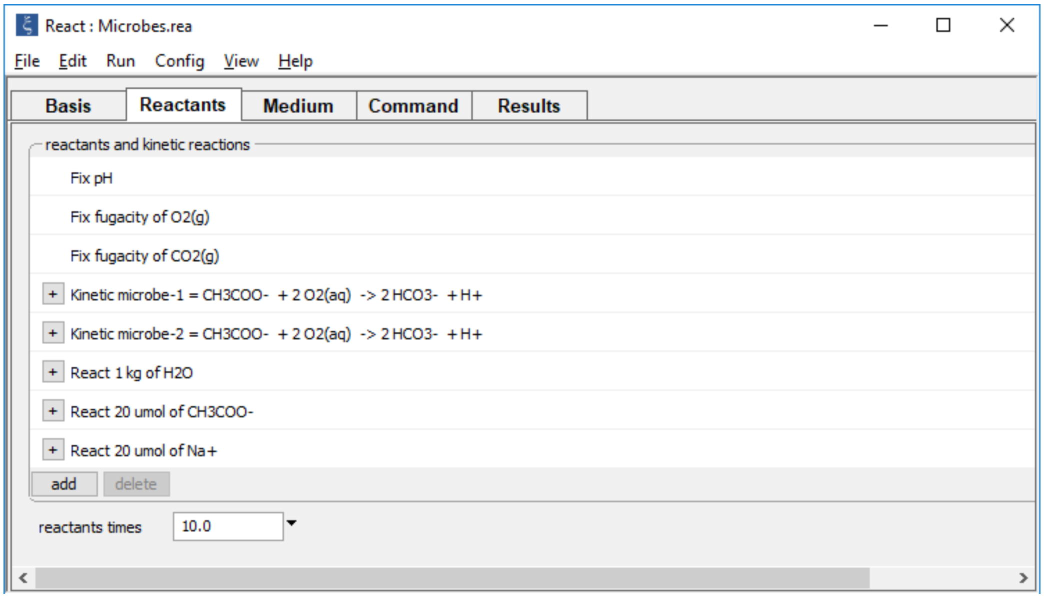
constraints on initial system

H2O			1.0	free kg	solvent
Na+	⇄		50.0	mmolal	charge balance
Cl-	⇄		50.0	mmolal	
CH3COO-	⇄		1.0	umolal	
H+	⇄		7.0	pH	
O2(g)	⇄	O2(aq)	0.2	fugacity	
CO2(g)	⇄	HCO3-	-3.5	log fugacity	
temperature			25.0	C	

☑ time start 0.0 day end 30.0 day

which describes the initial fluid composition in a chemostat experiment open to the atmosphere.

Move to the **Reactants** pane.


React : Microbes.rea
File Edit Run Config View Help
Basis Reactants Medium Command Results
add delete

reactants and kinetic reactions

Fix pH

Fix fugacity of O2(g)

Fix fugacity of CO2(g)

+ Kinetic microbe-1 = CH3COO- + 2 O2(aq) -> 2 HCO3- + H+

+ Kinetic microbe-2 = CH3COO- + 2 O2(aq) -> 2 HCO3- + H+

+ React 1 kg of H2O

+ React 20 umol of CH3COO-

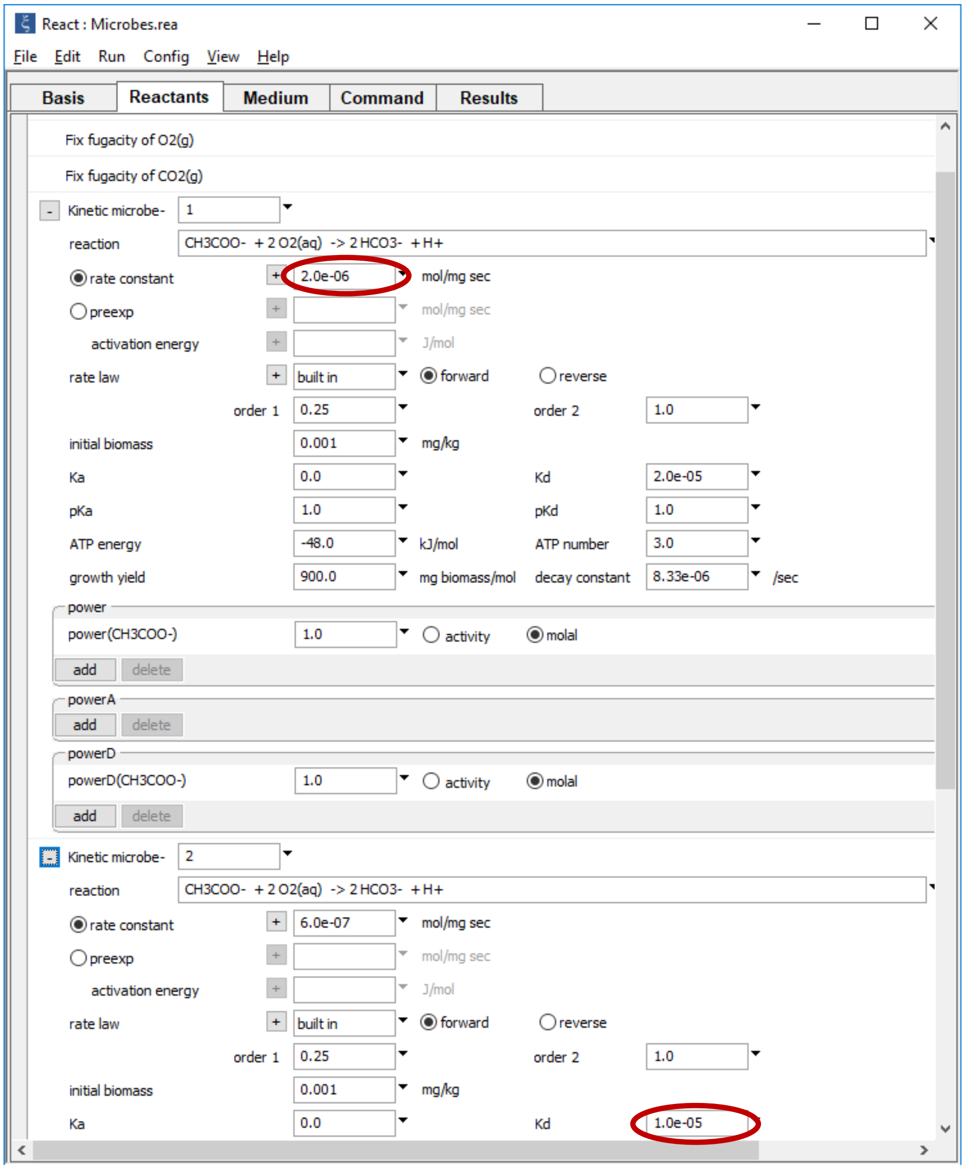
+ React 20 umol of Na+

reactants times 10.0

Fix the pH and gas fugacities to simulate an experiment open to the atmosphere.

The simple reactants and the value set for “reactants times”, combined with the “flush” option enabled from the **Config → Stepping…** dialog, cause the fluid in the chemostat to be replaced by a 20 µmolal Na-CH_3COO solution 10 times.

Expand the entries for reactants “microbe-1” and “microbe-2” to view the details of the kinetic rate laws specified.



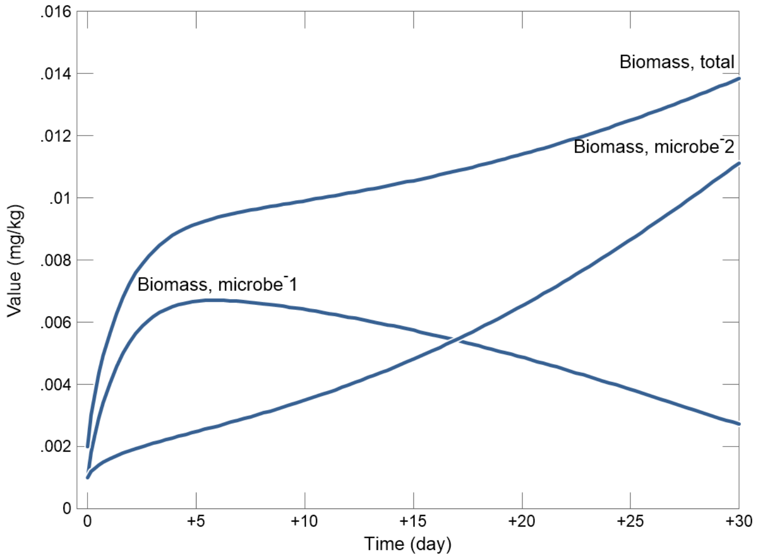
Fix fugacity of O2(g)
Fix fugacity of CO2(g)
Kinetic microbe- 1
reaction CH3COO- + 2 O2(aq) -> 2 HCO3- + H+
● rate constant 2.0e-06 mol/mg sec
○ preexp mol/mg sec
activation energy J/mol
rate law built in ● forward ○ reverse
order 1 0.25 order 2 1.0
initial biomass 0.001 mg/kg
Ka 0.0 Kd 2.0e-05
pKa 1.0 pKd 1.0
ATP energy -48.0 kJ/mol ATP number 3.0
growth yield 900.0 mg biomass/mol decay constant 8.33e-06 /sec
power
power(CH3COO-) 1.0 ○ activity ● molal
add delete
powerA
add delete
powerD
powerD(CH3COO-) 1.0 ○ activity ● molal
add delete
Kinetic microbe- 2
reaction CH3COO- + 2 O2(aq) -> 2 HCO3- + H+
● rate constant 6.0e-07 mol/mg sec
○ preexp mol/mg sec
activation energy J/mol
rate law built in ● forward ○ reverse
order 1 0.25 order 2 1.0
initial biomass 0.001 mg/kg
Ka 0.0 Kd 1.0e-05

The first strain is faster at metabolizing than the second (it has a higher k_+), whereas the second is more efficient (lower K_D).

The following commands configure the same simulation.

```
decouple CH3COO-

time end = 30 day

Na+ = 50 mmolal
Cl- = 50 mmolal
CH3COO- = 1 umolal
balance on Na+
pH = 7

fix pH

swap O2(g) for O2(aq)
f O2(g) = 0.2
fix O2(g)

swap CO2(g) for HCO3-
log f CO2(g) = -3.5
fix f CO2(g)

kinetic microbe-1 \
  rxn = "CH3COO- + 2 O2(aq) -> 2 HCO3- + H+", \
  rate_con = 2e-6, KD = 2e-5, \
  mpower(CH3COO-) = 1, mpowerD(CH3COO-) = 1, \
  order1 = 0.25, ATP_energy = -48, ATP_number = 3, \
  biomass = .001, growth_yield = 900, decay_con = 8.33e-6

kinetic microbe-2 \
  rxn = "CH3COO- + 2 O2(aq) -> 2 HCO3- + H+", \
  rate_con = .6e-6, KD = 1e-5, \
  mpower(CH3COO-) = 1, mpowerD(CH3COO-) = 1, \
  order1 = 0.25, ATP_energy = -48, ATP_number = 3, \
  biomass = .001, growth_yield = 450, decay_con = 1.67e-6

react 1 kg of H2O
react 20 umol of CH3COO
react 20 umol of Na+
reactants times 10

flush
```

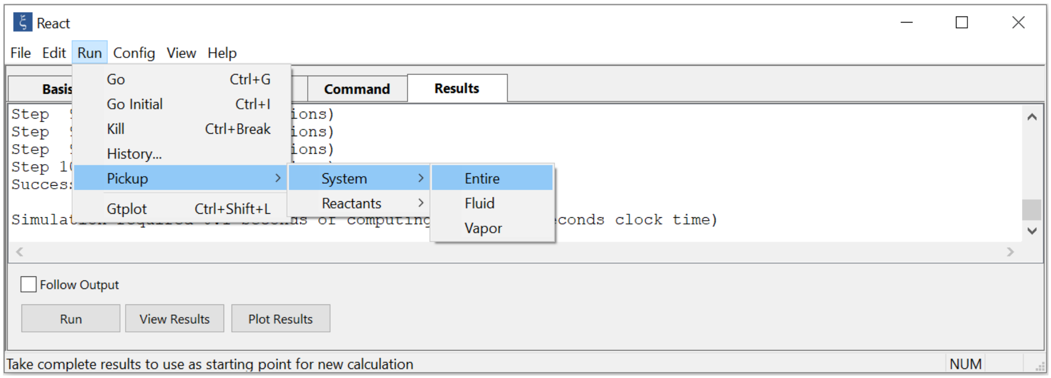
Select **Run → Go** to calculate the model. In the calculation results, the rapidly metabolizing strain grows fastest initially, but with time is replaced by the more efficient strain.



4.8 Chaining together kinetic paths

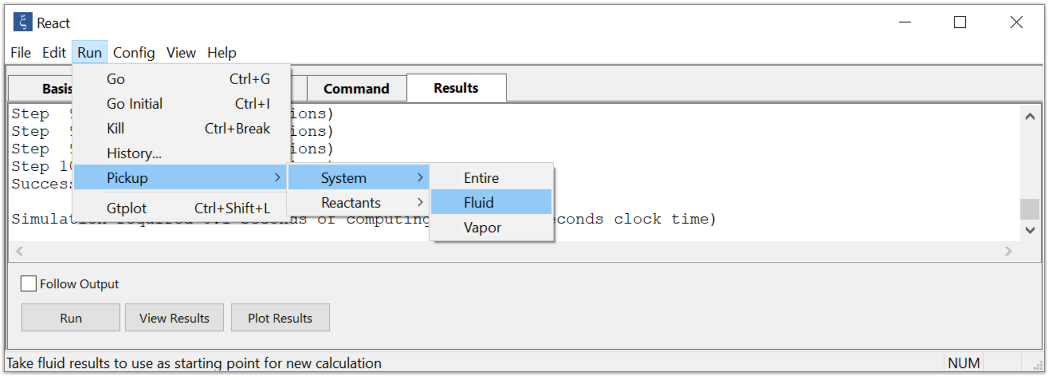
You can use the `pickup` command (see Picking up the results of a run in the previous chapter) to chain together kinetic reaction paths. When you pick up the results of a calculation, **React** retains all or some of the kinetic reactions being traced, at their endpoint states. You can then initiate a subsequent simulation that traces continued progress of the kinetic reactions.

In the simplest case, when you pick up the entire system using **Run → Pickup → System → Entire** (or the command `pickup`),



```
Step                           ions)
Step                           ions)
Step                           ions)
Step 1(
Succes:

Simulat                        as of computing          conds clock time)
```

Take complete results to use as starting point for new calculation the program retains whatever kinetic reactions are set. Using **Run → Pickup → System → Fluid** (or `pickup fluid`),



```
Step           ions)
Step           ions)
Step           ions)
Step 1
Succes

Simulat        s of computing            conds clock time)
```

Take fluid results to use as starting point for new calculation

causes **React** to retain the kinetic reactions occurring within the aqueous phase—the kinetic redox and aqueous complexation reactions—for the new reaction path; kinetic reactions involving minerals, end members, surfaces, a gas phase, or microbes are discarded.

Picking up the entire endpoint system, or just the endpoint minerals, as reactants (i.e., `pickup reactants` or `pickup reactants = minerals`) causes the program to retain the kinetic reactions involving precipitation and dissolution of minerals and end members. The `pickup reactants = minerals` command, then, lets you model experiments in water-rock interactions in which minerals react kinetically with a fluid that is periodically refreshed.

To see how kinetic reactions can be chained together, start by simulating the kinetic dissolution of quartz in dilute water at 100°C:

```
time end = 10 hours
T = 100 C
SiO2(aq) = 1 umol/kg
react 5000 g Quartz
kinetic Quartz rate_con = 2.e-15 surface = 1000
go
```

We wish to figure the effect, after reacting the quartz for 10 hours, of adding a reaction promoter to the solution. To do so, pick up the calculation results, increase the rate constant by a factor of 5 to reflect the promoter, and run a second path.

```
pickup
kinetic Quartz rate_con = 10.e-15
go
```

Over the course of the second reaction path, which begins at t = 10 hours and continues to t = 20 hours, quartz dissolves more rapidly than in the first, reflecting the increased value of the rate constant.

Custom Rate Laws

React's built-in rate laws for mineral and end member dissolution and precipitation, aqueous and surface complex dissociation and association, redox, microbe-mediated reactions, and gas transfer can be applied to the great majority of kinetic problems in geochemistry. For cases in which you wish to apply a rate law of a differing form, however, you can easily define one of your own.

You can define a rate law in any of three ways:

1. Specify its form as a character string. For example, you might set `rate_con * surface * (1 - Q/K)` as a rate law.
2. Prepare a file that contains a script of commands for calculating the reaction rate. Write the script in a simplified version of the MS Visual BASIC language, as described below.
3. Write and compile your own function for evaluating the rate law (you will probably want to use C++), and link it into a library. **React**, as it runs, opens the library and executes the function directly whenever it needs to evaluate the rate law.

Each method is described below.

5.1 Setting the rate law form directly

The simplest method for defining a rate law is to specify its form as a character string. **React** contains an interpreter (prepared with the "Bison" compiler; see Further Reading) that converts the character string into computer instructions that **React** executes.

You set the form of the rate law with the `rate_law` parameter of the `kinetic` command. For example, you might enter the command

```
kinetic Quartz rate_law = "rate_con * surface * (1 - Q/K)", \
   rate_con = 1e-18, surface = 1000
```

This command will cause the reaction rate for quartz to be calculated as the product of the rate constant, the mineral's surface area, and its saturation.

Table 5.1 lists the parameters, such as `rate_con`, that you can use to set rate laws in this manner. There are also a number of "helper functions" that you can invoke; these are listed in Table 5.2. For example, in your script you can use the helper function `activity("H+")` to denote activity of the hydrogen ion.

In addition to these parameters, a number of parameters identifying the current nodal block and describing its properties are available in **X1t** and **X2t**. For a list of these parameters, see the appendix on **Heterogeneity** in the ***GWB Reactive Transport Modeling Guide***.

If you enclose helper function arguments in double quotes ("), you need to delimit the rate law itself with single quotes ('). For example,

```
kinetic redox-1 rate_law = 'Wmass * molality("Na") * rate_con'
```

Alternatively, use single quotes to set off helper function arguments and double quotes for the rate law string.

5.2 Rate law scripts

You can define more sophisticated rate laws by preparing a script in a simplified form of the BASIC language, as described in Table 5.3. Save the script in a file, the name of which must end with the extension .bas. You then specify the file's name in the kinetic command with the `rate_law` parameter. For example, the command

```
kinetic Quartz rate_law = My_script.bas rate_constant = 10^-16
```

tells **React** that the rate law for the reaction of Quartz is to be found in the specified file. You also use the kinetic command in the usual way to set any parameters used by your script, such as the rate constant or specific surface area, as shown above. **React** looks for your .bas file first in the current working directory, and then in the directory where the GWB is installed (for example, C:\Program Files\GWB). You can, of course, specify a full path name to the file, to avoid ambiguity.

Alternatively, you can set your rate law script directly within your input file. For example, the input

```
kinetic Quartz rate_law = {
   rate = Wmass * rate_con / activity("H+")
   RETURN rate
}
```

sets the rate law script directly, without need to refer to an external file.

Your script can access the parameters in Table 5.1, as well as the "helper functions" in Table 5.2. Parameters in rate law scripts are copies of the values carried internally by **React**; with four exceptions, these values cannot be changed from your script. The exceptions are the biomass, the two kinetic factors in the microbial model, and the thermodynamic potential factor (parameters "biomass", "FD", "FA", and "TPF"), which you can set with the special function `setgwbvar`. For example, the entry

```
setgwbvar("biomass", newbiom)
```

sets the current value carried by **React** for "biomass" to the value of "newbiom", previously calculated in the script.

An example of a rate law script (the example scripts in this section are found in files within the "src" subdirectory, where the GWB is installed) is

```
IF QoverK < 1 THEN 20 ELSE 40
20: rate = rate_con * surface * (1 - Q/K)
GOTO 60
40: rate = 0.0
60: RETURN rate
```

In this example, the dissolution rate of a mineral is calculated from its rate constant, surface area, and saturation. The mineral's rate of precipitation, should it appear supersaturated, however, is 0.

As a second example, we account in the rate law script for the effects of promoting and inhibiting species on a mineral's reaction rate.

```
cat = 1.0
i = 1
begin_loop: IF i > ncatal THEN GOTO end_loop
   cat = cat * acatal(i)^pcatal(i)
   i = i + 1
GOTO begin_loop
end_loop: rate = cat * rate_con * surface * (1.0 - QoverK)
RETURN rate
```

Similarly, we could write a script to return the rate of a redox reaction in the form

```
cat = 1.0
i = 1
begin_loop: IF i > ncatal THEN GOTO end_loop
   cat = cat * acatal(i)^pcatal(i)
   i = i + 1
GOTO begin_loop
end_loop: rate = Wmass * cat * rate_con * (1.0 - QoverK)
RETURN rate
```

As a final example, we construct a script to return the rate of microbial metabolism, taking Ω to be unity. In this case, we need to calculate how the microbial biomass and the thermodynamic potential factor vary, and update each in **React**'s memory space using the `setgwbvar` function, as described above.

```
aff = QoverK * exp(-ATP_number * ATP_energy / (8.3143 * TK))
aff = aff^order1
IF (aff > 1.0) THEN aff=1.0

prod = 1.0
i=1
begin_loop1: IF i>ncatal THEN GOTO end_loop1
   prod = prod * acatal(i)^pcatal(i)
   i=i+1
GOTO begin_loop1
end_loop1:

prodKd1 = KD
prodKd2 = 1.0
i=1
begin_loop2: IF i>nspecD THEN GOTO end_loop2
   IF signD(i) > 0 THEN prodKd1 = prodKd1*actD(i)^powerD(i)
   IF signD(i) < 0 THEN prodKd2 = prodKd2*actD(i)^powerD(i)
   i=i+1
GOTO begin_loop2
end_loop2:
```

(cont'd)

```
prodKa1 = KA
prodKa2 = 1.0
i=1
begin_loop3: IF i>nspecA THEN GOTO end_loop3
   IF signA(i) > 0 THEN prodKa1 = prodKa1*actA(i)^powerA(i)
   IF signA(i) < 0 THEN prodKa2 = prodKa2*actA(i)^powerA(i)
   i=i+1
GOTO begin_loop3
end_loop3:

tpf = 1.0 - aff
fd = prodKd2 / (prodKd1 + prodKd2)^PKD
fa = prodKa2 / (prodKa1 + prodKa2)^PKA
setgwbvar("TPF", tpf)
setgwbvar("FD", fd)
setgwbvar("FA", fa)
denom = (prodKd1 + prodKd2)^PKD * (prodKa1 + prodKa2)^PKA
rprime = rate_con * (prod/denom) * tpf

IF (Deltat <= 0.0) THEN GOTO end_block
   rprime0 = dndt0/(biomass0 * Wmass)
   rpave = (1.0-Theta)*rprime0 + Theta*rprime
   biom = biomass0*exp((growth_yield*rpave-decay_con)*Deltat)
   IF biom < 0.0 THEN biom = 0.0
   setgwbvar("biomass", biom)
end_block:

return rprime * biomass * Wmass
```

Note the technique used in this script to update the biomass. The time rate of change in biomass concentration is given as

$$\frac{d[X]}{dt} = \frac{Y r_{\vec{k}}}{n_w} - D[X] \tag{5.1}$$

as described in the previous section of this guide. We can re-express this relation in integral form to give biomass concentration $[X]$ at the end of a reaction step

$$[X] = [X]_0 \, e^{(Y r' - D)\Delta t} \tag{5.2}$$

from the concentration $[X]_0$ at the beginning of the step, the reaction rate r' per unit biomass concentration per kg water (where $r_{\vec{k}} = r' n_w [X]$), and the step length Δt.

5.3 Compiled rate laws

If you have a little interest in programming, you may wish to write your own rate law function, most likely in C++. **React**, as it runs, can link to your function and call it directly. This method provides the most efficient and powerful way to define a custom rate law.

There are three steps in the procedure:

1. Write a function that returns the reaction rate (and optionally its derivatives) for a kinetic reactant, using the “gwb_context.h” and “ratelaw_param.h” header files supplied with the GWB software release. Copies of these files are found in the “src” subdirectory, where the GWB is installed.
2. Compile the function and link it into a dynamic link library (or DLL). The DLL must end in the extension .dll. You may link, within a single library, one or several rate law functions.
3. In the **React** input, specify the names of the library and function, separated by a colon (`:`). For example, setting

```
kinetic Quartz rate_law = my_lib.dll:my_ratelaw
```

Each rate law function takes three parameters:

1. An integer value *i*. When *i* is –1, the function should calculate reaction rate, returning a value in mol/sec. When *i* is the index of a basis entry (i.e., it is in the range 0 to the number of basis entries, less 1), the function should calculate the derivative of reaction rate with respect to the indicated basis entry. Alternatively, the function may return the special value “ANULL”, in which case **React** will calculate the derivative itself, using finite differences.
2. Reference to a “Reactant” data structure that describes the kinetic reaction in question. This reference should be named “r”. The form of this data structure is defined in a header file “gwb_context.h”, which is distributed with the GWB.
3. Reference to a “GWBcontext” data structure that contains information about current conditions (temperature, ionic strength, and so on) in the **React** simulation. This reference is named “c”. Again, the form of this data structure is defined in “gwb_context.h”.

The parameters in Table 5.1 are available to your function. These parameters are defined in a second header file “ratelaw_param.h”, which should be included at the top of the function. You can also use as parameters any of the entries in the “GWBcontext” data structure, as set out in “gwb_context.h”. In addition, the “helper functions” listed in Table 5.2 are available. To debug functions, use the `printf` function (but not `cout`) to display data.

As an example of a rate law function, consider a mineral that dissolves according to its rate constant, surface area, and saturation, but does not precipitate. Such a function, here named “example1”, can be written

```
#define EXPORT extern "C" __declspec(dllexport)
#include <windows.h>
#include <stdio.h>
#include <math.h>
#include "gwb_context.h"

EXPORT double example1(int i, Reactant &r, GWBcontext &c)
{
   #include "ratelaw_param.h"

   if (QoverK > 1.0) // Mineral does not precipitate.
      return 0.0;
   else if (-1 == i) // Dissolution rate.
      return rate_con * surface * (1.0 - QoverK);
   else // Let React determine derivatives.
      return ANULL;
}
```

Note the need to export the function from the DLL. We specify a C rather than C++ function, to avoid “name globbing” conflicts among compilers. Also note that by returning “ANULL” when **React** requests a derivative, we tell **React** to compute these values itself, using finite differences.

To compile the DLL, open your compiler’s 64-bit command line environment corresponding if you have 64-bit GWB version installed on your computer, or the 32-bit environment if you are running 32-bit GWB. The commands

```
cl /MD /O2 /c example1.cpp
link /dll example1.obj react.lib /out:mylib.dll
```

compile and link this function using the Microsoft C++ compiler, and

```
icl /MD /O2 /c example1.cpp
xilink /dll example1.obj react.lib /out:mylib.dll
```

do the same for the Intel C++ compiler. Here, file “example1.cpp” contains the C++ source, and “react.lib” is a short dataset supplied with the GWB; parallel files “phase2.lib”, “x1t.lib”, “x2t.lib”, and “chemplugin.lib” are supplied for use with programs **Phase2**, **X1t**, **X2t**, and **ChemPlugin**.

Note you should change to a working directory in your file space before compiling and linking; the process will not work correctly in a system directory, such as `\Program Files`.

It is important to note that the internal data structures in the GWB will likely change as the software continues to evolve. You should preserve the source code for your rate law functions, so that you can recompile it for future software releases.

As a second example, we consider the same rate law, but instead of allowing **React** to compute the derivatives by finite differences, which can be time consuming, we calculate them directly. The forms of the derivatives are given in the *Geochemical and Biogeochemical Reaction Modeling* text (see **Further Reading**). The code (omitting the global headers) is:

```
EXPORT double example2(int i, Reactant &r, GWBcontext &c)
{
   #include "ratelaw_param.h"

   if (QoverK > 1.0)
      return 0.0;
   else if (-1 == i)
      return rate_con * surface * (1.0 - QoverK);
   else if (H2O == i) // Derivative wrt water mass = zero.
      return 0.0;
   else if (i > H2O && i < c.Nbasis) // Other derivatives.
      return  -rate_con  *  surface  *  mtcoef[i]  *  QoverK  /
Bas[i].c;
   else
      return 0.0;
}
```

As a third example, we code a rate law that includes promoting and inhibiting species.

```
EXPORT double example3(int i, Reactant &r, GWBcontext &c)
{
   #include "ratelaw_param.h"

   if (-1 == i) {
      double cat = 1.0;
      for (int j=0; j<ncatal; j++)
         cat *= pow(acatal(j), pcatal(j));
      return cat * rate_con * surface * (1.0 - QoverK);
   }
   else
      return ANULL;
}
```

We consider in a fourth example how we might represent the rate law for a redox reaction.

```
EXPORT double example4(int i, Reactant &r, GWBcontext &c)
{
   #include "ratelaw_param.h"

   if (-1 == i) {
      double cat = 1.0;
      for (int j=0; j<ncatal; j++)
         cat *= pow(acatal(j), pcatal(j));
      return Wmass * cat * rate_con * (1.0 - QoverK);
   }
   else
      return ANULL;
}
```

In a final example, we write a function to calculate the rate of microbial metabolism, taking Ω to be unity. Note that, in this case, if we are to account for growth, we must update the microbial biomass (variable `biomass`); we also calculate the kinetic factors “FD” and “FA”, as well as the thermodynamic potential factor “TPF”, so they will be available in the program output.

```
EXPORT double example5(int i, Reactant &r, GWBcontext &c)
{
   #include "ratelaw_param.h"
   double aff, rpave, rprime, rprime0;
   double prod, prodKd1, prodKd2, prodKa1, prodKa2;

   if (-1 == i) {
      aff = QoverK * exp(-ATP_number*ATP_energy/ (8.3143*TK));
      aff = pow(aff, order1);
      aff = fmin(aff, 1.0);
      prod = 1.0;
      for (int j=0; j<ncatal; j++)
         prod *= pow(acatal(j), pcatal(j));

      prodKd1 = KD;
      prodKd2 = 1.0;
      for (int j=0; j<nspecD; j++)
         if (signD(j) > 0)
            prodKd1 *= pow(actD(j), powerD(j));
         else if (signD(j) < 0)
            prodKd2 *= pow(actD(j), powerD(j));

      prodKa1 = KA;
      prodKa2 = 1.0;
      for (int j=0; j<nspecA; j++)
         if (signA(j) > 0)
            prodKa1 *= pow(actA(j), powerA(j));
         else if (signA(j) < 0)
            prodKa2 *= pow(actA(j), powerA(j));

      TPF = 1.0 - aff;
      FD = nspecD > 0 ?
         prodKd2 / pow(prodKd1 + prodKd2, PKD) : ANULL;
      FA = nspecA > 0 ?
         prodKa2 / pow(prodKa1 + prodKa2, PKA) : ANULL;
      rprime = rate_con * (prod /
         (pow(prodKd1 + prodKd2, PKD) *
          pow(prodKa1 + prodKa2, PKA))) * TPF;
      if (Deltat > 0.0) {
         rprime0 = dndt0 / (biomass0 * Wmass);
         rpave = (1.0 - Theta) * rprime0 + Theta * rprime;
         biomass = biomass0 * exp((growth_yield * rpave -
            decay_con) * Deltat);
         biomass = fmax(biomass, 0.0);
      }
```

(cont'd)

```
      return rprime * biomass * Wmass;
   }
   else
      return ANULL;
}
```

The technique for updating biomass here is described in the previous topic, **Rate law scripts**.

Table 5.1 Internal parameters for defining rate laws. Parameters are case sensitive.

Parameter	*Meaning*
Chemical conditions	
pH	*p*H
logfO2	Log di-oxygen fugacity
Eh	Redox potential (V)
pe	pe
TDS	Dissolved solids (mg/kg)
IS	Ionic strength (molal)
SIS	Stoichiometric Ionic strength (molal)
chlorinity	Chlorinity (molal)
Watact	Water activity
soln_density	Solution density (g/cm^3)
soln_viscosity	Solution viscosity (cp)
alkalinity	Water alkalinity (mg/kg as $CaCO_3$)
vapor_pressure	Total gas (vapor) pressure (bar)
moles_vapor	Total moles of gases in vapor phase
EC	Electrical conductivity (uS/cm)
hardness	Total hardness (mg/kg as $CaCO_3$)
hardness_carb	Carbonate hardness (mg/kg as $CaCO_3$)

hardness_ncarb	Non-carbonate hardness (mg/kg as $CaCO_3$)
Phases	
Wmass	Mass of solvent water (kg)
soln_mass	Mass of fluid in nodal block (kg)
rock_mass	Mass of minerals in nodal block (kg)
soln_volume	Volume of fluid in nodal block (cm^3)
rock_volume	Volume of minerals in nodal block (cm^3)
vapor_volume	Volume of vapor phase in nodal block (cm^3)
Reactant properties	
name	Reactant name
rxn, reaction	Kinetic reaction, as a character string
mw	Mole weight of a kinetic mineral (g/mol)
mv	Mole volume of a kinetic mineral (cm^3/mol)
order1	Rate law order ω for Q/K term
order2	Rate law order Ω for (1 – Q/K) term
ncatal	Number of promoting and inhibiting species in rate law
catal (1), catal (2), …	Names of promoting and inhibiting species
pcatal (1), pcatal (2), …	Powers (exponents) of promoting and inhibiting species
xaffin	Name of cross-affinity mineral
catalyst	Name of catalyst
enzyme	Name of enzyme
KA	Half saturation constant K_A for forward reaction in enzyme model, and electron accepting reaction in microbial model
KD	Half saturation constant K_D for electron donating reaction in microbial model
KP	Half saturation constant K_P for reverse reaction, enzyme model
nspecD	Number of species in “K_D” product functions in the microbial model
specD(1), specD(2), …	Names of species in “K_D” products
powerD(1), powerD(2), …	Powers (exponents) of species in “K_D” products

signD(1), signD(2), …	Signs (+ for right side of reaction, - for left) of reaction coefficients for species in “K_D” products
nspecA	Number of species in “K_A” product functions in the microbial model
specA(1), specA(2), …	Names of species in “K_A” products
powerA(1), powerA(2), …	Powers (exponents) of species in “K_A” products
signA(1), signA(2), …	Signs (+ for right side of reaction, - for left) of reaction coefficients for species in “K_A” products
PKA	Power (exponent) for K_A term in microbial model
PKD	Power (exponent) for K_D term in microbial model
growth_yield	Growth yield, microbial model
decay_con	Decay constant (/sec), microbial model
ATP_energy	Free energy of ATP synthesis (J/mol), microbial model
ATP_number	Number of protons translocated in microbial model

Reactant status at nodal block

QoverK	*Q/K*
logQoverK	log *Q/K*
Q	Activity product Q for reaction
K	Equilibrium constant *K*
logQ	log *Q*
logK	log *K*
moles, mass	Mass of a kinetic mineral (mol)
dndt	Reaction rate (mol/sec) averaged over time step
dndt0	Instantaneous reaction rate (mol/sec) at beginning of time step
rate_con	Rate constant (in absence of promoting or inhibiting species: mol/cm^2 sec, molal/sec, or molal/cm^2 sec)
act_en	Reaction activation energy (J/mol)
pre_exp	Reaction pre-exponential factor (same units as rate constant)
sparea	Specific surface area of a kinetic mineral
surface	Total surface area of a kinetic or catalyst mineral (cm^2)

nucleus	Nucleation area of a kinetic mineral (cm^2/cm^3 fluid)
critSI	Critical saturation index for nucleation
a_enzyme	Enzyme molality or activity
acatal(1), acatal(2), …	Activities, molalities, fugacities of promoting and inhibiting species
actD(1), actD(2), …	Activities, molalities, fugacities of species in “K_D” products
actA(1), actA(2), …	Activities, molalities, fugacities of species in “K_A” products
biomass	Biomass (mg/kg) at end of time step
biomass0	Biomass (mg/kg) at beginning of step
TPF	Thermodynamic potential factor, microbial model
FD	Kinetic factor for donating reaction, microbial model
FA	Kinetic factor for accepting reaction, microbial model
mtcoef	Mass transfer coefficient
Temperature	
temperature, TC, Tempc	Temperature (°C)
TK	Temperature (K)
Domain	
Nnode	Number of nodal blocks in total
Nx	Number of nodal blocks along the x direction
Ny	Number of nodal blocks along the y direction
Length	Size of domain along the x direction (cm)
Width	Size of domain along the y direction (cm)
Height	Size of domain along the z direction (cm)
Radius1	Beginning radius of radial or spherical domain (cm)
Radius2	Ending radius of radial or spherical domain (cm)
Wedge_angle	Radial angle of radial or spherical domain (radians)
Indexing	
Inode	Node index
Ix, Xindex	Index of nodal block along the x direction
Jy, Yindex	Index of nodal block along the y direction

Nodal block	
Xposition	Position of center of nodal block along the x direction (cm)
Yposition	Position of center of nodal block along the y direction (cm)
Deltax	Length of nodal block (cm)
Deltay	Width of nodal block (cm)
Deltaz	Height of nodal block (cm)
bulk_volume, Vbulk	Bulk volume of nodal block (cm^3)
inert_volume, Vinert	Non-reacting volume in nodal block (cm^3)
Xfree	Free-flowing fraction of nodal block
freeflowing, Vfree	Volume of free-flowing zone in block (cm^3)
stagnant, Vstag	Volume of stagnant zone in block (cm^3)
Xarea	Area (cm^2) of nodal block normal to x direction
Yarea	Area (cm^2) of nodal block normal to y direction
Poros0	Sediment initial porosity, as a fraction
porosity	Sediment porosity, as a fraction
Time marching	
Time	Time (s)
Tstart	Starting time (s)
Tend	Ending time (s)
Deltat	Length of time step (s)
Theta	Time weighting variable
Xi	Reaction progress
PV	Pore volumes displaced

Table 5.2 "Helper functions" available in rate law scripts. Each function takes a character variable or string as an argument and returns a numeric value. Function names are case sensitive.

"Helper function"	*Returns*
`activity`(*species*)	Activity of a species or end member (or fugacity of a gas)
`fugacity`(*gas*)	Fugacity of a gas
`gamma`(*species*)	Activity coefficient of a species or end member; fugacity coefficient of a gas
`boltzman`(*surface species*)	Boltzman factor for a surface species
`mass`(*species*)	Mass of a species, mineral, end member, or reactant (mol)
`molality`(*species*)	Molal concentration of a free species
`xss`(*species*)	Mole fraction of an end member
`partial_pressure`(*gas*)	Partial pressure of a gas (bar)
`xgas`(*gas*)	Mole fraction of a gas in vapor phase
`dissolved`(*gas*)	Molal concentration of a dissolved molecular gas species
`totmolal`(*basis entry*)	Molal concentration of a component (a basis entry, before swapping) in the fluid
`totmoles`(*basis entry*)	Moles of a component in the fluid
`totmoles_rck`(*basis entry*)	Moles of a component in the rock (minerals and end members)
`totmoles_sys`(*basis entry*)	Moles of a component in the system
`totmoles_vap`(*basis entry*)	Moles of a component in the vapor phase
`totmoles_sor`(*basis entry*)	Moles of a component in the sorbate
`totmoles_col`(*basis entry*)	Moles of a component in mobile colloids
`totmoles_stg`(*basis entry*)	Moles of a component in the stagnant zone

`QovK`(*reactant*)	*Q/K* for a mineral, solid solution, or kinetic reaction
`logQovK`(*reactant*)	Affinity of a mineral, solid solution, or kinetic reaction
`reaction_rate`(*reactant*)	Rate of a mineral or kinetic reaction (mol/s)
`surf_capacitance`(*surface type*)	Specific cap. of a two-layer surface (F/m^2)
`surf_charge`(*surface type*) *or* `surf_charge0`(*surface type*)	Electrical charge at the 0-plane (C/m^2) of a sorbing surface
`surf_chargeb`(*surface type*)	Electrical charge at the β-plane (C/m^2)
`surf_charged`(*surface type*)	Electrical charge at the *d*-plane (C/m^2)
`surf_potential`(*surface_type*) *or* `surf_potential0`(*surface_type*)	Electrical potential at the 0-plane of a sorbing surface (V)
`surf_potentialb`(*surface_type*)	Electrical potential at the β-plane (V)
`surf_potentiald`(*surface_type*)	Electrical potential at the *d*-plane (V)
`sorb_area`(*surface_type*)	Area of a sorbing surface (m^2)
`Eh`(*redox couple*)	Redox potential of a redox couple (V)
`mw`(*species*)	Mole weight of a species or reactant (g/mol)
`mv`(*species*)	Mole volume, mineral or reactant (cm^3/mol)
`species_index`(*species*)	Index in program arrays of aqueous species
`mineral_index`(*species*)	Index of a mineral species
`ssoln_index`(*solid solution*)	Index of a solid solution
`gas_index`(*species*)	Index of a gas species
`surf_index`(*species*)	Index of a surface species
`surftype_index`(*surface type*)	Index of a surface type
`couple_index`(*couple*)	Index of a redox couple
`reactant_index` (*reactant*)	Index of a reactant

Table 5.3 Summary of syntax for rate law scripts. Keywords are not case sensitive.

`LET` *variable = expression*	Assignment of a numerical value or character string. The LET is optional. Variable names start with a letter and may contain digits and underscore characters
label: statement	Statement label, may be numeric or a character string
statement : statement	A colon (:) separates statements on a single line

`GOTO` *label*	Begin executing statements following label. `GOTO` and `GO_TO` are synonymous. The label may be a number or a character string
	Examples:
	`GOTO 100`
	`GOTO Line1`

`IF` *condition* `THEN`
statement or *label*
`ELSE`
statement or *label*
	Conditional execution; the `ELSE` clause is optional. The condition is false if it evaluates to 0, otherwise true.
	Example (appears on one line):
	`IF x > 0 THEN 20 ELSE PRINT “too small”`

`REM` *comment*	A remark. Any characters until the end of the line are ignored.
`+, -, *, /, ^, MOD`	Arithmetic operations, including exponentiation. Example: `x + y`
-	Negation: - y
`=, <, >, <=, >=, <>`	Comparisons of numeric or character values, such as `x < y` Comparison of character strings is case-insensitive
`AND, OR, XOR, NOT`	Logical operators. Each except `NOT` takes two operands, such as `x AND y`
`PI`	Value of π (3.14159…)

(*expression*)	Grouping and association: `x * ( y + z )`
`ABS, EXP, LN, LOG10, SQRT, SGN`	Absolute value, exponentiation, natural and common logarithms, square root, and sign functions. `EXP (x) is` e^x
`SIN, COS, TAN, ASIN, ACOS, ATAN, SINH, COSH, TANH`	Trigonometric functions (taking values in radians), their inverses, and hyperbolic trig functions
`MIN, MAX`	Minimum or maximum of two values
`DIM` *var1 (dimension), var2 (dimension)*	Declare and allocate memory for one or more arrays. In the example: `DIM A (3)` Array A has entries A (1), A (2), and A (3). Arrays may contain up to 1000 entries.
`PRINT` *item1, item2*	Print the numerical values or character strings in a list of items. Items are separated by commas, or by semicolons, in which case the corresponding values appear without an intervening space. Example: `PRINT “rate_constant =”, rate_con`
`PAUSE` *item1, item2*	Print any items listed and pause execution until the user touches a key. Syntax is the same as for the `PRINT` command
`RETURN` *expression*	Terminate the program and return the value of “expression”
`END`	Terminates the script. This statement is not required.

Using Gtplot

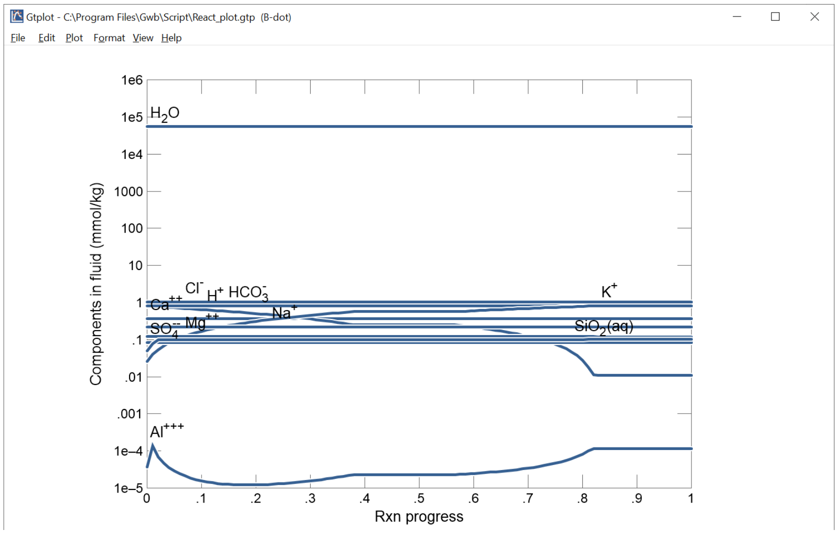
Gtplot is an interactive, mouse-driven program that produces various types of diagrams showing the results of tracing reaction paths with **React**.

After tracing a reaction path with **React**, start **Gtplot** by clicking on **Plot Results** on the **Results** pane, or selecting **Run → Gtplot** from **React**'s menubar. The program will read the calculation results from the ".gtp" file **React** produced and render them graphically. You can leave **Gtplot** active when you run further **React** simulations. Each time **React** completes a run, it signals **Gtplot**, which updates its display to reflect the new results.

You can also start **Gtplot** by opening any ".gtp" or ".gtc" file, or by clicking on the **Gtplot** icon on the Apps pane of the GWB dashboard. You can take input from a different dataset by choosing **File → Open → Data File...** from the menubar.

6.1 Initial plot

Gtplot opens a window on your screen, reads input data from a ".gtp" file, and renders the data as a two-dimensional plot. The window has three parts: a graphics area, a menubar with pulldown menus, and a frame showing the name of the input dataset and its activity model.



The initial plot shows how bulk fluid composition in the **React** calculation, expressed in terms of thermodynamic components, varies with reaction progress ξ. The program, however, can make different kinds of plots that show the following types of variables:

- Mass and volume of minerals and end members in the modeled system
- Concentration, activity, and activity coefficients of dissolved species
- Elemental composition of the fluid, minerals, sorbate, and the bulk system
- Composition of the fluid, minerals, sorbate, and the bulk system, expressed in terms of thermodynamic components
- Gas fugacity and partial pressure
- Saturation indices Q/K with respect to various minerals and solid solutions
- Bulk stable isotopic composition of the fluid, minerals, sorbate, and the entire system, as well as the composition of individual species, gases, and minerals
- Fraction of various components sorbed onto mineral surfaces
- Variables such as temperature, pH, and reacted mass

In addition, you can plot the results of the **React** calculation on any of the special plot types, such as Piper and Durov diagrams (see **Using Gtplot** in the ***GWB Essentials Guide***). Use the **Plot** menu to control the type of plot produced and the configuration of the plot.

Gtplot works using information about the system modeled, including its bulk composition (i.e., its composition in terms of Na^+, Ca^{++}, and so on), and the fluid's pH, TDS, carbonate speciation, and so on. Some of the "special" plots show variables labeled "HCO_3 + CO_3". Such plots show the sum of the concentrations in solution, in terms of electrical equivalents per kg fluid, of the free HCO_3^- and $CO_3^=$ species. The program calculates this sum from the fluid's carbonate alkalinity, as determined by the **React** program. Hence, the value shown for the variable depends not only on carbonate concentration, but the fluid's pH. In contrast, plots labeled "HCO3" represent the bulk carbonate concentration of the fluid, taking no account of the speciation of carbonate to the $CO_3^=$, HCO_3^-, and CO_2(aq) species.

When **Gtplot** starts for the first time in a given directory, it assumes a default configuration. Upon finishing, the program saves for its next run a dataset containing the current configuration of the plots. You can select options from the **File** menu to specify an alternative configuration (see **Loading and saving plot configuration**) or reset the entire program configuration.

6.2 XY Plot configuration

Choose **Plot → XY Plot…** to specify the configuration of the xy plot. Alternatively, you can alter the configuration by interacting with the graphical aspects of the plot, such as any data line, data label, axis, tick mark, label, or grid line. A **right-click** on an aspect of the plot displays a menu showing the options available. **Double-click** on an aspect to invoke the **XY Plot Configuration** dialog. Dragging an inner or outermost axis tick mark, number, or grid line will shift or scale, respectively, the axis range, (see **Using Gtplot** in the ***GWB Essentials Guide***).

The **X Axis** and **Y Axis** tabs let you select one variable to plot on the x axis, and one or more variables to plot on the y axis. Use **Variable type** and **Subtype** to control the type of variables to appear along the plot's axes. Variable type options are:

- **System**: Results describing progress (e.g. Rxn progress, Time elapsed), chemistry (e.g. Temperature, Pressure, and pH), the fluid (e.g. density), mass and volume (e.g. Solution mass and mineral mass), and so on
- **Reactants**: Net mass reacted, buffered values, kinetic rate law terms, and so on
- **Reaction rate**: Simple reactants, fixed and sliding buffers, and kinetic reactants
- **Species**: Concentration of chemical species; volume of minerals, end members, and lost gases
- **Components**: Bulk composition of the system, fluid, minerals, sorbate, vapor, stagnant zone, and mobile colloids, in terms of thermodynamic components
- **Elements**: Bulk composition, in terms of elements
- **Activity**: Aqueous species, end members, surfaces species; gas fugacity
- **Activity coefficients**: Aqueous species, end members, and surface species; fug. coefficients
- **Saturation**: *Q/K* of the fluid with respect to minerals and solid solutions
- **Nernst potentials**: Theoretical redox potentials for independent redox couples
- **Solid solutions**: Amount and composition of solid solutions
- **Surfaces**: Charge density, potential, sorbing surface area, and so on
- **Sorption**: Sorbed fraction and Apparent Kd's of sorbing components
- **Gases**: Partial pressure and other properties of gas species; dissolved gas concentrations
- **Isotopes**: Composition of the bulk system, fluid, rock, sorbate, and vapor, as well as individual species, for active isotope systems

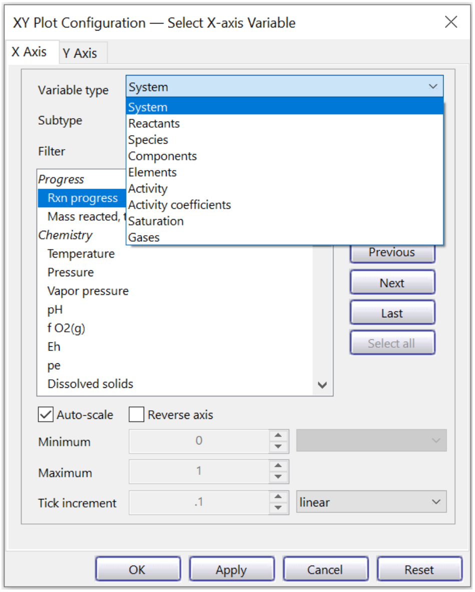
Use the **Filter** menu to set, for species or mineral saturation plots, a basis species to consider. In a typical reaction path calculation, so many dissolved species are considered, and saturation states are determined for so many minerals, that the data fit poorly on a single plot. For this reason, **Gtplot** can group species and minerals into sets of just those that contain a given basis species. You can choose to plot, for example, the species containing Na^+, or the saturation indices of minerals containing Al^{+++}. Alternatively, you may choose to consider all species and minerals.

The selection of a variable type displays a list of possible variables. Click on a variable you wish to select. Use **Ctrl+click** to select multiple variables. Use **Shift+click** to select a range of variables.

Use the following to specify the way the variables are plotted:

- **Variable type** to control the type of variables to appear along the plot's axis.
- **Subtype** to list subsets of the chosen variable type

- **Filter** to list, for species or mineral saturation plots, the set of just those that contain a given basis species.
- **Auto-scale** to set the data range for the axis to span the data to be plotted.
- **Minimum**, **Maximum**, and **Tick increment** to set the data range for the axis.
- **Reverse axis** to reverse the sense of the axis.
- **Units** to select alternative units, if any, for the axis.
- **As** to plot values as elemental equivalents, or as protonated or deprotonated species equivalents.
- **Type** to set the axis to a linear, log, or delta scale. A delta scale shows change in a variable's value from the initial point in the reaction path.
- **First**, **Previous**, **Next**, and **Last** to cycle through the variable choices.


XY Plot Configuration — Select X-axis Variable

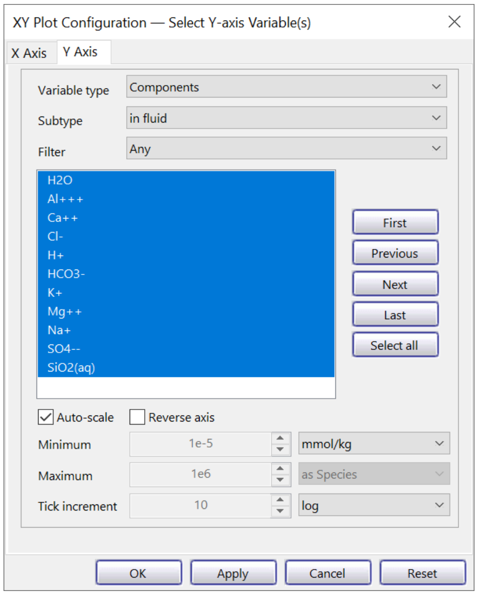
Variable type: System

☑ Auto-scale ☐ Reverse axis

Minimum: 0

Maximum: 1

Tick increment: .1 — linear


XY Plot Configuration — Select Y-axis Variable(s)

Variable type: Components

Subtype: in fluid

Filter: Any

☑ Auto-scale ☐ Reverse axis

Minimum: 1e-5 — mmol/kg

Maximum: 1e6 — as Species

Tick increment: 10 — log

6.3 Plot types

In addition to using the xy plot, you can plot the results of the **React** calculation on any of the special plot types, including as a ternary diagram, Piper diagram, Durov diagram, Schoeller diagram, Stiff diagram, radial plot, bar chart, and pie chart. Use the **Plot** menu to control the type of plot produced and the configuration of the plot. Each selection invokes a dialog box specific to the type of plot in question (see **Using Gtplot** in the ***GWB Essentials Guide***).

6.4 Editing plot appearance

Gtplot allows you to interactively modify many aspects of the diagram. The details of using active items, aspect menus, and dialogs to edit the plot appearance are given in the **Using Gtplot** section of the ***GWB Essentials Guide***.

6.5 Scatter data

Gtplot can overlay the data in a **GSS** data sheet (a .gss file) as "scatter data" on a plot of the results of a **React** calculation. The program can add scatter data to any of the plots it makes, except bar and pie charts. To coordinate the plotting of sample dates and times in the **GSS** data sheet on the reaction path, set an explicit starting date and time in **React**. To read a .gss file into **Gtplot**, select **File → Open → Scatter Data...**. Clear scatter data from a diagram on the **Open → Scatter Data...** dialog, with the **OFF** button.

The **Scatter data** section of the **Using Gtplot** chapter in the ***GWB Essentials Guide*** gives details on plotting scatter data from .gss files.

In GWB releases 7.0 and earlier, the program took scatter data from a specially formatted text file, rather than a .gss data sheet. Legacy scatter files are still supported, and are described in the **Scatter Data** chapter in the ***GWB Reference Manual***.

6.6 Loading and saving plot configuration

Upon finishing, **Gtplot** writes into the user's working directory a file, "gtplot_conf.gtc", containing the configuration of the current plots. When the program starts again in the same directory, it reads the file and assumes the same configurations.

Choosing **File → Reset Configuration** or the `reset` option from the command line (see Gtplot command line) returns the configuration for each plot type to its default state.

You can also save plot settings in ".gtc" configuration files. To do so, select **File → Save As...**. You can then specify that file as the configuration for a later **Gtplot** run from the command line (the `-c` flag; see Gtplot command line) or read it into **Gtplot** by selecting **File → Open → Configuration...**.

Exiting the program by choosing **File → Abort (No Save)** causes an immediate exit from the program; the plot configuration is lost.

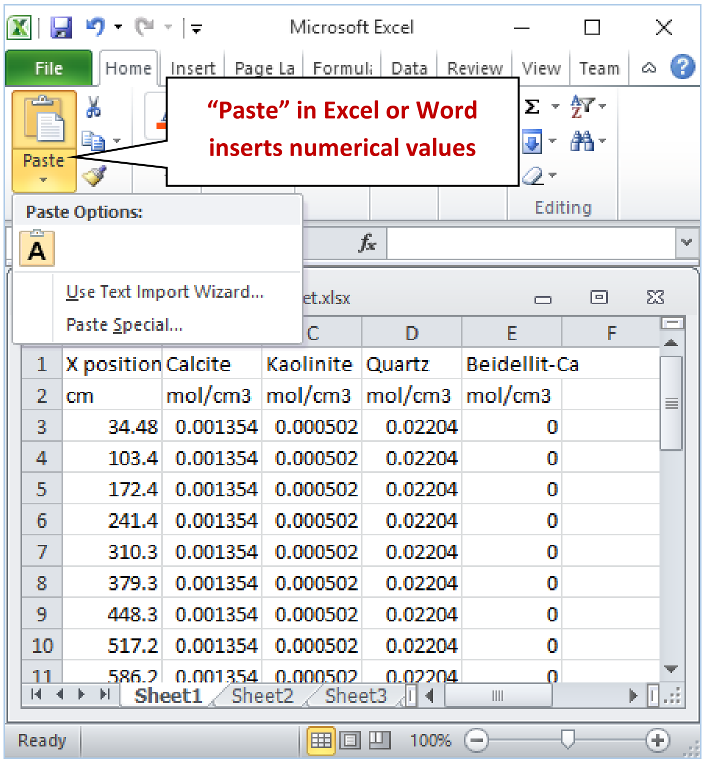
6.7 Exporting the plot

Gtplot makes it convenient to use the plots you create in articles, reports, presentations, and databases. You can copy the current plot to the clipboard and then paste it into a variety of applications, in a format meaningful to the application.

To copy a plot, use **Edit → Copy** or **Ctrl+C**. If you paste the plot into MS PowerPoint, it will appear as an EMF (an MS Enhanced Metafile) graphic object. Pasting into Adobe Illustrator places a native AI graphic.

If you paste the plot into MS Excel or a text editor, such as Notepad or MS Word, the numerical values of the data points that make up the lines on the plot will appear in spreadsheet format.

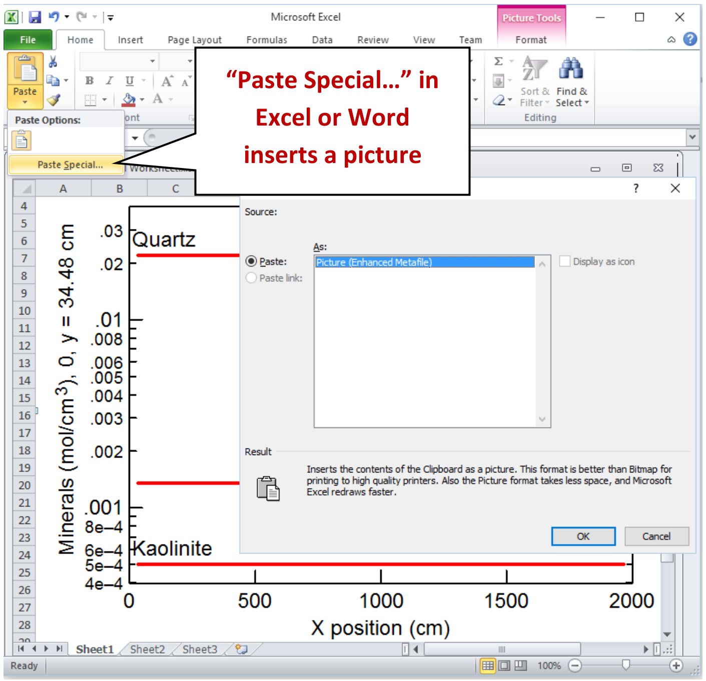
You can control the format in which the plot is copied to the clipboard by selecting **Edit → Copy As**. You can choose to copy the plot as an AI object, an EMF object, or a bitmap, or to copy the data points in the plot as tab delimited or space delimited text. Use the tab delimited option to paste the data into a spreadsheet program like MS Excel. For examining the data in a text file created with an editor like Notepad or MS Word, copying from **Gtplot** using the space delimited option and then pasting into the editor writes a nicely aligned table.



"Paste" in Excel or Word inserts numerical values

	A	B	C	D	E	F
1	X position	Calcite	Kaolinite	Quartz	Beidellit-Ca	
2	cm	mol/cm3	mol/cm3	mol/cm3	mol/cm3	
3	34.48	0.001354	0.000502	0.02204	0	
4	103.4	0.001354	0.000502	0.02204	0	
5	172.4	0.001354	0.000502	0.02204	0	
6	241.4	0.001354	0.000502	0.02204	0	
7	310.3	0.001354	0.000502	0.02204	0	
8	379.3	0.001354	0.000502	0.02204	0	
9	448.3	0.001354	0.000502	0.02204	0	
10	517.2	0.001354	0.000502	0.02204	0	
11	586.2	0.001354	0.000502	0.02204	0	

In MS Word or MS Excel, use **Paste Special…** to paste the plot as a picture instead.



“Paste Special...” in Excel or Word inserts a picture

Inserts the contents of the Clipboard as a picture. This format is better than Bitmap for printing to high quality printers. Also the Picture format takes less space, and Microsoft Excel redraws faster.

Use **File → Save Image…** to copy the plot, in your choice of formats (selected under **Save as type:**), from the graphics area to a file in the specified directory. The **Enhanced Metafile** option, for example, saves the plot image in a format that can be read by most art and illustration programs (see **Graphics Output** in the ***GWB Reference Manual***). The other file formats available are: **PNG**, **JPEG**, **TIFF**, **Bitmap**, **Adobe Illustrator**, **PDF**, **Scalable Vector Graphics**, **Compressed SVG**, **Encapsulated PostScript**, **Color PostScript**, and **B/W PostScript**.

When saving a PNG, JPEG, TIFF, or bitmap file, you may specify the quality of the saved image by choosing its resolution: High, Medium, Low, or Custom. Use **Custom…** to set the pixel width and height of the image, and to choose whether to preserve the aspect ratio of the plot. Use the **Spreadsheet File (Tab delimited)** or **Text File (Space delimited)** option to save the numerical coordinates of the data points on the plot as a table. The spreadsheet table can be read directly into many popular spreadsheet programs.

Certain graphics types support font embedding. PDF files should always display and print properly, regardless of fonts installed on the system. PostScript files should also, if you have used the option to embed fonts. If you may want to edit the PostScript file, however, you should deselect the option to embed fonts, because programs such as Adobe Illustrator may restrict your ability to edit a document using embedded fonts. To edit these files, be sure that all of the required fonts are installed on your computer (see **Font for data markers** in the ***GWB Reference Manual***).

When importing AI graphics to Adobe Illustrator, the program may prompt you to update the legacy text before you can edit the file. In this case, choose “Update”. You need to release the clipping mask before you attempt to edit individual elements of the plot. Use the “Ungroup” and “Group” functions when repositioning or modifying elements.

6.8 Gtplot command line

To run **Gtplot**, click on the **Gtplot** icon on the GWB dashboard, or open a “.gtp” or “.gtc” file. The program can also be initiated from the Windows command prompt by typing `gtplot.exe`. Starting the program in this way allows you to make use of the command line arguments described below.

Gtplot accepts a number of arguments from the command line. For example, the command

```
gtplot –i React_plot1.gtp –c Config1.gtc
```

causes **Gtplot** to read as input the file “React_plot1.gtp”, and to use the plot configuration stored in “Config1.gtc”.

The following options are available from the command line:

`<input_data>`
`-i <input_data>`
	Set the “React_plot.gtp” dataset, produced by a React run, which contains the data to be plotted. The program, by default, looks for file React_plot.gtp in the user’s working directory.

`-c <config_file>`	Set the configuration file to be read at startup. By default, the program reads the file gtplot_conf.gtc if it exists in the working directory.
`-scat <scatter_file>`	Take scatter data from the named dataset. By default, Gtplot does not plot scatter data.
`-graph <plot_type>`	Set the type of plot displayed at startup. By default, it is set from the configuration file. Choose from: xyplot, ternary, piper, durov, schoeller, stiff, radial, bar, pie, series, and time.
`-reset`	Set the default configuration at startup; do not read gtplot_conf.gtc.

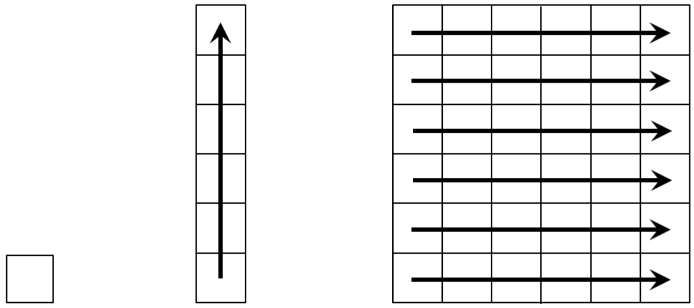
Table 6.1	Keyboard shortcuts in Gtplot
`Ctrl+Shift+A`	Copy plot to the clipboard as Adobe Illustrator AI file
`Ctrl+Shift+B`	Copy plot to the clipboard as a bitmap
`Ctrl+C`	Copy plot to the clipboard
`Ctrl+Shift+E`	Copy plot to the clipboard as an Enhanced Metafile
`Ctrl+F`	Refresh display
`Ctrl+Shift+G`	Get scatter data
`Ctrl+L`	Load a configuration file
`Ctrl+M`	Save the graphic image to a file
`Ctrl+O`	Open an input (“React_plot.gtp”) file
`Ctrl+P`	Print the plot
`Ctrl+Q`	Quit Gtplot, save configuration
`Ctrl+Shift+Q`	Abort Gtplot, do not save configuration
`Ctrl+R`	Reset the plot configuration
`Ctrl+S`	Save the current configuration to a file
`Ctrl+Shift+S`	Copy plot to the clipboard as Spreadsheet (tab delimited)
`Ctrl+Shift+T`	Copy plot to the clipboard as Text (space delimited)
`Ctrl+U`	Update the plot from the current input file
`Ctrl+X`	Reset data range on the x axis
`Ctrl+Y`	Reset data range on the y axis
`Ctrl+Z`	Reset data ranges on the x and y axes
`F1`	Open GWB Reaction Modeling Guide

Using Phase2

Phase2, as described in the Introduction to this guide, is a program designed for calculating a broad variety of two-dimensional diagrams for geochemical systems. The program works by tracing a stacked series of reaction paths that traverse the plot axes, then assembling the results of those paths into a **Phase2** diagram.

7.1 Conceptual model

A **Phase2** calculation has three main components: the initial system, the staging reaction path, and the set of scanning reaction paths.



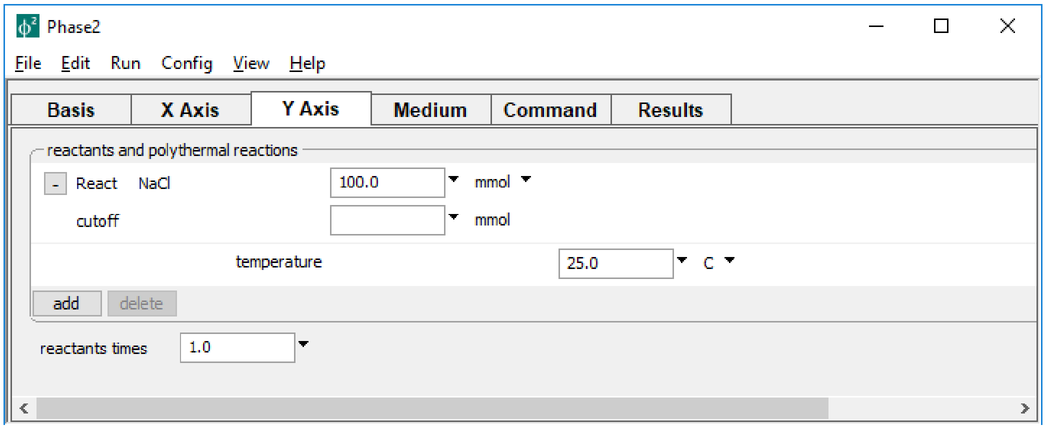
Phase2 begins a simulation by calculating the system's initial equilibrium state, which represents the bottom left corner of the diagram.

Beginning from that point, the program traces a staging path that defines the diagram's left margin. Along the staging path, **Phase2** changes the system, just as **React** does, by adding or removing reactants to vary the system's composition, changing the temperature, varying the fugacity of gases in an external buffer, and so on.

Starting from intermediate points along the staging path, the program initiates a series of scanning paths that trace a second reaction process. The two-dimensional grid of calculation results makes up the resulting **Phase2** diagram, which can be rendered using the **P2plot** program.

7.2 Staging and scanning paths

To configure the staging path, move to the **Y Axis** pane, where you can set reactants, activity and fugacity buffers, and fixed or varying temperature, much as you would on the **Reactants** pane in **React**



reactants and polythermal reactions

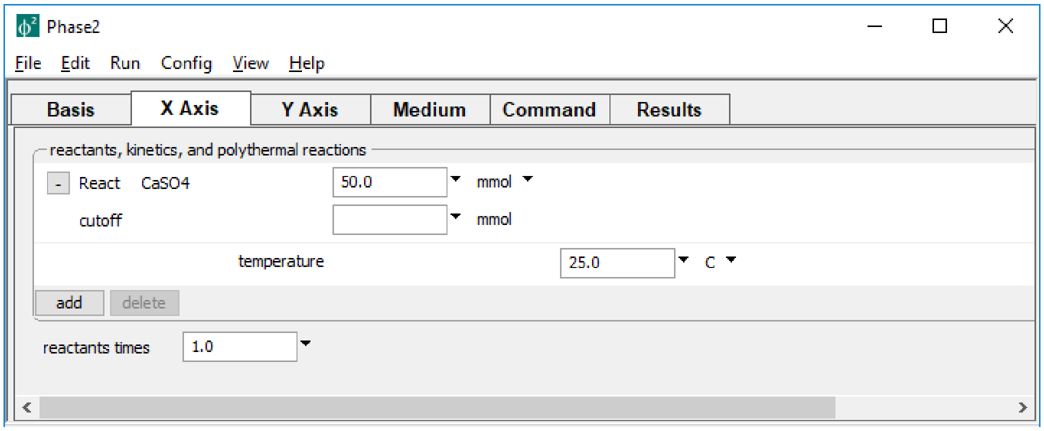
React NaCl 100.0 mmol

cutoff mmol

temperature 25.0 C

reactants times 1.0

In a similar fashion, you move to the **X Axis** pane to set reactants, buffers, and the temperature trajectory that define the scanning paths



reactants, kinetics, and polythermal reactions

React CaSO4 50.0 mmol

cutoff mmol

temperature 25.0 C

reactants times 1.0

You may equivalently configure the staging and scanning paths from the **Command** pane. The commands

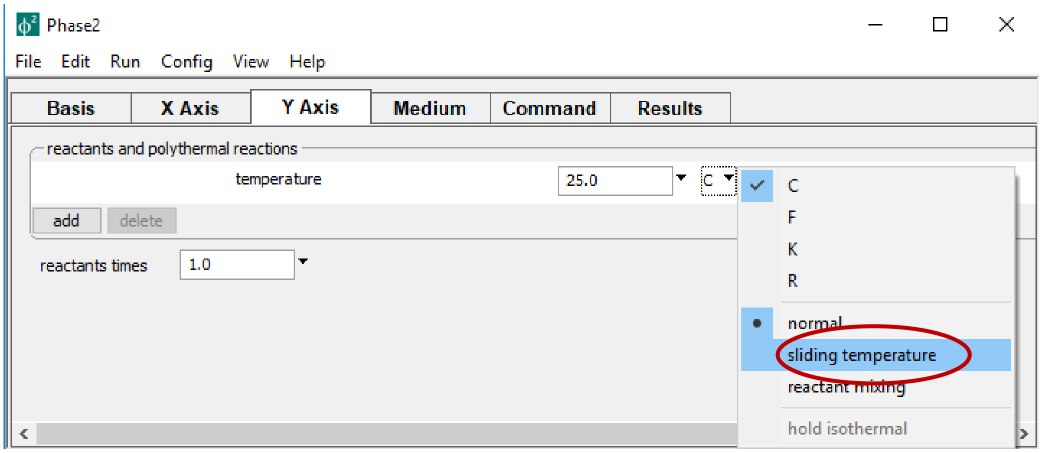
```
react y 100 mmol NaCl
react x 50 mol CaSO4
```

set the system’s NaCl content to vary over the staging path, and its $CaSO_4$ composition to vary along the scanning paths.

The `scope` command lets you specify the target axis for subsequent commands. The commands above might be equivalently entered

```
scope y
      react 100 mmol NaCl
scope x
      react 50 mmol CaSO4
```

To set a polythermal axis, move to the **X Axis** or **Y Axis** pane, click the pulldown next to the unit for temperature, and choose “sliding temperature”

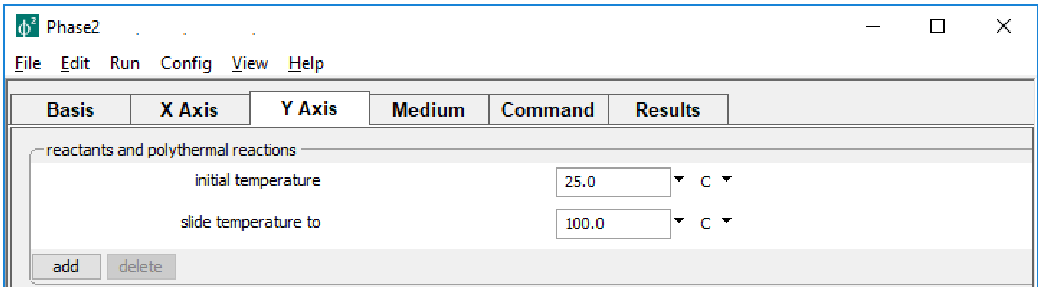


reactants and polythermal reactions

temperature 25.0 C

reactants times 1.0

Upon selecting this option, two fields are available: “initial temperature” and “slide temperature to”



reactants and polythermal reactions

initial temperature 25.0 C

slide temperature to 100.0 C

In this case, the system begins at 25°C and warms to 100°C. The initial temperature represents the left-side temperature on the **X Axis** pane, or, on the **Y Axis** pane, temperature at the diagram's lower left. Note that either axis, but not both, may be polythermal in a given run.

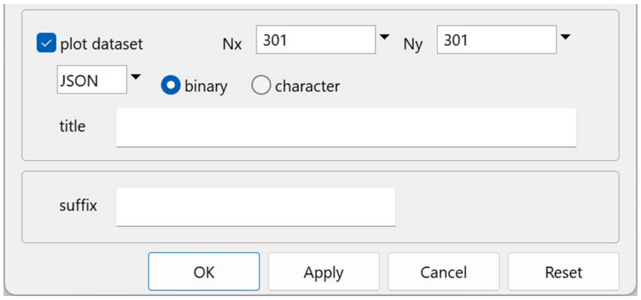
From the **Command** pane, use either the `temperature` command on a single line

```
temperature y initial = 25, final = 100
```

or within a scope block

```
scope y
   temperature initial = 25, final = 100
```

You control the diagram's resolution by setting the "Nx" and "Ny" fields on the **Config → Output…** dialog



☑ plot dataset Nx 301 Ny 301

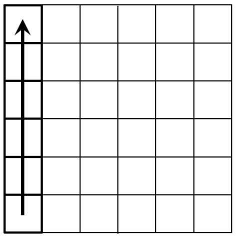
JSON ● binary ○ character

title

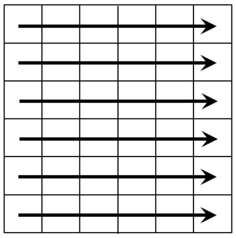
suffix

or using the "Nx" and "Ny" commands. Note that computing time and memory use increases as the product of the *x*- and *y*-direction resolutions, so you should avoid setting overly large values.

When you set a sliding buffer—a sliding pH, pe, Eh, activity, fugacity, or ratio—along one axis, **Phase2** automatically sets up the corresponding fixed buffer on the other. To make a "true" Eh-pH diagram, for example, you would set a sliding Eh path on the staging axis, and a sliding pH path on the scanning axis. In this case, the program fixes pH over the staging axis to the left-side value. Then, along each scanning path,



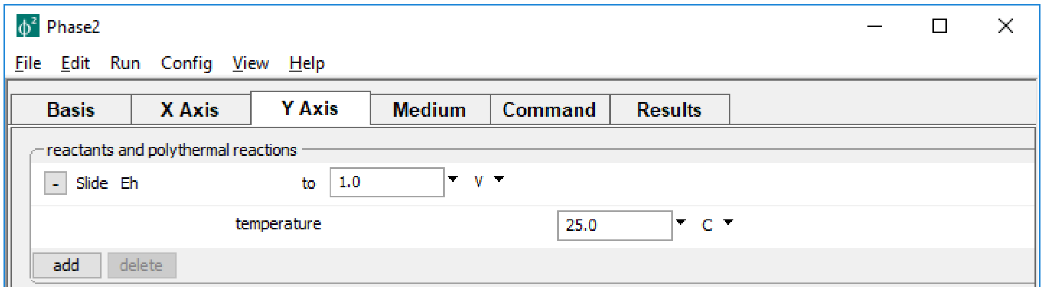
Slide Eh, fix pH



Slide pH, fix Eh

the program picks up Eh at the beginning of the path and holds it constant.

To see how this works, set a sliding Eh reactant on the **Y Axis** pane

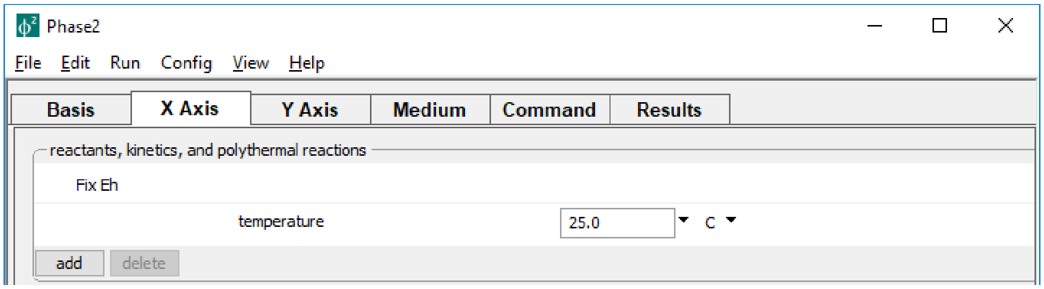


reactants and polythermal reactions

- Slide Eh to 1.0 V

temperature 25.0 C

then move to the **X Axis** pane, where the setting “fix Eh” has been automatically applied

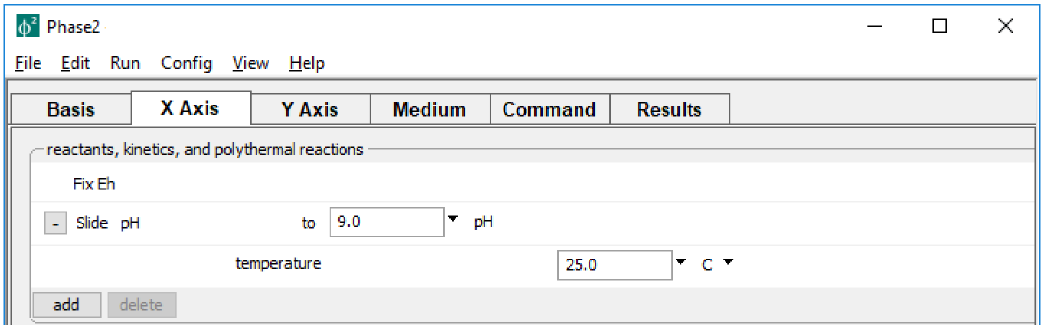


reactants, kinetics, and polythermal reactions

Fix Eh

temperature 25.0 C

While on the **X Axis** pane, add a sliding pH path



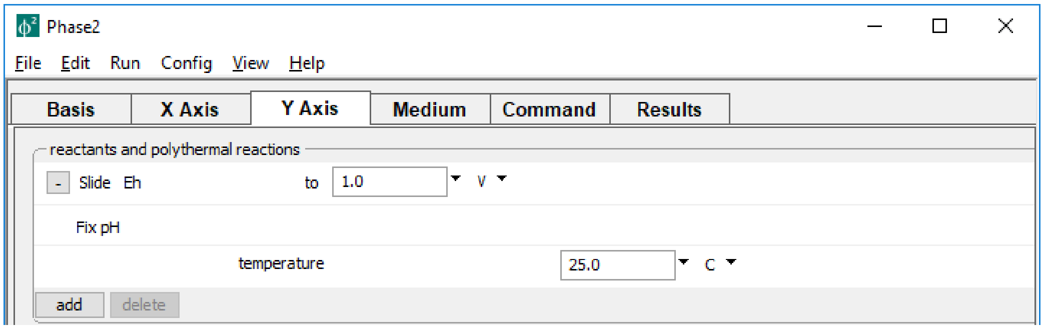
reactants, kinetics, and polythermal reactions

Fix Eh

- Slide pH to 9.0 pH

temperature 25.0 C

Returning to the **Y Axis** pane, you will see the “fix pH” setting already applied



reactants and polythermal reactions

Slide Eh to 1.0 V

Fix pH

temperature 25.0 C

Time in **Phase2** can vary along only the scanning axis. You can, therefore, define kinetic reactions, use time rates to describe mass transfer, set internal heat sources, and so on, only along the *x* axis. You may also implement the special model configurations—flow-through, flush, and flash models, as well as open-system vapor loss—only along the horizontal.

Since time advances only along *x*, kinetic reactions are held at their initial states of disequilibrium everywhere along the staging axis. A mineral that reacts according to a kinetic rate law, for example, neither precipitates nor dissolves as the program traces the staging path. Then, as the program traces the scanning paths, the mineral reacts at the rate specified by its kinetic rate law.

7.3 Progress variables

Once **Phase2** has completed a calculation, it writes the results to a file “Phase2_plot.p2p”, to be read by the plotting program **P2plot**. When you start **P2plot**, the plot axes will be labeled initially in terms of reaction progress. Start configuring your diagram by changing the axis variables to something more meaningful, and choosing whether to display the axes in linear or logarithmic coordinates.

Phase2 passes for each axis a list of potentially meaningful variables—the “progress variables”—to **P2plot**. The intent in choosing a progress variable along an axis is for its value to be invariant normal to the axis. In other words, scanning from one side of the diagram from a tick mark to the other, the progress variable should not change in value.

In many cases, such invariance is precise only if appropriate units are chosen. For example, if the *x* axis represents addition of Na^+, choosing units of mol gives a diagram in which the axis reads correctly at any point along *y*. Setting units of mol/kg, on the other hand, might not map precisely if the salinity, and hence kg of fluid, vary along *y*. To verify validity of a progress variable, simply use **P2lot** to plot it against the counter-axis.

Each simple reactant is listed as a progress variable such as “Mass reacted, Al+++”. If an original basis entry is added as a simple reactant and its total mass is not affected by other reactants, by charge rebalancing, or by special mass-transfer configurations, the component’s total mass (e.g. “Al+++ in system”) is also listed.

Consider the example

```
scope x
      react 1 mol Na+
      react 1 mol Cl-
scope y
      react 1 mol K+
      react 1 mol Cl-
```

On the *x* axis, “Mass reacted, Na+” and “Mass reacted, Cl-” are reported as progress variables, as is “Na+ in system”. On the *y* axis, “Mass reacted, K+” and “Mass reacted, Cl-” are available, along with “K+ in system”. No variable “Cl- in system” is available on either axis, though, because Cl^- concentration changes along both axes.

When an axis represents a sliding buffer, the buffered value and its logarithm are each available as a progress variable. If a time span is set for the scanning axis, time is also set as a progress variable. The “Mixing fraction” is listed as a progress variable along the scanning axis for flash models.

Temperature is available as a progress variable for a polythermal axis, for a simple sliding temperature run. For runs accounting for polythermal mixing or internal heat sources, in contrast, temperature is determined by the model, rather than prescribed, and so does not constitute a valid measure of progress along an axis.

7.4 Linear and log stepping

You should in general configure **Phase2** to step along an axis in a manner that matches how you envision rendering the final plot. If the axis is to be plotted on a linear scale, **Phase2** should step along the axis linearly, so that the plot points are evenly spaced along the axis. When an axis variable is to be rendered on a log scale, however, you should set up stepping along the variable’s logarithm.

How you choose between linear and log stepping depends on the variable being rendered on the axis. Most progress variables, such as time, temperature, or mass reacted, step linearly by default, but can be set to vary logarithmically by setting the `log` argument to the `delxi` command.

When you set a sliding activity or fugacity reactant (see **Sliding activity and fugacity** in this guide) activity, fugacity, activity ratio, and fugacity ratio vary linearly, by default. To have **Phase2** step along the variable’s logarithm, you constrain the reactant in terms of log units. The pH, Eh, and pe always vary linearly in sliding paths.

In a simulation spanning 100 days, for example, a run with linear stepping might report evenly-spaced results at 0, 50, 100, and so on, days. A run with log stepping, however, might give output at 1, 10, 100, and so on, days. The initial condition, corresponding to 0 days in this example, is not displayed in plots with logarithmic axes.

Consider a case in which you want to display the concentration of a component on a logarithmic scale. The commands

```
scope y
   react 1 mol Na+
   delxi .01 log
```

will add 0.01 moles of Na$^+$ in the first step (corresponding to Xi = .01), then continue taking log steps until the entire 1 mole has been added. A plot of Na$^+$ reacted might show the variable with evenly spaced ticks at 0.01, 0.1, and 1 mole.

You can combine the `delxi` and `dx_init` commands to give detail at the beginning of a path. The commands

```
scope y
   react 1 mol Na+
   delxi .01 log
   dx_init = 1e-5
```

will cause the program to add 10^{-5} moles in the first step (corresponding to Xi = 1e-5), and then proceed with log steps until the entire mole has been added. In this way, a plot of Na+ reacted might show the variable with evenly spaced ticks at 10^{-5}, 10^{-4}, 0.001, 0.01, 0.1, and 1 mol.

You may commonly find need for plotting the “total concentration” of a species (e.g. Na$^+$ in system) as a progress variable, rather than the amount reacted. In this case, the initial concentration (e.g., Na$^+$) specified in the **Basis** pane should be negligibly small, but nonzero. A value roughly 4 orders of magnitude smaller than the value corresponding to the first step works well. A constraint of 10^{-6} mol/kg Na$^+$ would work well for the first example described above, but 10^{-9} mol/kg might be used for the second example.

7.5 Settable variables

Phase2 allows you to alter the values of certain variables carried in the calculation. Table 7.1 lists variables that differ from those carried by **React** (see Table 2.1), along with their units and default values.

Table 7.1 Settable variables in Phase2 differing from React, their units and default values

Pane or Dialog	*Variable*	*Default*	*Unit*	*Description*
Stepping	`delxi x | y`	.01	–	Step size in reaction progress, log/linear stepping.
	`dx_init x | y`	*not set*	–	Initial step size in reaction progress.
	`step_increase x | y`	1.5	–	Maximum proportional increase in step size.
	`step_max x | y`	*not set*	–	Number of reaction steps the program may take in any single path.
	`threads`	*equal to number of cores*	–	Number of threads to spawn, for parallel execution.
Output	`dxprint x | y`	.1	–	Interval between printing results.
	`Nx, Ny`	301, 301	–	Number of data points in the plot grid

To change a variable, type the new value on the **Basis** or **Medium** pane or in the dialog boxes under **Config**, or enter as a command the variable name followed by the new value. Examples:

```
Nx = 101
delxi y = 0.1 log
dx_init y = 1e-10
```

Variables accepting an `x` or `y` argument can alternatively be set within a scope block:

```
scope y
   delxi = 0.1 log
   dx_init = 1e-10
```

Values are entered in the units carried internally by the program, as listed in Table 2.1 or Table 7.1. To restore a variable to its default value, enter a blank field in the dialog box, or on the **Command** pane type its name without a value or followed by a ”?”:

```
Nx
Nx = ?
```

Current settings can be viewed at any time on the relevant dialog box, or using the command `show variables`.

7.6 Multicore execution

Phase2 is multithreaded in such a way that multicore computers trace the scanning paths in parallel. Like **X1t** and **X2t**, described in the ***GWB Reactive Transport Modeling Guide***, you can control the number of threads **Phase2** spawns on the **Config → Stepping...** dialog, or with the `threads` command.

From the same dialog, or with the `pluses` command, you can control the amount of output shown on the **Results** pane after each step. Like **X1t** and **X2t**, **Phase2** reports upon completion of a calculation the computing time needed to execute parallel and serial sections of the code, as well as the clock time elapsed over the run.

7.7 Running a model

Once you have configured a simulation, you are ready to run it. A number of example input files for **Phase2** are installed with the GWB Professional package, in subdirectory “Script” within the installation directory (e.g., in “\Program Files\GWB\Script”). The example files have “.ph2” extensions. To give the program a test drive, double-click on one of the files.

Start the simulation in any of several ways: selecting **Run → Go**, pressing **Ctrl+G**, typing `go` on the **Command** pane, or clicking the **Run** button on the **Results** pane. You can watch on the **Results** pane as the model follows the simulation procedure, first computing the initial conditions, then tracing the staging reaction path, and finally tracing each scanning reaction path.

When the calculation is complete, click on the **Plot Results** button to launch **P2plot**. This program lets you render the simulation results graphically, as two-dimensional diagrams, in which you render the results as maps of species predominance or mineral assemblages. You can also color map and contour any variable. Additionally, you can plot cross-sections through the diagram. The chapter Using P2plot in this guide gives further details.

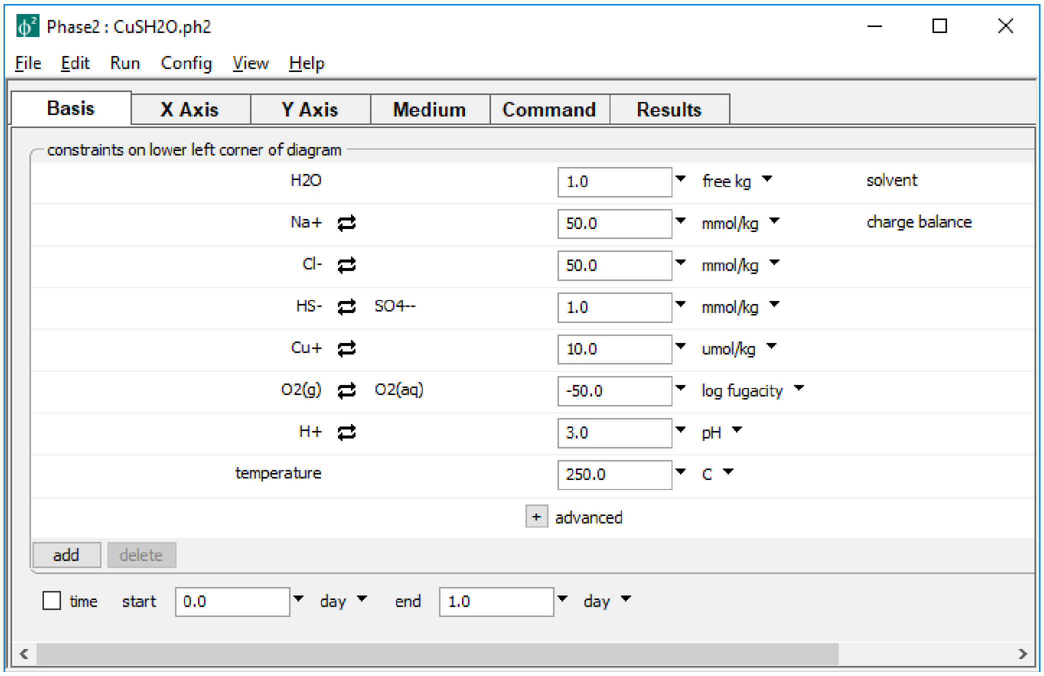
You can have the model produce text-format or “print” results describing the chemical state, along with the isotopic state, if applicable, at each stage of the calculation. To do so, select **Config → Output…** and then select the “print dataset” option. The program will write results at intervals in the reaction progress (ξ) set by variable “dxprint” on the **Config → Output…** dialog. When the model is complete, click on **View Results**. The print option can produce large amounts of output and, therefore, is by default turned off.

After finishing a simulation, you can revise the model configuration and run it again. When the new simulation is complete, any instances of **P2plot** running on your computer will automatically update their plots to reflect the latest results. You can open several **P2plot** windows on your computer at the same time, allowing you to view results plotted in different ways.

By clicking on **Run → Go Initial** or typing `go initial`, you cause the program to evaluate the initial conditions without tracing either the staging or scanning reaction paths. You can also click on **Run → Go Y** or type `go y` to trace the staging path (the left edge of the diagram). Similarly, click on **Run → Go X** or type `go x` to trace a single scanning path (the bottom edge of the domain). Once done, you can render the initial conditions with **P2plot**, as you could the results of any simulation. These features are handy because they let you verify that the initial system or either reaction path is set up correctly, without completing the entire simulation.

7.8 Example: Speciation diagram

As a first example, we construct a “true” fO_2-pH diagram for the copper-sulfur-water system at 250°C. Double-click on file “CuSH2O.ph2” to launch **Phase2** and move to the **Basis** pane


Phase2 : CuSH2O.ph2
File Edit Run Config View Help
Basis X Axis Y Axis Medium Command Results
add delete
+ advanced

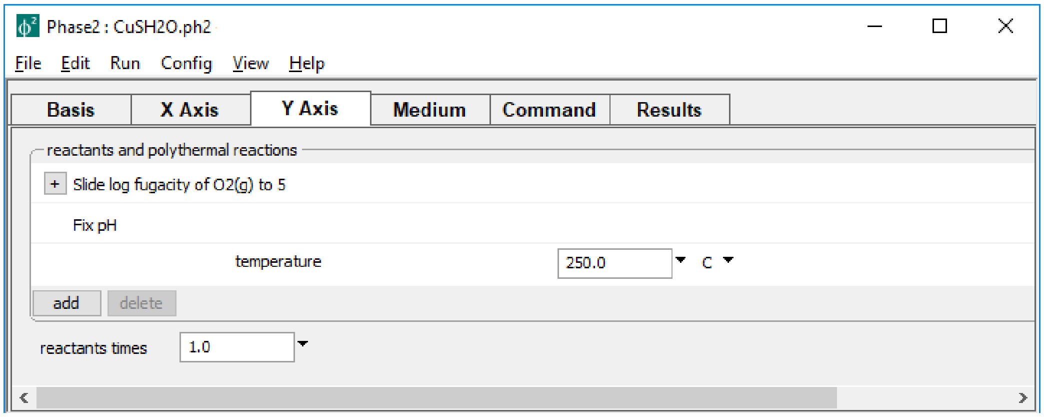
constraints on lower left corner of diagram

H2O		1.0	free kg	solvent
Na+ ⇄		50.0	mmol/kg	charge balance
Cl- ⇄		50.0	mmol/kg	
HS- ⇄	SO4--	1.0	mmol/kg	
Cu+ ⇄		10.0	umol/kg	
O2(g) ⇄	O2(aq)	-50.0	log fugacity	
H+ ⇄		3.0	pH	
temperature		250.0	C	

□ time start 0.0 day end 1.0 day

Here we constrain the chemical system at the bottom-left corner of the diagram: the background electrolyte is 50 mmol/kg NaCl, and the fluid contains 1 mmol/kg sulfur and 10 µmol/kg copper. The corner marks a log fugacity of –50 and a pH of 3.

The **Y Axis** pane sets up the staging axis


Phase2 : CuSH2O.ph2
File Edit Run Config View Help
Basis X Axis Y Axis Medium Command Results
add delete

reactants and polythermal reactions

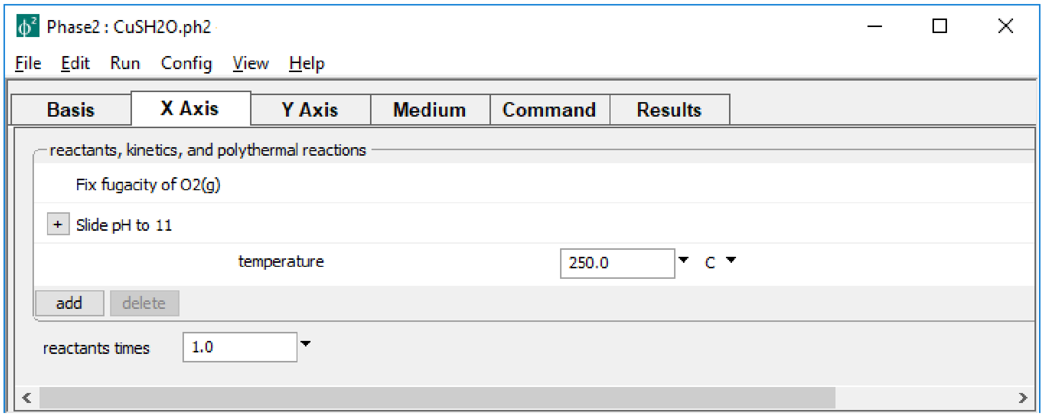
+ Slide log fugacity of O2(g) to 5

Fix pH

temperature 250.0 C

reactants times 1.0

such that the logarithm of oxygen fugacity slides to 5, while pH holds constant. In contrast, the **X Axis** pane



reactants, kinetics, and polythermal reactions

Fix fugacity of O2(g)

+ Slide pH to 11

temperature 250.0 C

reactants times 1.0

fixes fugacity, but slides pH to 11.

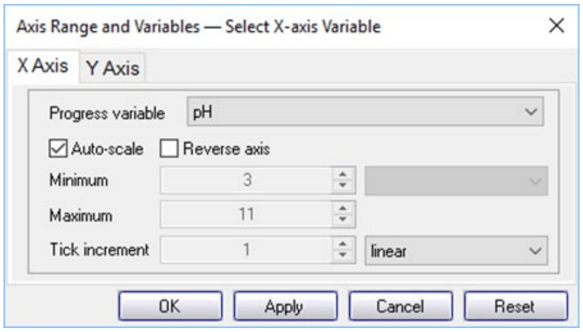
You can alternatively use the commands

```
T = 250 C

Na+           = 50 mmol/kg
Cl-           = 50 mmol/kg
balance on Na+

swap HS- for SO4--
HS-           =  1 mmol/kg
Cu+           = 10 umol/kg

swap O2(g) for O2(aq)
O2(g)         = -50 log fugacity
pH            = 3

scope y
   slide log fugacity of O2(g) to 5
scope x
   slide pH to 11
```

from the **Command** pane to configure the same calculation.

Move to the **Results** pane and click Run to trigger the calculation. When the program finishes, choose Plot Results to open dataset "Phase2_plot.p2p", which holds the calculation results, in program **P2plot**.

In **P2plot**, select **Plot → 2D Diagram**. When the diagram displays, double-click on one of the axis variables (or go to **Format → Axis Range and Variables…**) and on the **X Axis** tab set "Progress Variable" to "pH"



Axis Range and Variables — Select X-axis Variable

X Axis | Y Axis

Progress variable: pH

☑ Auto-scale ☐ Reverse axis

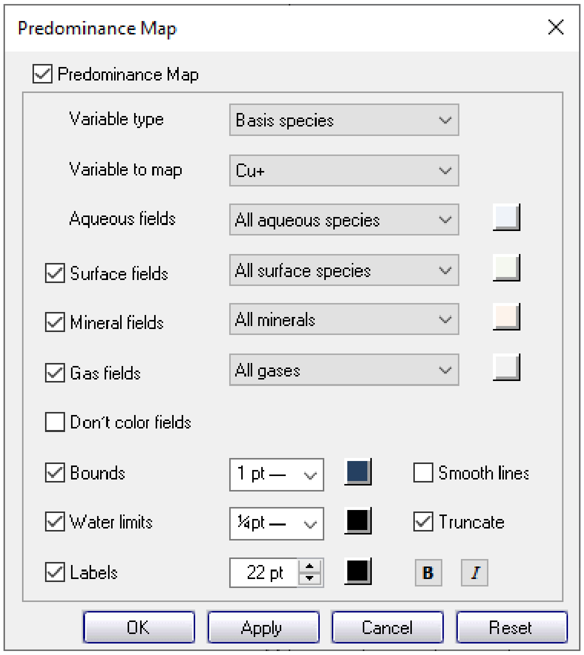
Minimum: 3

Maximum: 11

Tick increment: 1 linear

Then, on the **Y Axis** tab, set the variable for the *y* axis to “log f O2(g)”.

Next, click on **Format → Predominance Map…** to expose the Predominance Map dialog. Check the box next to “Predominance Map”, if it is not already, and next to “Variable to map” choose “Cu+”



Predominance Map

☑ Predominance Map

Variable type: Basis species

Variable to map: Cu+

Aqueous fields: All aqueous species

☑ Surface fields: All surface species

☑ Mineral fields: All minerals

☑ Gas fields: All gases

☐ Don't color fields

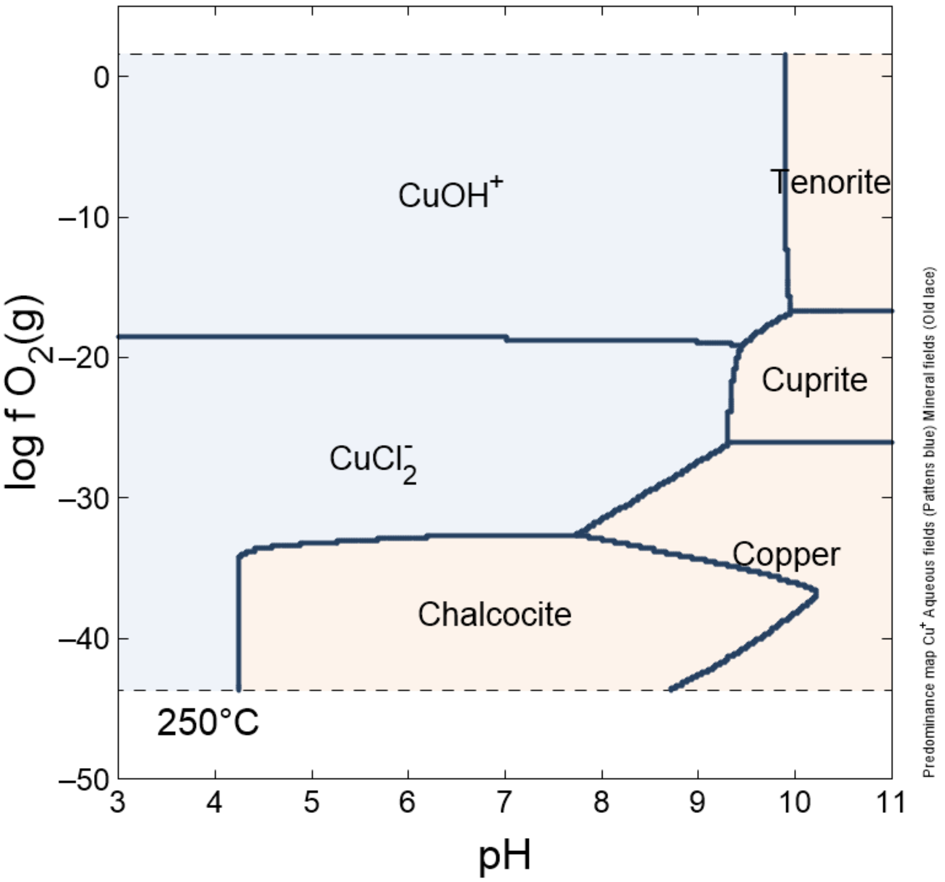
☑ Bounds: 1 pt — ☐ Smooth lines

☑ Water limits: ¼pt — ☑ Truncate

☑ Labels: 22 pt **B** *I*

Click on **Apply**.

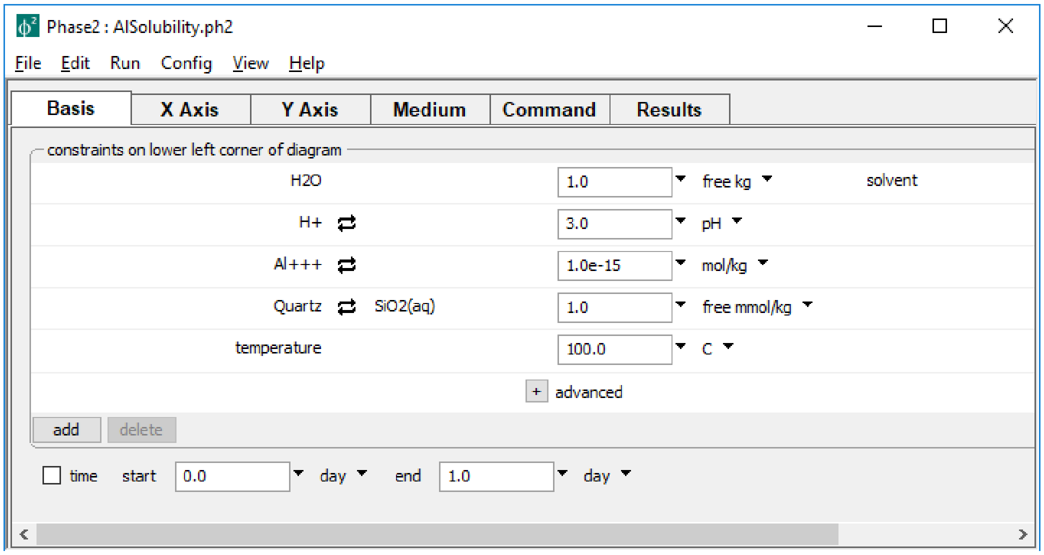
The resulting diagram shows the predominant copper-bearing species and minerals at 250°C as a function of pH and oxidation state



The plot is truncated at low and high fO_2 by the upper and lower stability limits of water. To extend the diagram, return to the **Predominance Map** dialog, uncheck “Truncate”, and click on **Apply**. To diagram predominant sulfur-bearing species, return to the dialog, change the “Variable to Map” to “SO4--", and once again hit **Apply**.

7.9 Example: Mineral solubility

The input file “AlSolubility.ph2” calculates the solubility of aluminum in the presence of quartz as a function of pH and the system’s aluminum content. Double-click the file to launch **Phase2**. The **Basis** pane



constraints on lower left corner of diagram

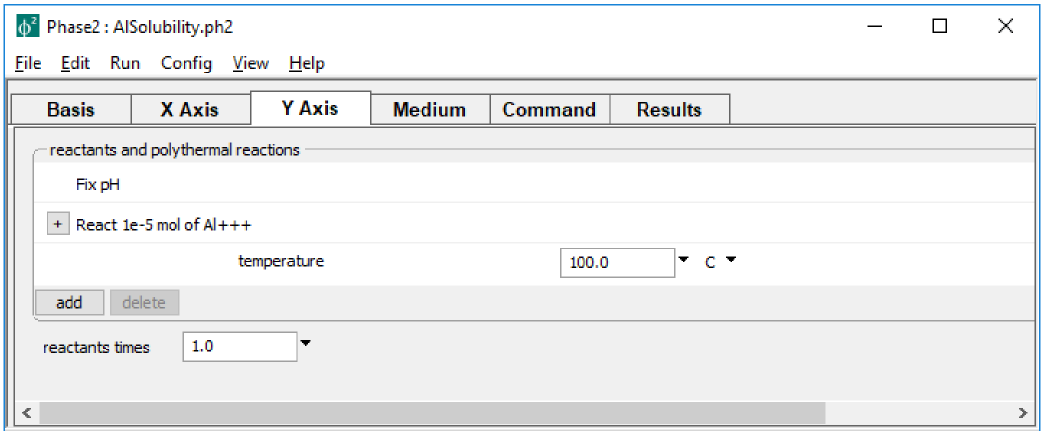
H2O	1.0	free kg	solvent
H+ ⇄	3.0	pH	
Al+++ ⇄	1.0e-15	mol/kg	
Quartz ⇄ SiO2(aq)	1.0	free mmol/kg	
temperature	100.0	C	

+ advanced

□ time start 0.0 day end 1.0 day

describes the starting point for our calculation, a pH 3 fluid in equilibrium with quartz at 100°C. The fluid carries a near-zero mass of dissolved aluminum. We don’t include a background electrolyte, so charge balancing is disabled.

The **Y Axis** pane



reactants and polythermal reactions

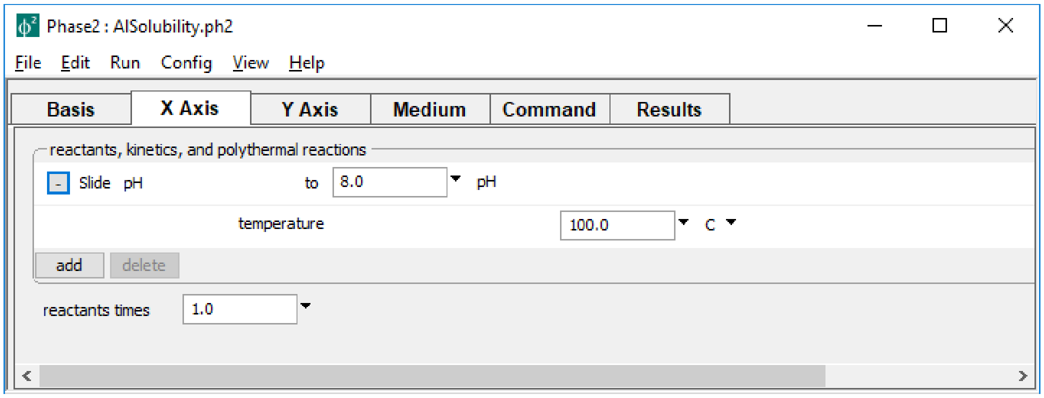
Fix pH

+ React 1e-5 mol of Al+++

temperature 100.0 C

reactants times 1.0

shows the staging path, along which 10 µmol of Al^{+++} is titrated into to the system, while pH is held constant. On the **X Axis** pane



reactants, kinetics, and polythermal reactions

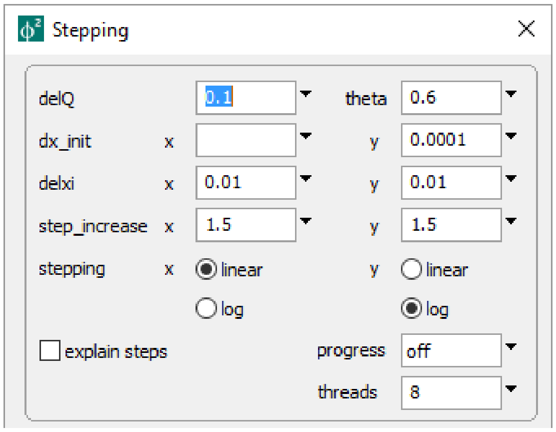
- Slide pH to 8.0 pH

temperature 100.0 C

reactants times 1.0

pH is set to slide from 3, the starting point, to a final value of 8.

On the **Config → Stepping…** dialog



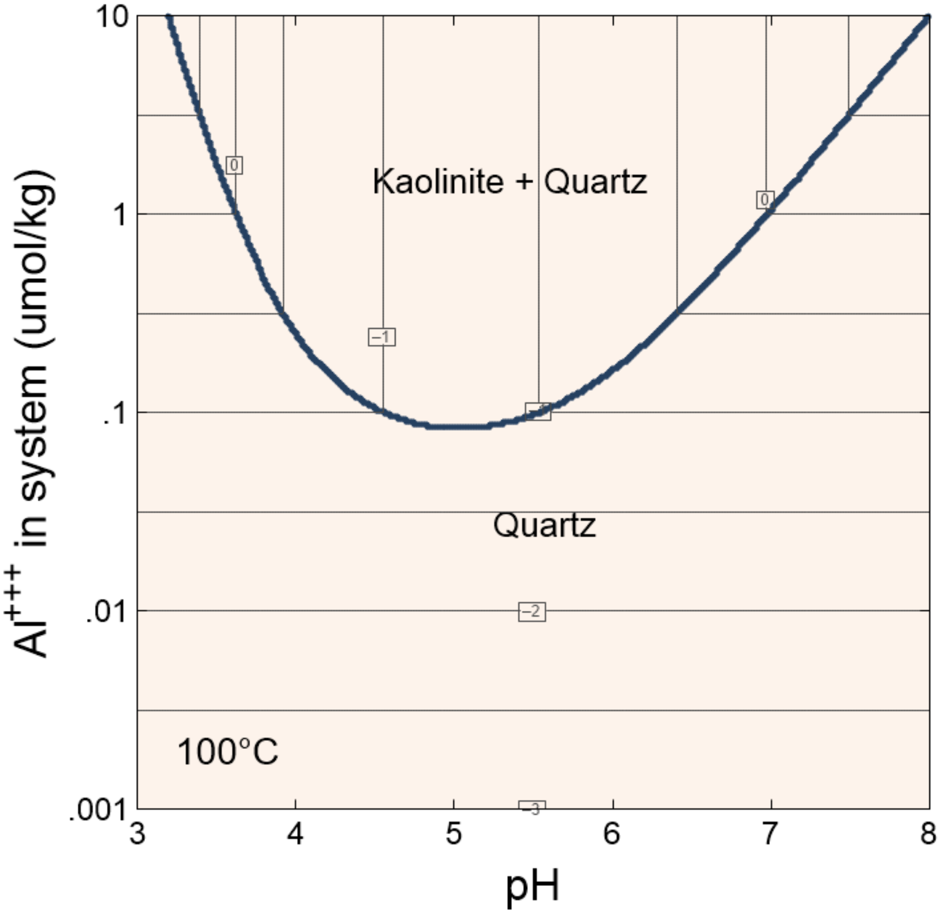
delQ		0.1	theta	0.6
dx_init	x		y	0.0001
delxi	x	0.01	y	0.01
step_increase	x	1.5	y	1.5
stepping	x	● linear ○ log	y	○ linear ● log
□ explain steps			progress	off
			threads	8

we set logarithmic stepping along the *y* axis, with an initial step size of 10^{-4}. Since we're ultimately adding 10^{-5} moles of Al^{+++}, **Phase2** will add $10^{-4}\times10^{-5} = 10^{-9}$ moles to the initially negligible mass on the first step, then exponentially larger aliquots as it takes subsequent steps.

Press the Run button on the **Results** pane to trace the calculation. Start **P2plot** by clicking on the Plot Results button and select **Plot → 2D Diagram**. Double-click on the *x* axis label and set "Progress Variable" to "pH". Move to the **Y axis** tab and set "Progress Variable" here to "Al+++ in system" with a log scale. Adjust the tick increment on each axis, if you wish, by de-selecting "Auto-scale", and click **OK**.

Next, select **Format → Assemblage Map…**, check the box at the top of the dialog, and click **OK**. Select **Format → Contour Map…** and check the box at the top of the dialog. Next to "Variable type", choose "Components", set "Subtype" to "in fluid", set "Variable to contour" to "Al+++", and for "Variable units", set "umol/kg". Make sure "Type of plot" is "Log", adjust the point size and color of the contour lines to your liking, and click OK.

The resulting diagram

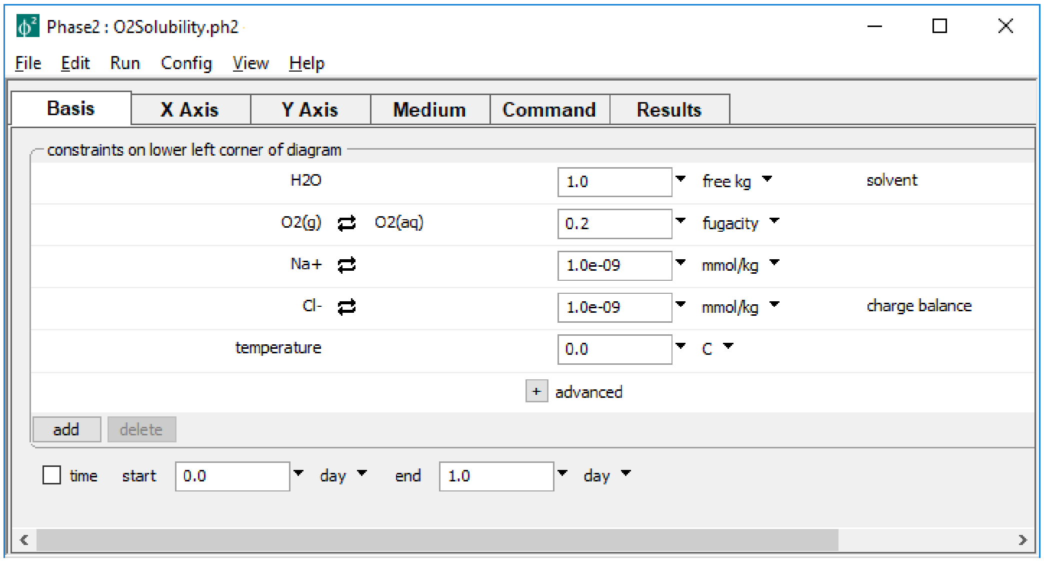


delineates the stability of kaolinite in the presence of quartz, and contours the logarithm of aluminum solubility, in µmol/kg.

7.10 Example: Gas solubility

In this example we diagram the solubility in water of atmospheric O_2(g) as a function of salinity and temperature. Our strategy is to define a dilute fluid in equilibrium with the atmosphere at 0°C to occupy the lower-left corner of the diagram. We will then set temperature to increase along the staging axis, and salinity along the scanning axis, while holding O_2(g) fugacity constant across both axes.

Begin by double-clicking on file“O2Solubility.ph2” to launch **Phase2**. The **Basis** pane

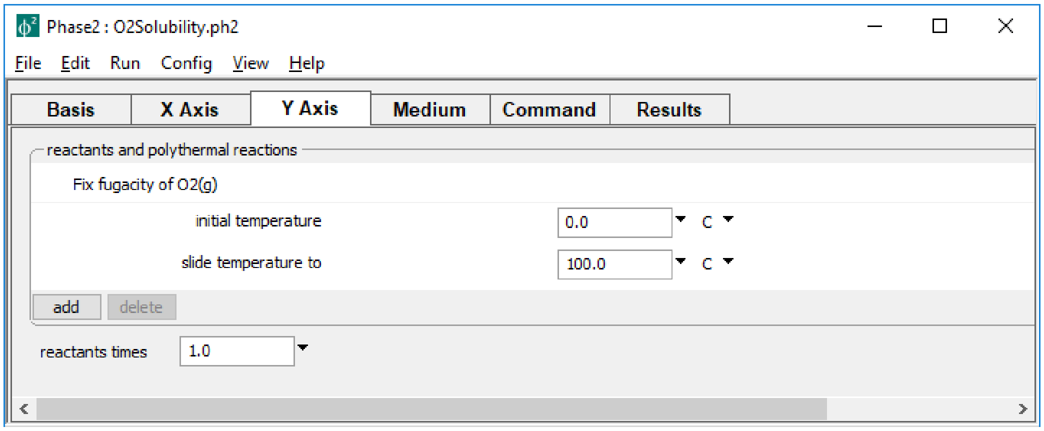


constraints on lower left corner of diagram

H2O	1.0	free kg	solvent
O2(g) ⇄ O2(aq)	0.2	fugacity	
Na+ ⇄	1.0e-09	mmol/kg	
Cl- ⇄	1.0e-09	mmol/kg	charge balance
temperature	0.0	C	

□ time start 0.0 day end 1.0 day

sets the bottom left corner of the diagram, whereas the **Y Axis** pane



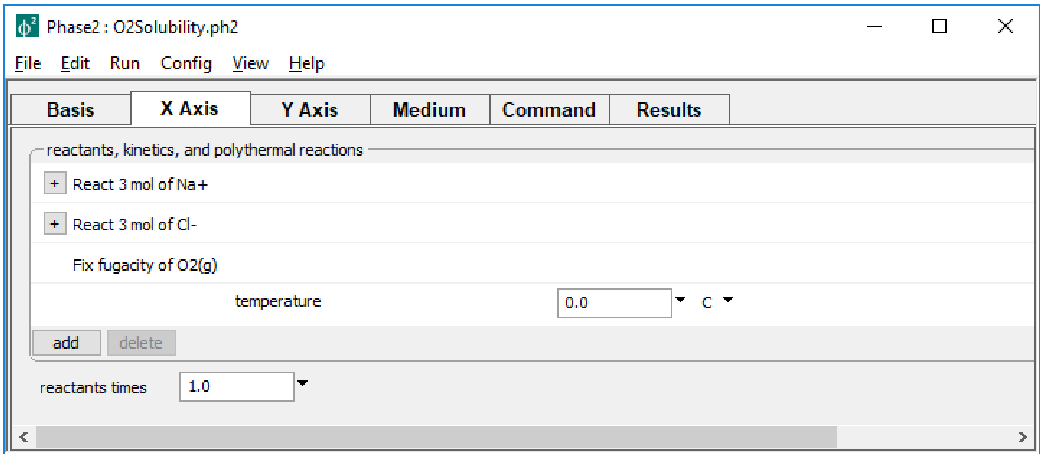
reactants and polythermal reactions

Fix fugacity of O2(g)

initial temperature	0.0	C
slide temperature to	100.0	C

reactants times 1.0

slides temperature along the staging path, and the **X Axis** pane



reactants, kinetics, and polythermal reactions

React 3 mol of Na+

React 3 mol of Cl-

Fix fugacity of O2(g)

temperature 0.0 C

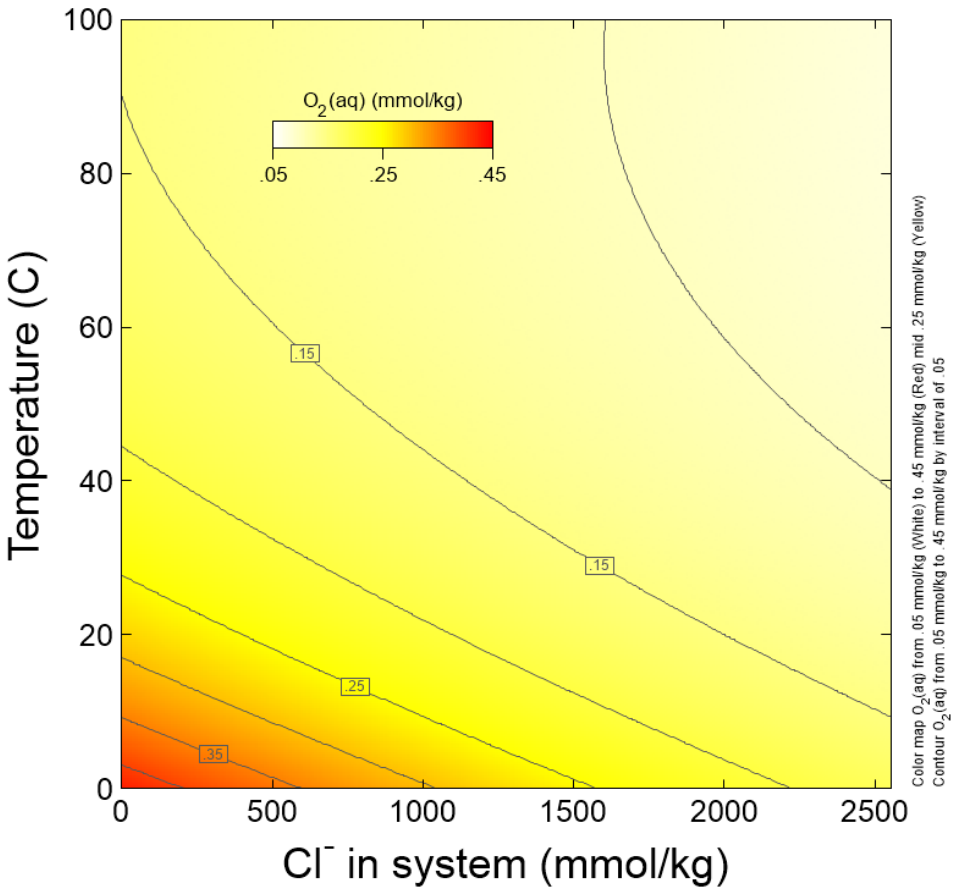
reactants times 1.0

sets up a titration of 3 mol of Na^+ and Cl^- as simple reactants along the scanning paths.

The simulation, then, consists of a series of isothermal titration paths, each traced at fixed oxygen fugacity, and each maintaining the temperature of its point of departure from the staging path. Press the Run button on the **Results** pane to trace the calculation.

Press Plot Results to launch **P2plot**, then select **Plot → 2D Diagram**. Double-click on one of the axis variables (or go to **Format → Axis Range and Variables…**), then set the *x* axis progress variable to “Cl- in system”, with units of “mmol/kg”, and the *y* axis variable to “Temperature”. Now, select **Format → Contour…** to expose the contour plot dialog. Choose “Species”, “Aqueous”, “O2(aq)”, “mmol/kg”, and “Linear” in the top five fields, then check the box next to “Contour Lines” and click on **Apply**. Under **Format → Color Map…**, choose the same settings and hit **Apply**.

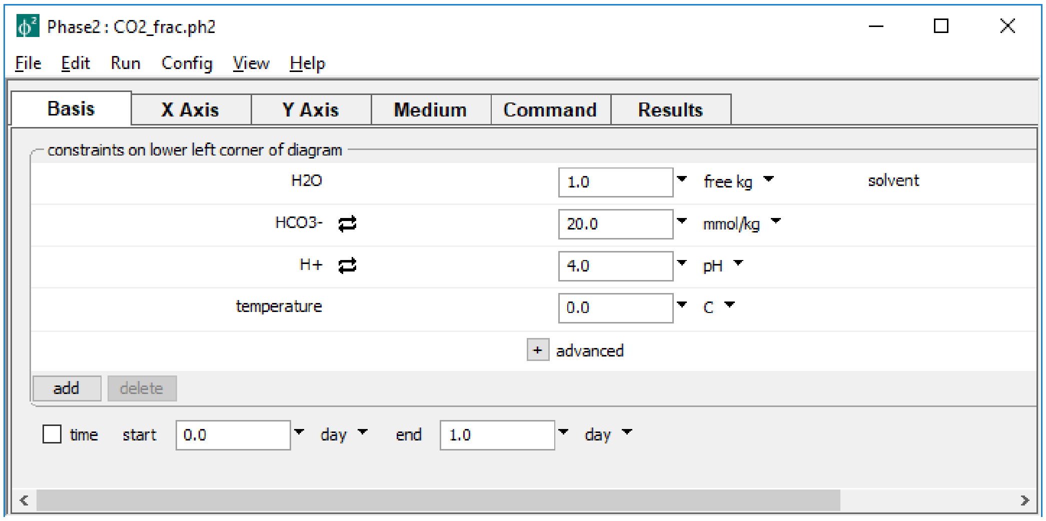
The resulting diagram shows how oxygen solubility decreases with increased temperature and salinity



The diagram reflects how the log *K* of the reaction between O_2(g) and O_2(aq) decreases with temperature, while the activity coefficient of O_2(aq) increases with ionic strength. The latter effect is commonly known as the “salting out” effect.

7.11 Example: Isotope fractionation

In this example, we diagram the ^{13}C composition of CO_2(g) in equilibrium with a carbonate-bearing fluid of known carbon isotopic composition, as a function of pH and temperature. Double-click on file “CO2_frac.ph2” and when **Phase2** opens, move to the **Basis** pane



constraints on lower left corner of diagram

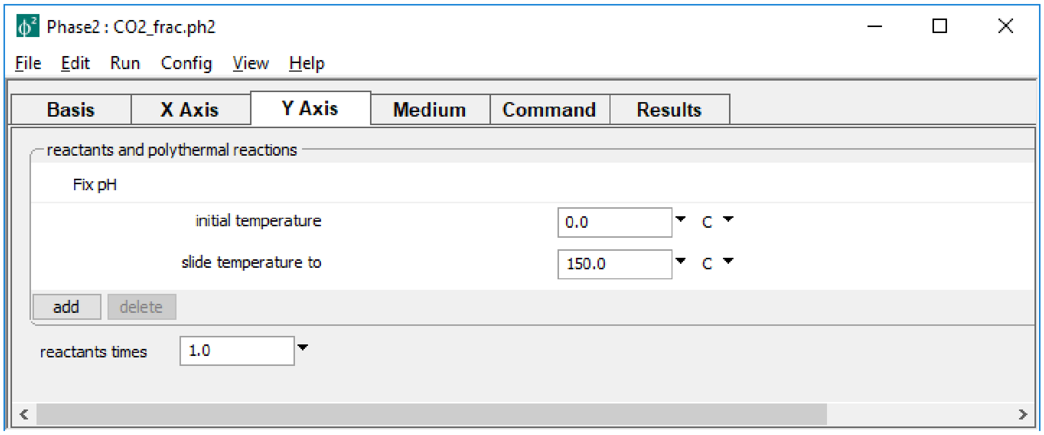
H2O	1.0	free kg	solvent
HCO3- ⇄	20.0	mmol/kg	
H+ ⇄	4.0	pH	
temperature	0.0	C	

+ advanced

□ time start 0.0 day end 1.0 day

Here, the lower-left corner of the diagram is a carbonate solution at pH 4 and 0 °C. Since we are not concerned with the background electrolyte, we disable charge balancing.

The **Y Axis** pane



reactants and polythermal reactions

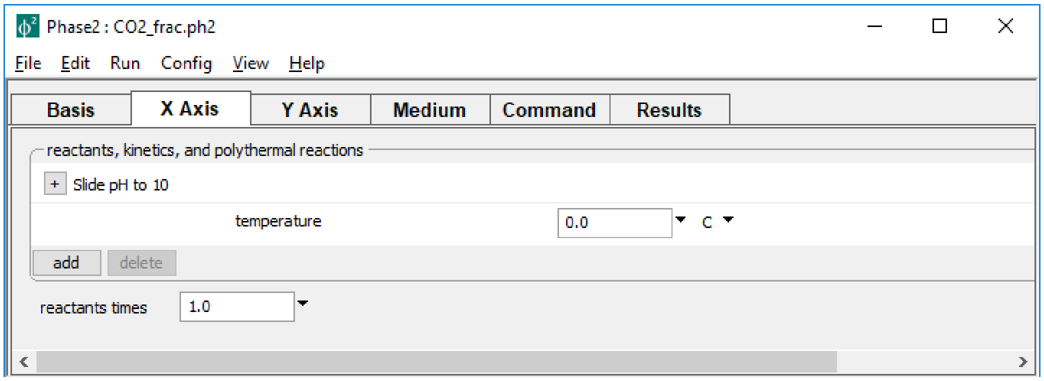
Fix pH

initial temperature	0.0	C
slide temperature to	150.0	C

reactants times 1.0

sets a polythermal staging axis in which temperature increases linearly to 150°C.

On the **X Axis** pane



reactants, kinetics, and polythermal reactions

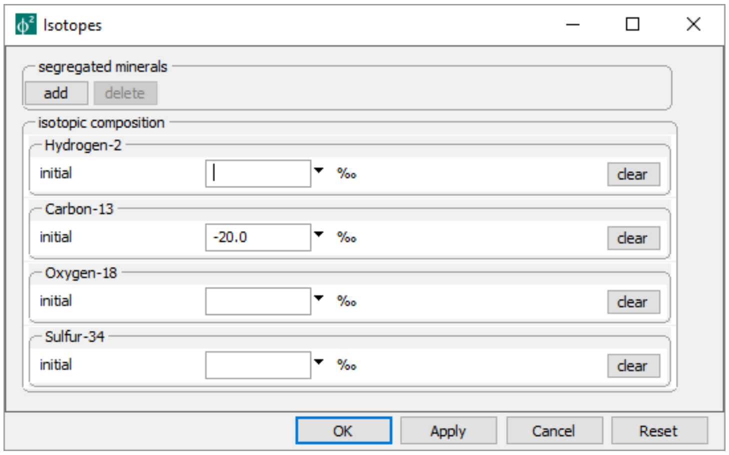
+ Slide pH to 10

temperature 0.0 C

reactants times 1.0

we slide pH to 10, while holding temperature constant.

On the **Config → Isotopes…** dialog



segregated minerals

isotopic composition

Hydrogen-2
initial ‰

Carbon-13
initial -20.0 ‰

Oxygen-18
initial ‰

Sulfur-34
initial ‰

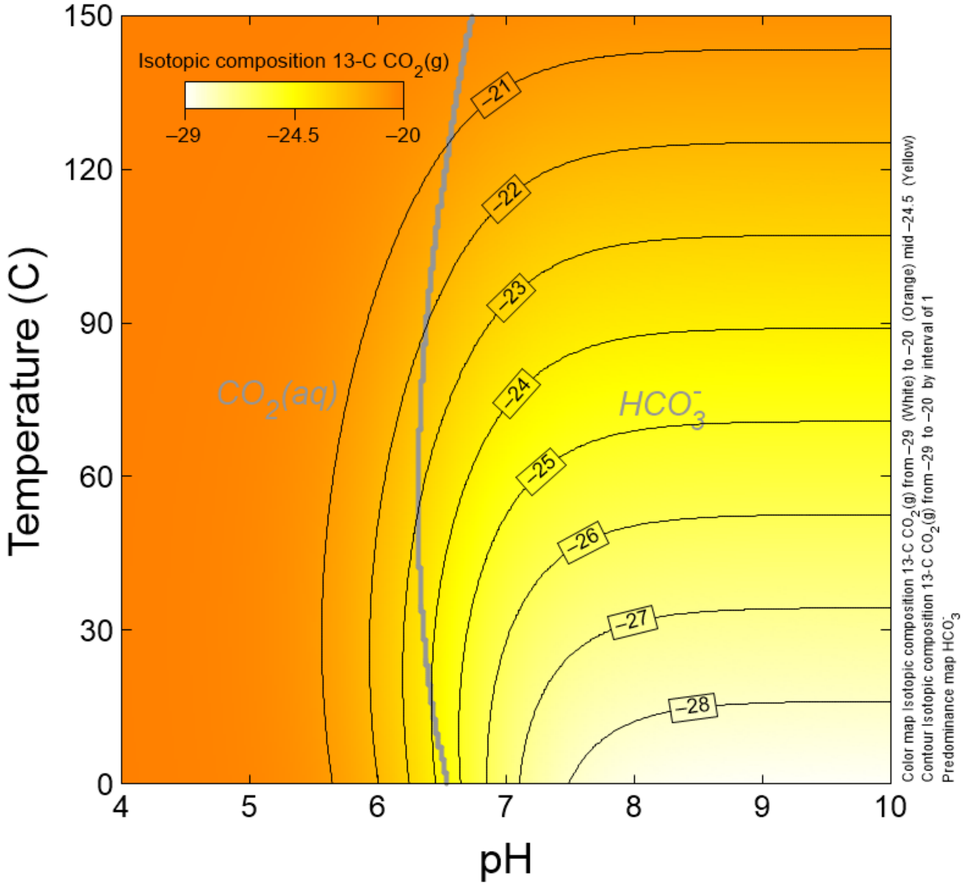
we set δ^{13}C of the fluid to –20 ‰ PDB. There is in the simulation no external source or sink of ^{13}C, and no mineral precipitates, so the fluid's bulk isotopic composition remains constant across the diagram.

Press the Run button on the **Results** pane to trace the calculation. Start **P2plot** by clicking on the Plot Results button. In **P2plot**, select **Plot → 2D Diagram**, then click on **Format → Axis Range and Variables…** to set the axis variables to "pH" along the *x* and "Temperature" along the *y* axis.

Next, click on **Format → Color Map…** and check the box on the top line. Next to "Variable type", choose "Isotopes", for "Subtype" choose "13-C", and then set "Variable to map" to "CO2(g)". Adjust the colors as you like and click **OK**. Now, open the **Format → Contour…** dialog, make the same settings for "Variable type", "Subtype", and "Variable to map", then set the appearance options to your liking, and again click **OK**. Finally, click

Format → Predominance Map… and check the box on the top line. Choose "HCO3-" for "Variable to map", then check "Don't color fields" and hit **OK**. The latter option displays predominance bounds and labels without coloring the fields, thus allowing the color map we've applied to remain visible.

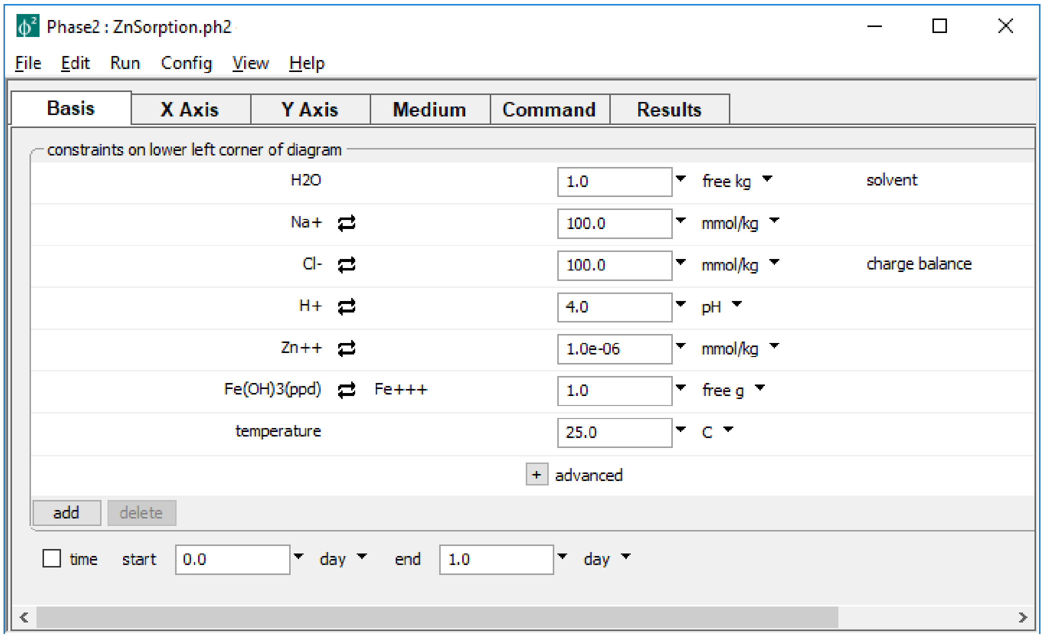
P2plot at this point should display a diagram similar to



As we see, the isotopic composition of CO_2(g) in equilibrium with the fluid varies with pH in response to the changing balance between CO_2(aq) and the carbonate species HCO_3^- and CO_3^{--}, and along the vertical axis as the isotopic fractionation factors shift with temperature.

7.12 Example: Surface complexation

Input file “ZnSorption.ph2” sets up calculation of a diagram showing the controls of pH and Zn^{++} content on the metal ion’s surface complexation with ferric hydroxide. Double-click the file to launch **Phase2** and move to the **Basis** pane



constraints on lower left corner of diagram

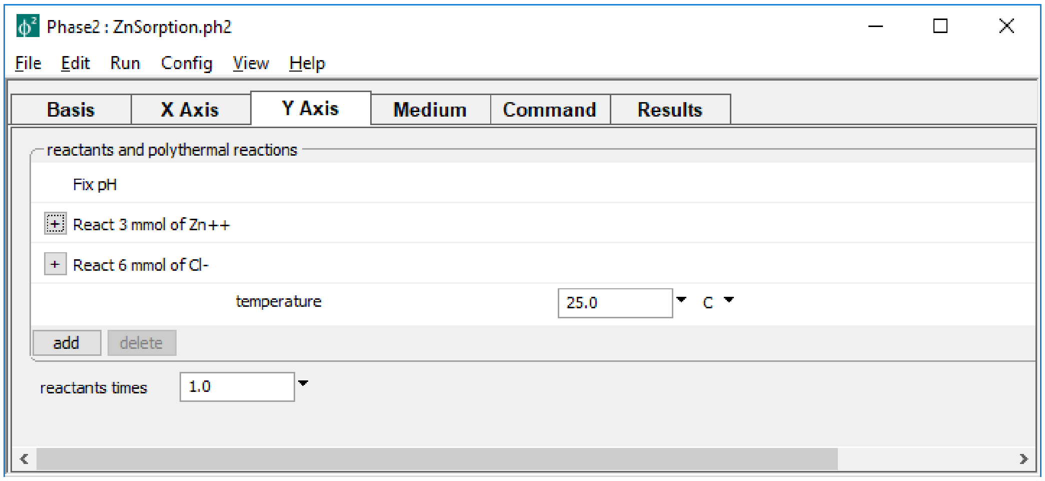
H2O		1.0	free kg	solvent
Na+ ⇄		100.0	mmol/kg	
Cl- ⇄		100.0	mmol/kg	charge balance
H+ ⇄		4.0	pH	
Zn++ ⇄		1.0e-06	mmol/kg	
Fe(OH)3(ppd) ⇄	Fe+++	1.0	free g	
temperature		25.0	C	

+ advanced

□ time start 0.0 day end 1.0 day

which sets at the bottom-left of the diagram a NaCl solution containing a small amount of Zn^{++}, in contract with ferric hydroxide. Note on the **Config → Redox couples…** dialog that the Fe^{+++}/Fe^{++} couple has been disabled, allowing us to carry ferric iron as a basis entry.

On the **Y Axis** pane



reactants and polythermal reactions

Fix pH

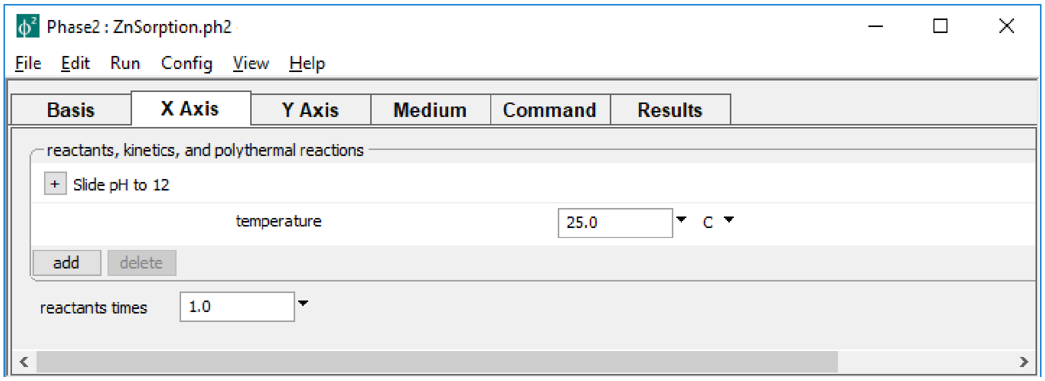
+ React 3 mmol of Zn++

+ React 6 mmol of Cl-

temperature 25.0 C

reactants times 1.0

Zn^{++} is titrated into the system while pH is held constant. The **X Axis** pane



reactants, kinetics, and polythermal reactions

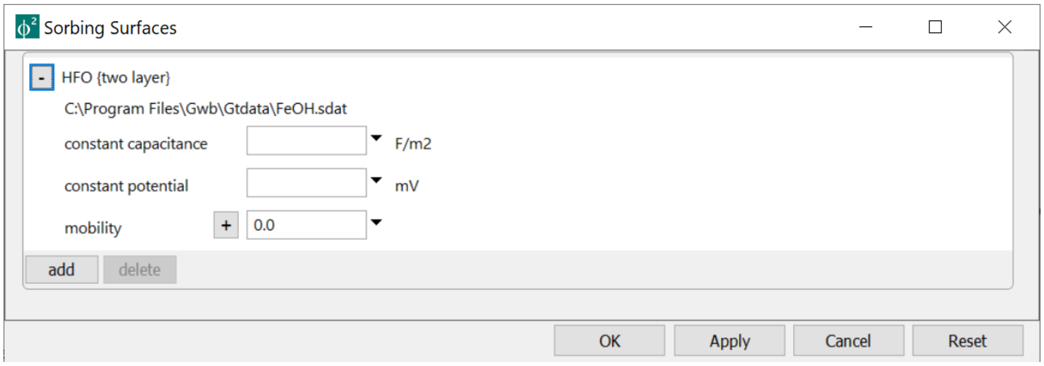
Slide pH to 12

temperature 25.0 C

reactants times 1.0

slides pH from its initial value of 4 at the left of the diagram, to 12 at the right.

On the **Config → Sorbing Surfaces…** dialog



HFO {two layer}

C:\Program Files\Gwb\Gtdata\FeOH.sdat

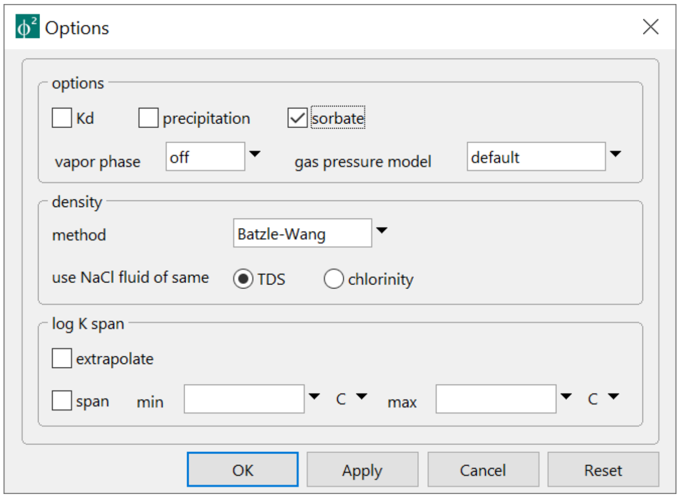
constant capacitance F/m2

constant potential mV

mobility 0.0

we see that the Dzombak and Morel dataset for ion complexation with hydrous ferric oxide is loaded.

On the **Config → Options…** dialog



Options

options

□ Kd □ precipitation ☑ sorbate

vapor phase off gas pressure model default

density

method Batzle-Wang

use NaCl fluid of same ● TDS ○ chlorinity

log K span

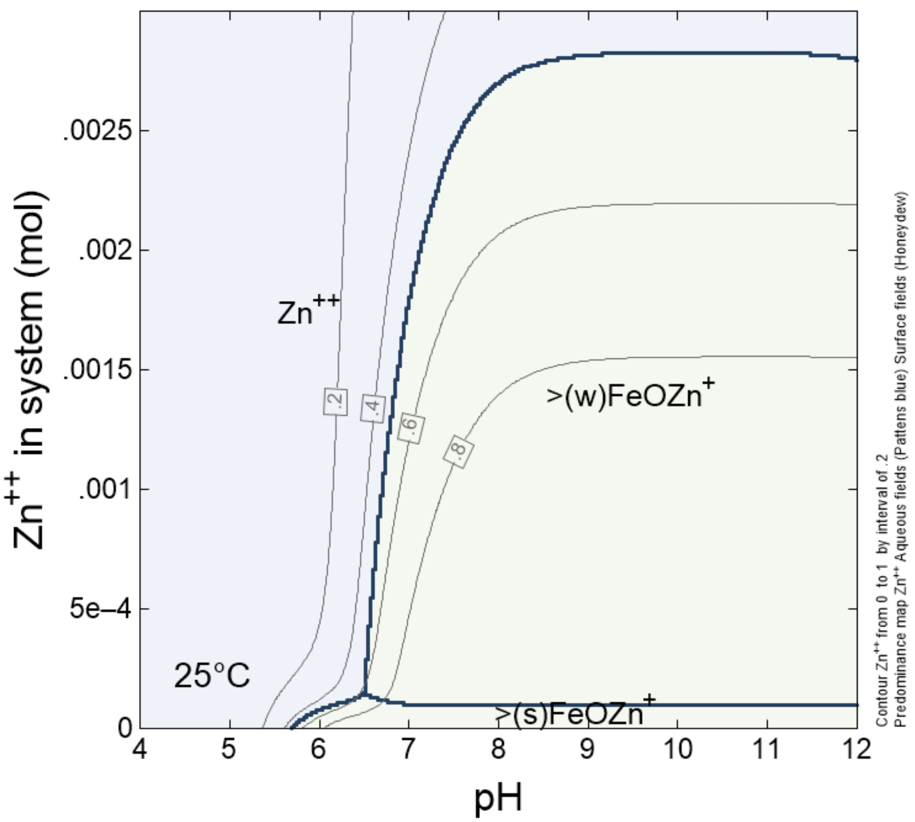
□ extrapolate

□ span min C max C

the "precipitation" option has been unchecked, preventing new minerals from forming during the simulation. Finally, the "sorbate" option here has been checked, specifying that the concentration of Zn^{++} set on the **Basis** pane represents the sum of the basis entry's dissolved and sorbed masses.

Press the Run button on the **Results** pane to trace the calculation, then start **P2plot** by clicking on the Plot Results button. In **P2plot**, select **Plot → 2D Diagram**. Then, on the **Format → Axis Range and Variables…** dialog, set "pH" on the *x* and "Zn+++ in system" on the *y* axis, and click **OK**. Select **Format → Predominance Map…** and choose "Zn++" for "Variable to map", then hit **Apply**. Go to **Edit → Contour…** , check the box at the top, and choose "Sorption" for "Variable type", "Fraction sorbed" for "Subtype", and "Zn++" for "Variable to contour". Make sure to select a "Linear" mapping, set the appearance and contour interval to your liking, and hit **Apply**.

The resulting diagram should look like

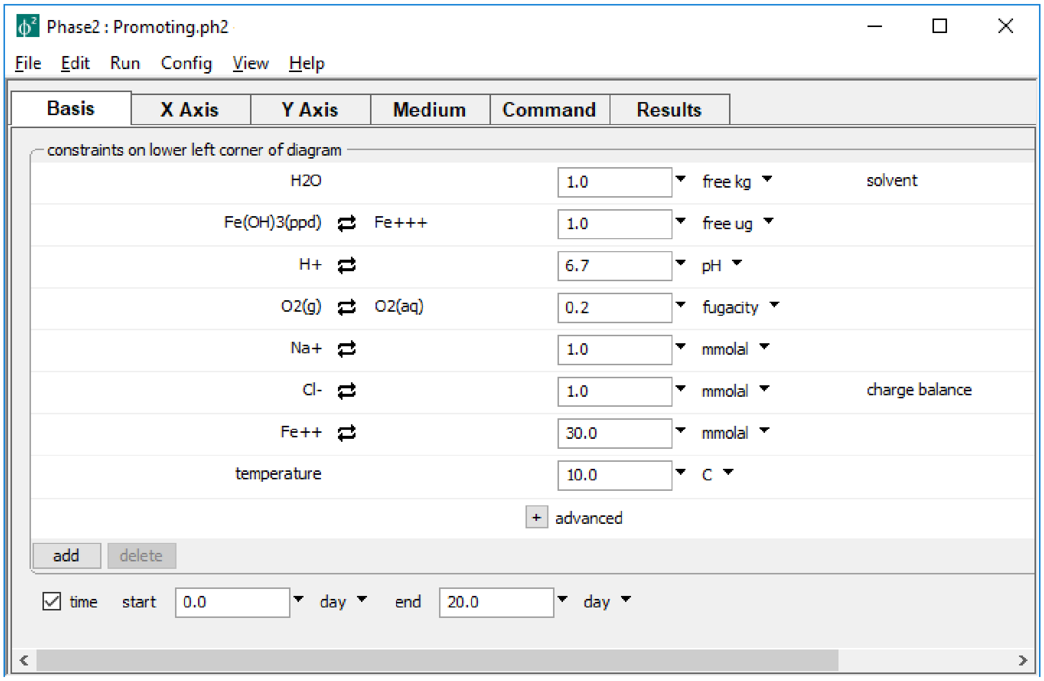


As is typical of a divalent metal, Zn^{++} does not sorb at low pH. Under neutral to alkaline conditions, the strong sites account for most of the sorbed Zn^{++} where the system contains little Zn^{++}. At higher zinc contents, however, the weak sites hold most of the zinc in the system, due to their greater abundance relative to the strong sites. Along the top of the diagram, the capacity of the ferric hydroxide to hold zinc is overwhelmed, and most of the metal is found in solution.

7.13 Example: Kinetic diagram

The input file “Promoting.ph2” builds on a **React** calculation borrowed from the section Kinetics of redox reactions in the Kinetic Reaction Paths chapter of this manual. In that example, contained in dataset “Redox.rea”, ferrous iron oxidizes to ferric hydroxide according to a kinetic rate law. In the rate law, OH^- promotes the oxidation reaction with a power of 2.

In this section, we construct a **Phase2** diagram showing how pH, which controls OH^- concentration, affects the oxidation rate. Opening “Promoting.ph2”, the **Basis** pane



constraints on lower left corner of diagram

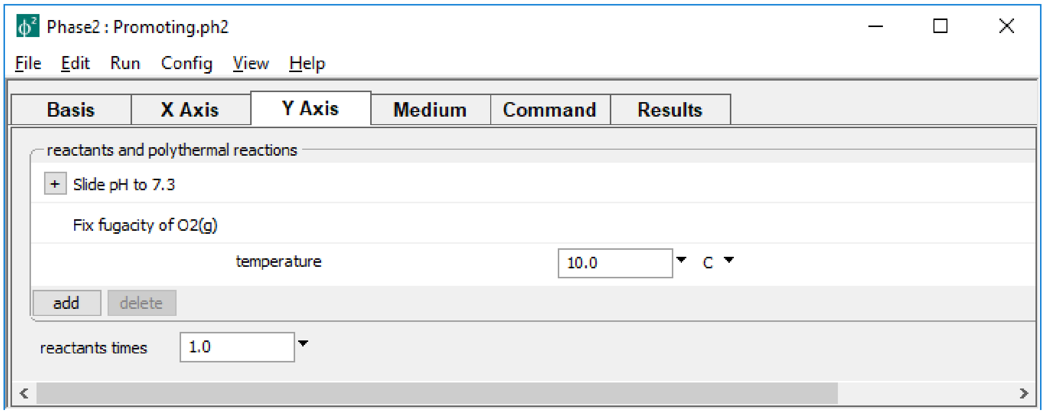
H2O		1.0	free kg	solvent
Fe(OH)3(ppd) ⇄	Fe+++	1.0	free ug	
H+ ⇄		6.7	pH	
O2(g) ⇄	O2(aq)	0.2	fugacity	
Na+ ⇄		1.0	mmolal	
Cl- ⇄		1.0	mmolal	charge balance
Fe++ ⇄		30.0	mmolal	
temperature		10.0	C	

+ advanced

☑ time start 0.0 day end 20.0 day

is much like that in the **React** example, except the pH is set slightly lower, to 6.7.

The **Y Axis** pane



reactants and polythermal reactions

+ Slide pH to 7.3

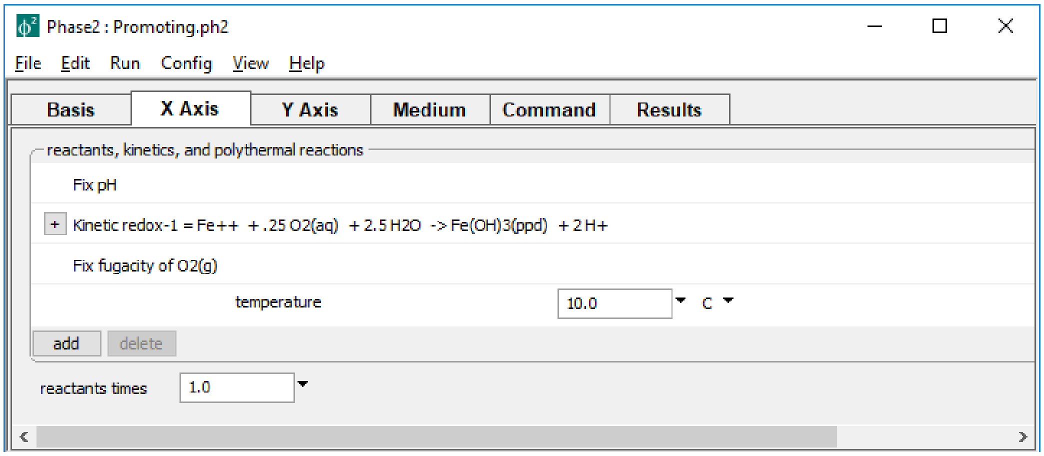
Fix fugacity of O2(g)

temperature 10.0 C

reactants times 1.0

sets up a staging path along which pH varies from its initial value at the bottom of the diagram to 7.3 at the top. The fluid is to remain in equilibrium with the atmosphere, so oxygen fugacity needs to be buffered here, as well as on the **X Axis** pane.

The settings on the **X Axis** pane



reactants, kinetics, and polythermal reactions

Fix pH

+ Kinetic redox-1 = Fe++ + .25 O2(aq) + 2.5 H2O -> Fe(OH)3(ppd) + 2 H+

Fix fugacity of O2(g)

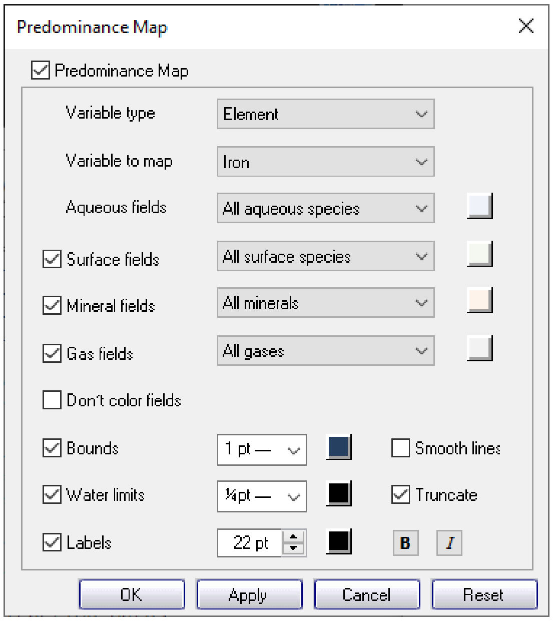
temperature 10.0 C

reactants times 1.0

fix pH according to the departure point on the staging path, define the kinetic rate law, and hold oxygen fugacity constant. The rate law entry, once expanded, is unchanged from the **React** example; notably, OH^- serves as a promoting species.

Press the Run button on the **Results** pane to trace the calculation, then start **P2plot** by clicking on the Plot Results button. Select **Plot → 2D Diagram**, set the *x* axis variable to “Time” in days, and the *y* axis to “pH”.

Open the **Format → Predominance Map…** dialog. Change the “Variable type” pulldown from “Basis species” to “Element”, select “Iron” as the “Variable to map”



Predominance Map

☑ Predominance Map

Variable type: Element

Variable to map: Iron

Aqueous fields: All aqueous species

☑ Surface fields: All surface species

☑ Mineral fields: All minerals

☑ Gas fields: All gases

☐ Don't color fields

☑ Bounds: 1 pt — ☐ Smooth lines

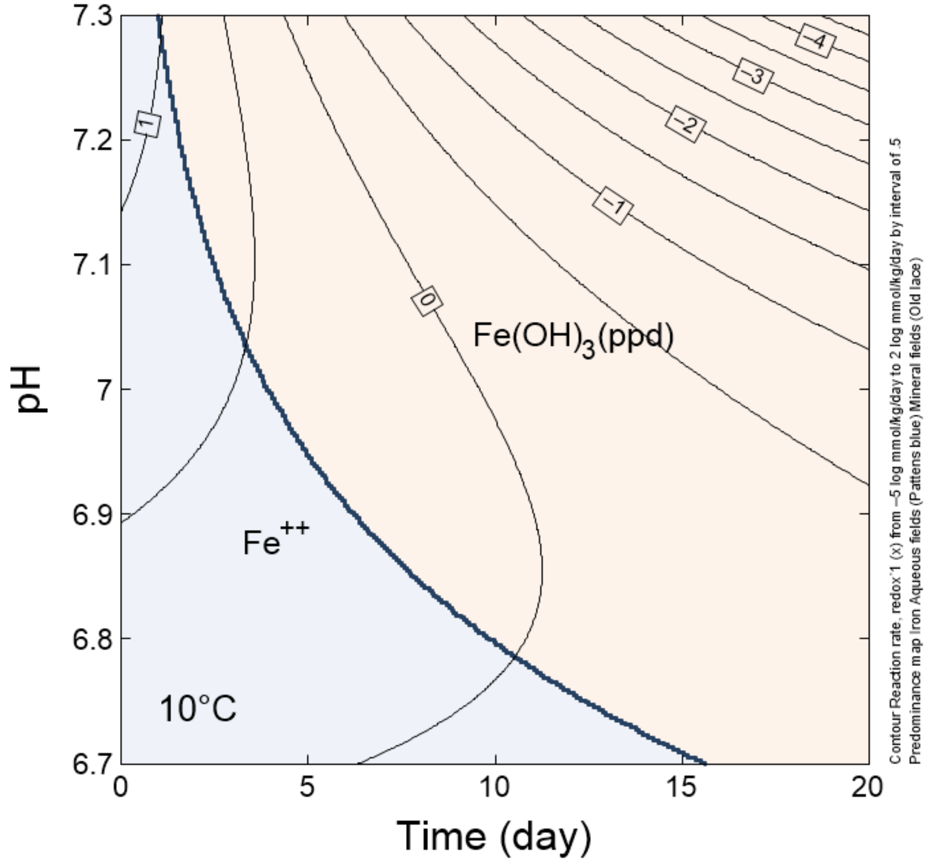
☑ Water limits: ¼pt — ☑ Truncate

☑ Labels: 22 pt

and hit **Apply**. Set in this way, the predominance diagram will include both ferrous and ferric iron-bearing species and mineral.

Open the **Format → Contour…** dialog and check the box at the top. Choose “Reaction rate” for “Variable type”, and for “Variable to contour, select “Reaction rate, redox-1 (x)”. Set “Variable units” to “mmol/kg/day” and, for “Type of plot”, set “Log”. Choose appearance options and a contour increment to your liking, and press **Apply**.

The diagram should look something like



As reflected by the contours of log reaction rate, reaction proceeds faster as pH increases, reflecting the role of the hydroxyl ion as a promoting species in the kinetic rate law. Along *x*, the reaction slows with time as another promoting species, the Fe^{++} ion, is depleted over time.

7.14 Cross-section plots

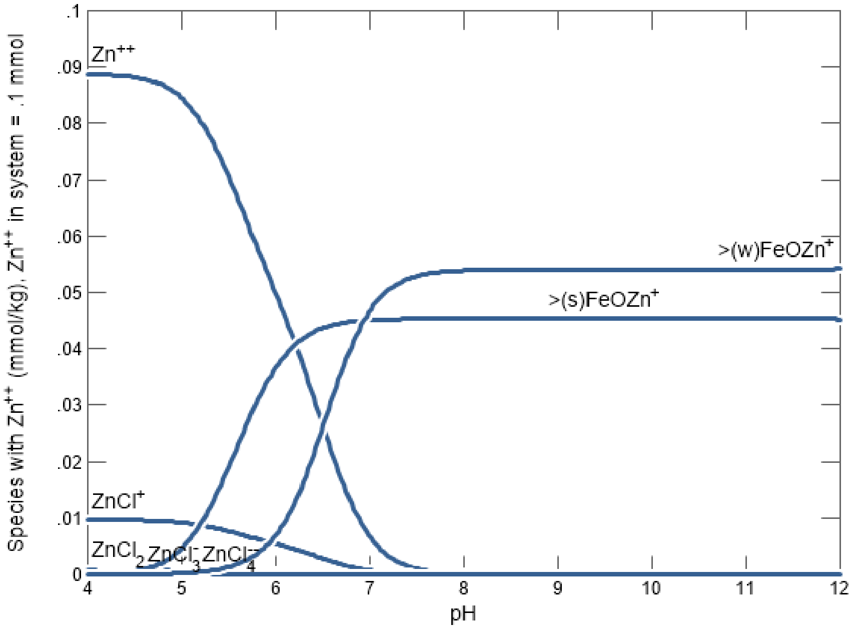
P2plot can diagram cross-sections through any diagram. You makes *x-y* plots of a broad range of variables such as pH, species concentration, mineral mass, gas pressure, isotopic composition, and so on.

To plot along a cross-section, select **Plot → Cross-section Plot…** to expose the configuration dialog. On the **X Axis** pane, choose the section's orientation by setting "Orientation" to either "Along scanning paths" or "Across scanning paths". On the **Y Axis** pane, if the orientation is along the scans, set "Display" to "Several values at one step along y" or "One value at several steps along y". Plotting across the scans, the choices both read "… along x".

Now, on the **X Axis** pane, select a category from the "Variable type" dropdown and choose the *x*-axis variable from the list. Similarly, set "Variable type" on the **Y Axis** pane. At this point, depending on the "Display" setting, you can choose either several variables on this pane and a single step on the **Step** axis, or one *y*-axis variable and several steps.

As an example, re-run file "ZnSorption.ph2", launch **P2plot**, and select **Plot → Cross-section Plot….** On the **X Axis** pane, set "Orientation" to "Along scanning paths" and choose to plot "pH" from the "System" "Variable type" and the "Chemistry" "Subtype"; under **Y Axis**, set "Variable type" to "Species", "Subtype" to "Any", "Filter" to "Zn++", "Display" to "Several values at one step along y", and then set a "linear" scale. Click on **Select all**, if necessary. Moving to the **Step** pane, set "Progress variable" to "Zn++ in system", set the unit to "mmol", and choose "Zn++ in system = .1 mmol".

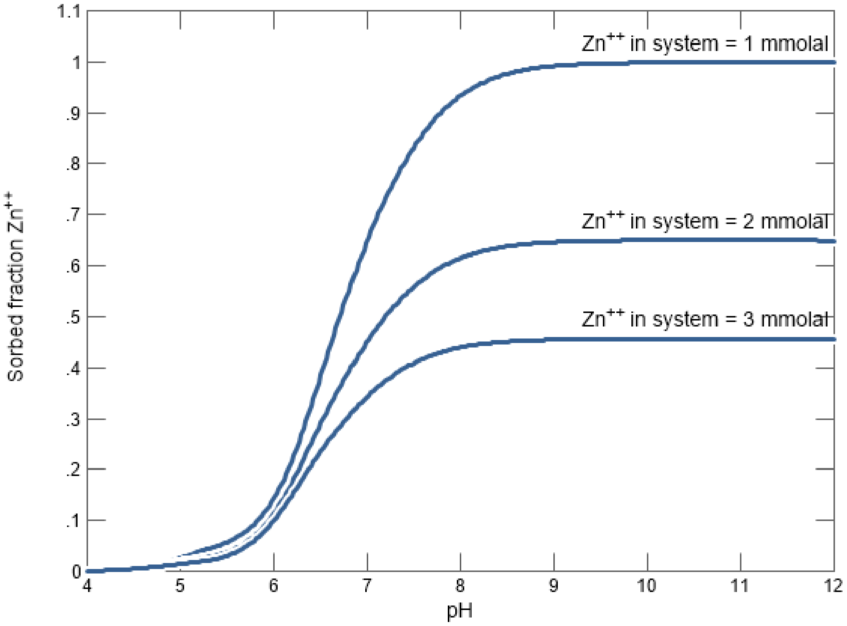
The resulting plot should look like



Here, the Zn^{++} ion is present in free form under acidic conditions, but complexed with strong and weak sites on the ferric surface at neutral to alkaline pH.

We can alternatively plot one variable at several points along the staging axis. Return to the **Y Axis** pane, set “Display” to “One value at several steps along y” and then “Variable type” to “Sorption” and “Subtype” to “Fraction sorbed”. On the **Step** pane, choose “Zn++ in system = 1 mmol”, hold down the Ctrl key while selecting the corresponding entries for “… 2 mmol” and “… 3 mmol”, and click **Apply**.

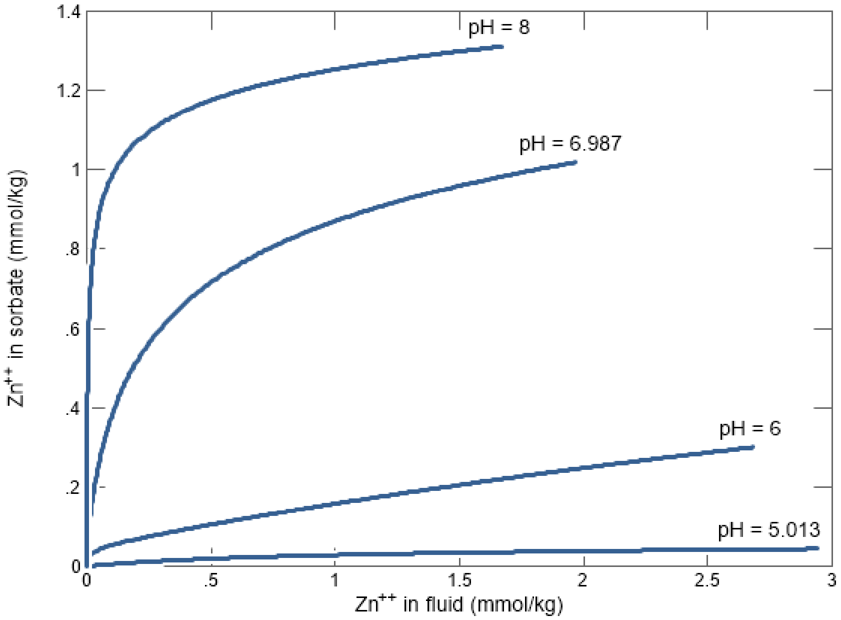
The resulting plot should look like



As more Zn^{++} is added to the system, a lesser fraction of it sorbs to the ferric hydroxide.

As a final example, we make a vertical section through the diagram. Returning to the **X Axis** pane, change “Variable type” to “Components in fluid”, set “Orientation” to “Across scanning paths”, and choose “Zn++” from the list. On the **Y Axis** pane, set “Variable type” to “Components in sorbate” and again choose “Zn++” from the list.

On the **Step** pane, set “Progress variable” to “pH” and choose the pH values closest to 5, 6, 7 and 8, then click on **Apply**. The resulting plot



shows the Zn^{++} sorption isotherms at the pH values of interest.

7.15 Phase2 command line

You can start **Phase2** by clicking the icon on the GWB dashboard, opening a "`.ph2`" file, or entering the command `phase2.exe` from the Windows "Command Prompt". When you start **Phase2** from the command line (as opposed to clicking on the icon), you can specify a number of arguments. For example, the command

```
phase2 -i my_script -d my_thermo.tdat
```

causes **Phase2** to read input commands from a file "my_script", and to use "my_thermo.tdat" as the thermodynamic database.

The following options are available from the command line:

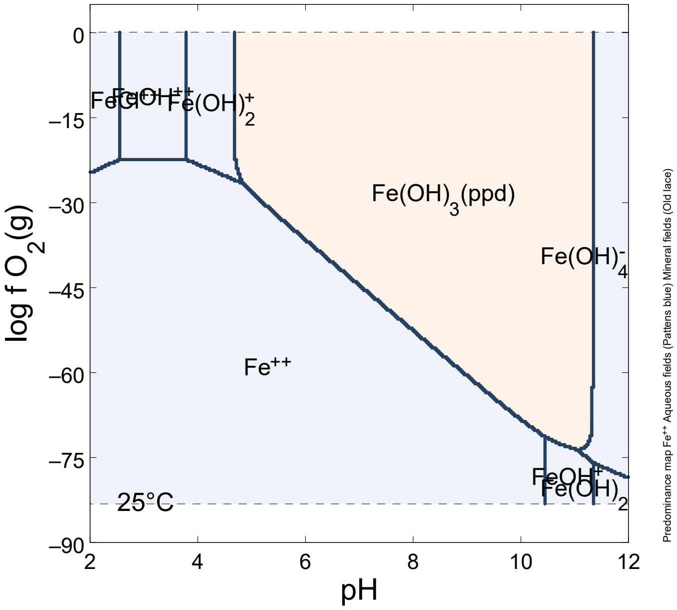
`-cd`	Change the working directory to the directory containing the input script specified with the –i option.
`-nocd`	Do not change the working directory.
`-i <input_script>`	Set a file from which to read input commands.
`-gtd <gtdata_dir>`	Set directory to search for thermodynamic datasets.
`-cond <cond_data>`	Set the dataset for calculating electrical conductivity.
`-d <thermo_data>`	Set the thermodynamic dataset.
`-iso <isotope_data>`	Set the dataset of isotope fractionation factors.
`-s <surface_data>`	Set a dataset of surface sorption reactions.

Using P2plot

P2plot is a graphics program that produces various types of diagrams showing the results of **Phase2** calculations. The program is similar to **Gtplot** and **Xtplot**, so if you have familiarized yourself with these programs, you already know much of what you need to know to run **P2plot**.

After running **Phase2**, start **P2plot** by clicking on **Plot Results** on the **Results** pane, or selecting **Run → P2plot** from **Phase2**'s menubar. The program will read the calculation results from the ".p2p" file **Phase2** produced and render them graphically. You can leave **P2plot** active when you run further **Phase2** simulations. Each time **Phase2** completes a run, it signals **P2plot**, which updates its display to reflect the new results.

You can also start **P2plot** by opening any ".p2p" or ".p2c" file, or by clicking on the **P2plot** icon on the Apps pane of the GWB dashboard. You can take input from a different dataset by dragging the dataset into the **P2plot** window, or choosing **File → Open → Data File…** from the menubar. Depending on the plot you choose, **P2plot** might look like



P2plot produces plots of various types. You can generate a two-dimensional rendering of the results, showing the predominant form of any original basis species or element, or a map of mineral assemblages. You can also show one or several variables represented by color shading or contours.

Alternatively, you can plot cross-sections through the diagram that show how certain variables change with reaction progress along staging or scanning paths. You can plot one or more variables from a single section, or one variable from several cross-sections.

Choose the diagram type from the **Plot** pulldown on the menubar. The program can render the following types of variables:

- Mass and volume of minerals and end members in the modeled system
- Concentration, activity, and activity coefficients of dissolved species
- Elemental composition of the fluid, minerals, sorbate, and the bulk system
- Composition of the fluid, minerals, sorbate, and the bulk system, expressed in terms of thermodynamic components
- Gas fugacity and partial pressure
- Saturation indices *Q/K* with respect to various minerals and solid solutions
- Bulk stable isotopic composition of the fluid, rock, sorbate, and the entire system, as well as the composition of individual species, gases, minerals, and end members
- Fraction of various components sorbed onto mineral surfaces
- Variables such as temperature, pH, and reacted mass

When **P2plot** starts for the first time in a given directory, it assumes a default configuration. Upon finishing, the program saves for its next run a dataset containing the current configuration of the plots. You can select options from the **File** menu to specify an alternative configuration (see Loading and saving plot configuration) or to reset the entire program configuration.

8.1 2D Diagram

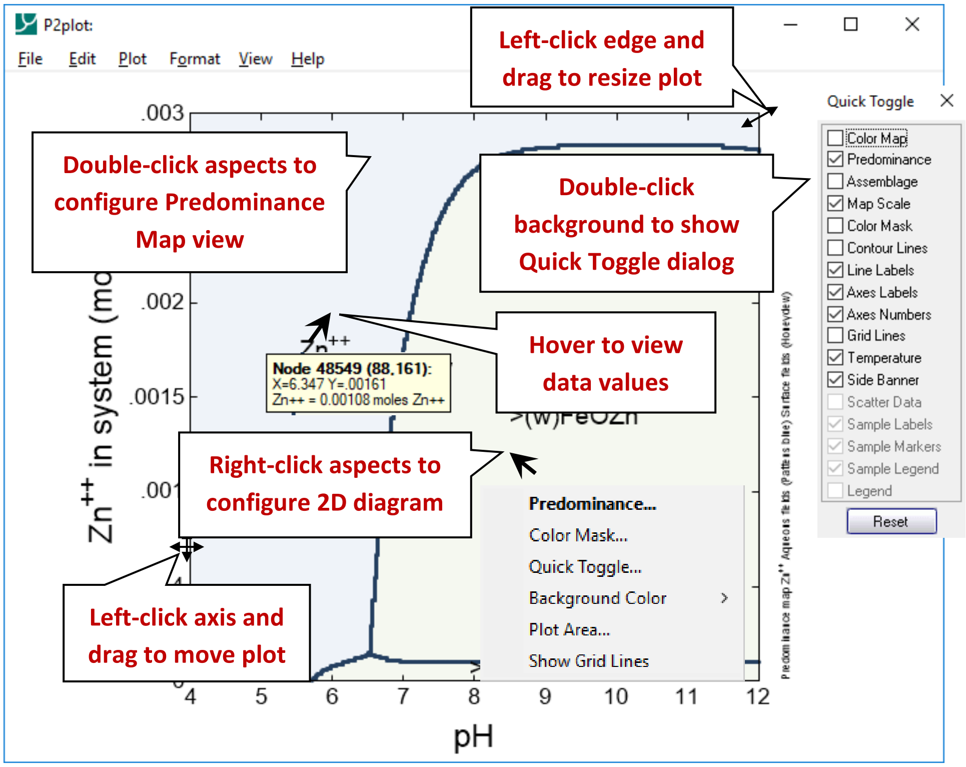
P2plot makes two-dimensional diagrams, in which variables are represented by color shading, color masking, and contour lines. You can plot species' fields of predominance, or mineral assemblages. For two-dimensional plots, the **Format** menu contains a number of configuration options

- **Quick Toggle…** to select the graphical elements of the plot to display
- **Color Map…** to define a shaded color map of a variable over the diagram axes
- **Color Mask…** to color mask of a variable's value range over the diagram axes
- **Contour…** to choose a variable to contour and configure the contouring
- **Predominance Map…** to plot the species accounting for the most mass of any original basis species or element

- **Assemblage Map…** to plot fields representing every mineral or combination of minerals over the diagram axes
- **Appearance…** to open a tabbed dialog containing the following:
    - **Axes and Ticks…** to define the appearance of the coordinate axes
    - **Plot Area…** to choose the dimensions, placement, and background color of the plot
    - **Title…** to define a title for the plot
    - **Font…** to control aspects of the text throughout the diagram
- **Axis Range and Variables…** to select the two variables appearing on the diagram's axes and adjust the range and tick increment
- **Side Banner…** to specify the appearance and placement of the informational banner along the side of the plot
- **Trace…** to overlay the trace of a reaction path calculated with **React**
- **GSS Data…** to affect the labeling of scatter data loaded from a GSS spreadsheet and the related legend

Alternatively, you can interactively modify many aspects of the plot, including lines, labels, axes, scales, and grid lines. A **right-click** on an aspect displays a menu showing the options available. A **double-click** on an aspect brings up a related dialog. **Left-click** on an aspect to select it for modification.

An aspect has been selected when its color changes, as in the case of lines, or when it is surrounded by manipulation handles, in the case of text, markers, and scales. Change the size of text, markers, scales, and the plot itself by dragging the sizing cursors. Adjust the placement of the plot by clicking within the plot or on either axis and dragging it. The plot title can be moved in the same way.


Left-click edge and drag to resize plot

Double-click aspects to configure Predominance Map view

Double-click background to show Quick Toggle dialog

Hover to view data values

Right-click aspects to configure 2D diagram

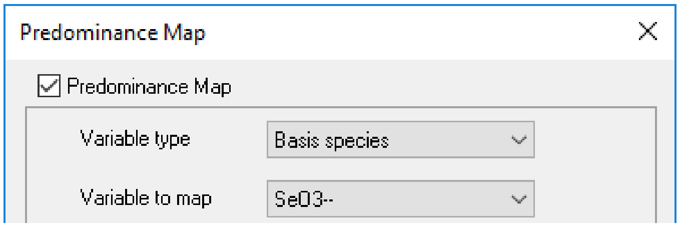
Left-click axis and drag to move plot

Node 48549 (88,161):
X=6.347 Y=.00161
Zn++ = 0.00108 moles Zn++

8.1.1 Predominance Maps

A predominance map shows the aqueous species, surface species, mineral, or end member that accounts for the greatest mass of any component or element. A gas predominates if its partial pressure exceeds the system pressure; gases in a vapor phase are not currently included in predominance diagrams.

Use the **Predominance Map...** dialog to control aspects of the map. First select the species you'd like to diagram. Sort by thermodynamic components by selecting "Basis species"

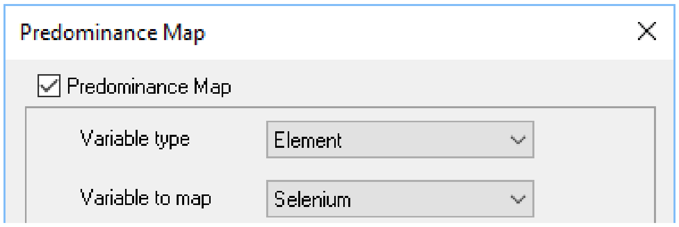

Predominance Map

☑ Predominance Map

Variable type: Basis species

Variable to map: SeO3--

or by elements by selecting “Element”



Predominance Map

☑ Predominance Map

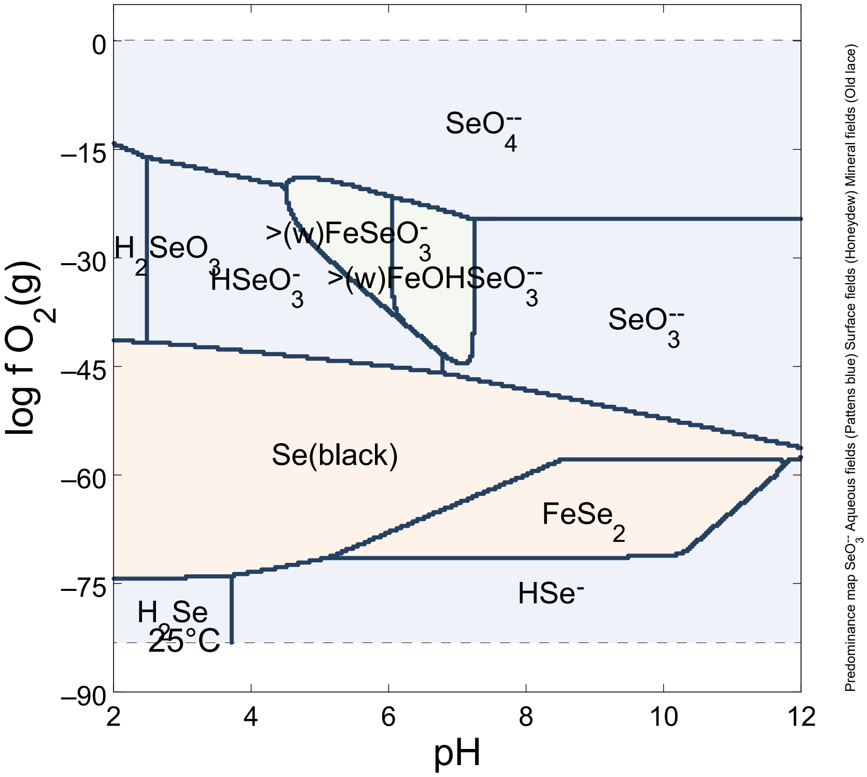
Variable type: Element

Variable to map: Selenium

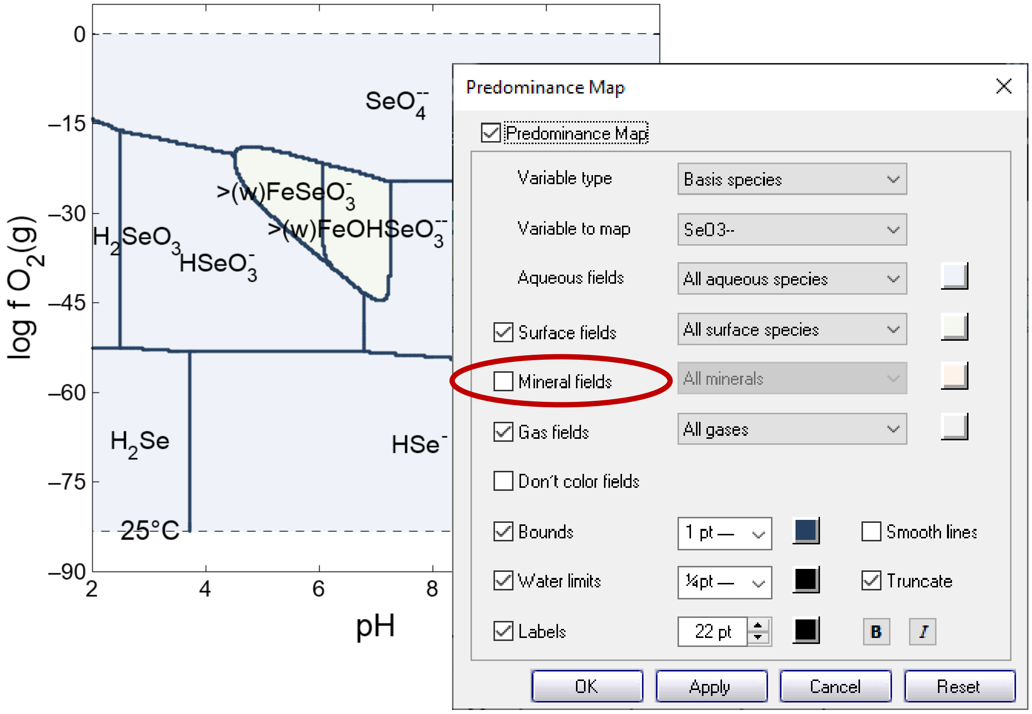
In many cases the resulting diagrams will be identical, but the latter option is especially useful for plotting decoupled redox species together.

You can hover on the plot to see the number of moles of the selected basis entry or element in the predominant species. For a gas, the partial pressure is reported.

By default all types of species are considered in the diagram, but you can hide any class by unchecking the appropriate box. In the example below



aqueous species, surface species, and minerals predominate at some point. Uncheck “Mineral Fields”



Predominance Map

☑ Predominance Map

Variable type: Basis species

Variable to map: SeO3--

Aqueous fields: All aqueous species

☑ Surface fields: All surface species

□ Mineral fields: All minerals

☑ Gas fields: All gases

□ Don't color fields

☑ Bounds: 1 pt — □ Smooth lines

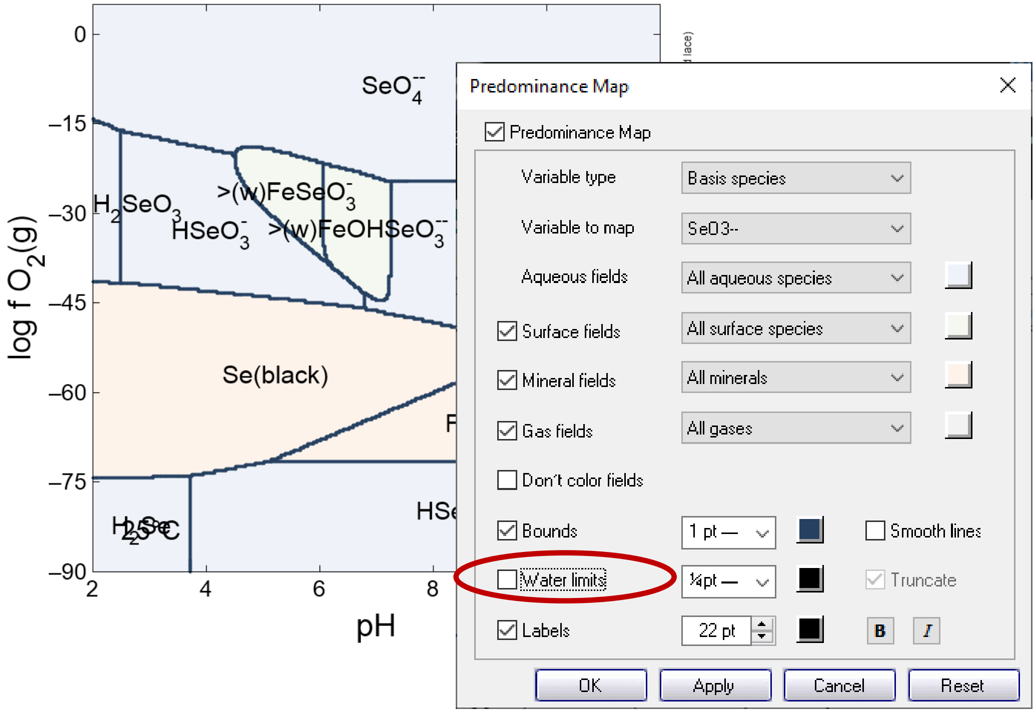
☑ Water limits: ¼pt — ☑ Truncate

☑ Labels: 22 pt **B** *I*

to ignore them in the tabulation of most abundant species.

Note, **P2plot** considers the predominance of species, not of phases. Thus, it does not account for the combined mass of basis species or elements from each end member in a solid solution phase when figuring predominance. In a calcite-rhodochrosite solid solution, for example, the phase contains bicarbonate from both end members, but each end member is treated separately in determining species predominance.

Predominance maps by default draw lines representing the stability region of water and do not display species outside this region. Water is treated as unstable anywhere the partial pressure of O_2(g) or H_2(g) exceeds the confining pressure. Uncheck "Water Limits" to ignore the stability limits. Or, with "Water Limits" checked, uncheck "Truncate" to draw outside the stability limits.



Predominance Map ✕

☑ Predominance Map

Variable type: Basis species

Variable to map: SeO3--

Aqueous fields: All aqueous species

☑ Surface fields: All surface species

☑ Mineral fields: All minerals

☑ Gas fields: All gases

☐ Don't color fields

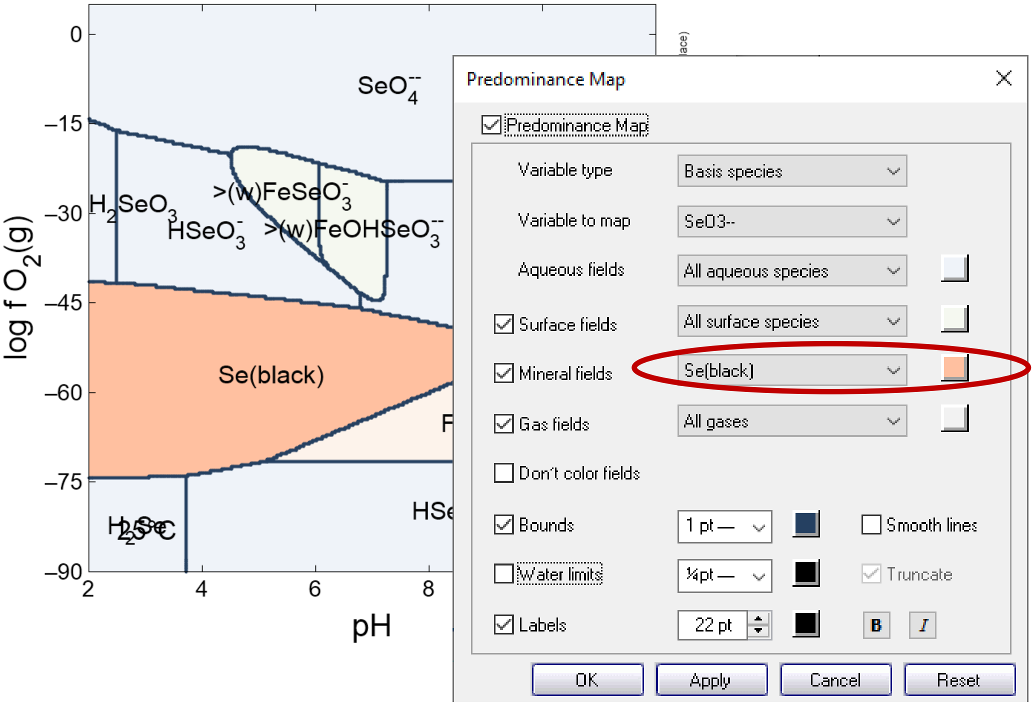
☑ Bounds 1 pt — ☐ Smooth lines

☐ Water limits ¼pt — ☑ Truncate

☑ Labels 22 pt **B** *I*

The Tables section of the thermo dataset shows the temperature expansion for pressure, either as a *T*-table or polynomial coefficients. The **Thermo Datasets** chapter in the ***GWB Reference Manual*** describes thermo datasets and the ***GWB Essentials Guide*** includes a description of the **TEdit** app used to view and modify the datsets.

Each type of field (Aqueous, Surface, Mineral, Gas) is colored differently by default. All minerals and end members, for example, are colored a light orange ("Old lace"). Click on the colored square next to a species type (e.g. All minerals) to change the color for that field. To set the color of an individual species, select it from the pulldown, then pick a color. To change the color of elemental selenium, for example, change the pulldown next to "Mineral Fields" from "All minerals" to "Se(black)", then click on the color box to pick a color.



☑ Predominance Map

Variable type: Basis species

Variable to map: SeO3--

Aqueous fields: All aqueous species

☑ Surface fields: All surface species

☑ Mineral fields: Se(black)

☑ Gas fields: All gases

□ Don't color fields

☑ Bounds: 1 pt — □ Smooth lines

□ Water limits: ¼pt — ☑ Truncate

☑ Labels: 22 pt B I

The $FeSe_2$ keeps its original color, but the Se(black) field has been updated.

You can select "Don't color fields" to draw predominance bounds and labels without coloring the fields. In this way, you can overlay the bounds on another map type, such as a color map or assemblage map.

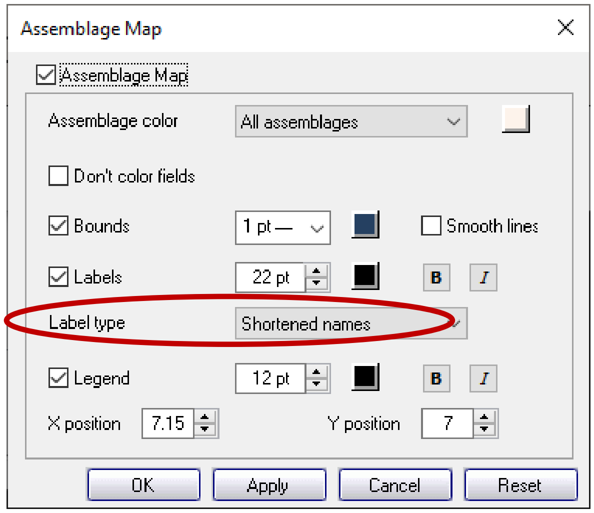
You can additionally change the color and thickness of lines separating species, as well as the water stability limits, and the size, color, and style of labels.

8.1.2 Assemblage Maps

Assemblage maps show fields in which a mineral or group of minerals exist in the calculation. You can hover over any field to see the assemblage of minerals present. Where no minerals form, the hover text displays "No minerals". Note, a pair of solid solution end members represents a single phase.

Use the **Format → Assemblage Map…** dialog to control aspects of the diagram. All assemblages plot in the same color by default, but you can color assemblages individually, in the same way described above for predominance fields.

Depending on the number of unique mineral assemblages present in a calculation, and the number of minerals defining each assemblage, assemblage maps can become quite crowded. Use the “Label type” pulldown to control whether full names, shortened names, or a number representing each assemblage labels each field. You can remove the labels entirely by unchecking the “Labels” box. A legend, which you can toggle on or off with the “Legend” box, lists each assemblage present and displays the field color, abbreviations, and/or index for each assemblage. In the example below



Assemblage Map

☑ Assemblage Map

Assemblage color: All assemblages

□ Don't color fields

☑ Bounds: 1 pt — □ Smooth lines

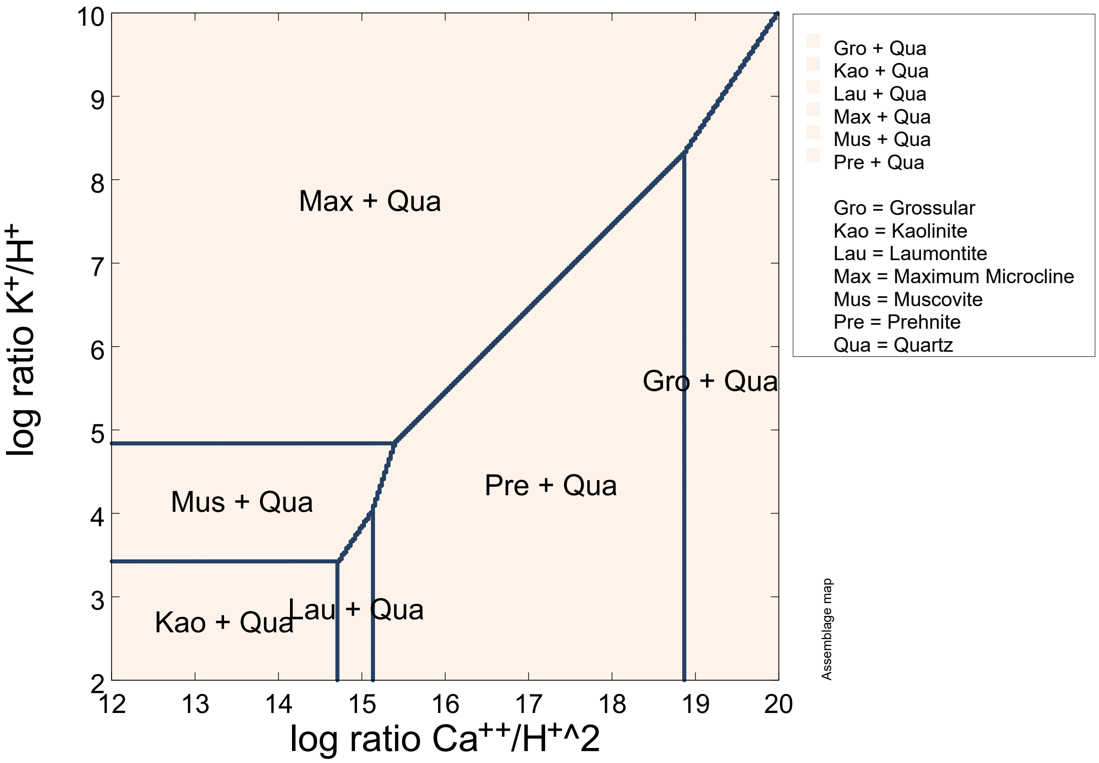
☑ Labels: 22 pt

Label type: Shortened names

☑ Legend: 12 pt

X position: 7.15 Y position: 7

the “Shortened names” option is chosen to simplify the assemblage labels. The legend lists each assemblage as it appears on the plot and defines each abbreviation.

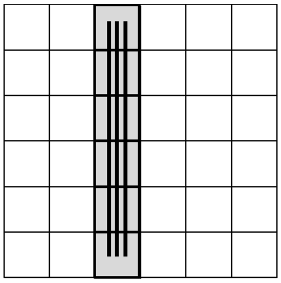


8.1.3 Axis Variables

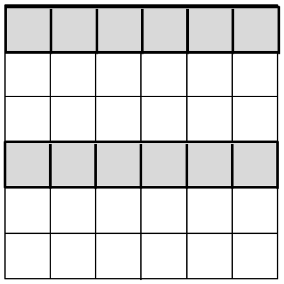
A 2D diagram's axes by default are labeled in terms of reaction progress, which has meaning only in terms of how the calculation is set up. Use the **Format → Axis Range and Variables…** dialog to control the variables appearing on the *x* and *y* axes, their units where applicable, and to set custom ranges and tick increments. For more information, see Progress variables in this guide.

8.2 Cross-section Plot configuration

You can plot one or more horizontal or vertical cross-sections through the calculated 2D diagram. Horizontal sections represent progress along the scanning paths, while vertical sections represent progress across the scanning paths.



Several values across a scanning path



One variable along several scanning paths

Choose **Plot → Cross-section Plot…** to configure a cross-section plot. You can alter the configuration by interacting with the graphical aspects of the plot, such as any data line, data label, axis, tick mark, label, or grid line. A **right-click** on an aspect of the plot displays a menu showing the options available. **Double-click** on an aspect to invoke the **Cross-section Plot Configuration** dialog. Dragging an inner or outermost axis tick mark, number, or grid line will shift or scale, respectively, the axis range, (see **Using Gtplot** in the ***GWB Essentials Guide***).

The **X Axis** tab lets you specify whether a horizontal or vertical cross-section is shown. The default *x*-axis setting is reaction progress in the *x* direction (a horizontal section). Use the **Orientation** pulldown to control the orientation of the cross-section.

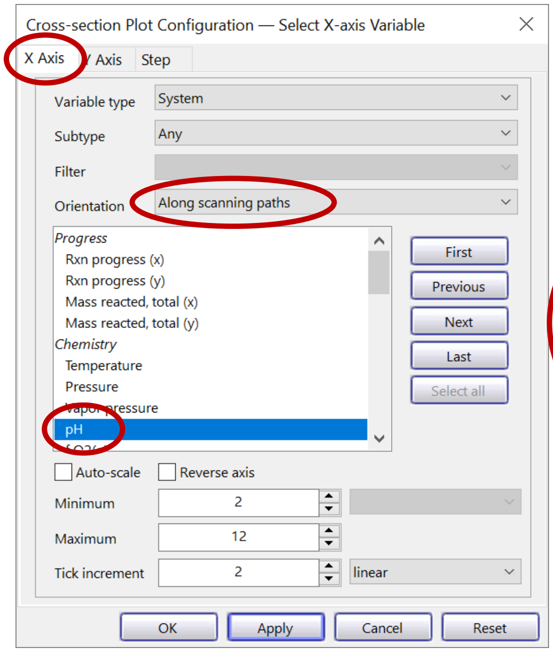
The **X Axis** tab also lets you select the variable to appear on the *x* axis. Most commonly you would choose the variable or variables representing reaction progress (see **Progress variables** in this guide), but a cross-section plot, like an XY plot in **Gtplot**, can plot any variable on the *x* axis. For this reason, progress variables are not listed separately from the larger set of variables. Use **Variable type** and **Subtype** to control the type of variables to appear along the *x* axis. Variable type options are:

- **System**: Results describing progress (e.g. Rxn progress, Time elapsed), chemistry (e.g. Temperature, Pressure, and pH), the fluid (e.g. density), mass and volume (e.g. Solution mass and mineral mass), and so on
- **Reactants**: Net mass reacted, buffered values, kinetic rate law terms, and so on
- **Reaction rate**: Simple reactants, fixed and sliding buffers, and kinetic reactants
- **Species**: Concentration of chemical species; volume of minerals, end members, and lost gases
- **Components**: Bulk composition of the system, fluid, minerals, sorbate, vapor, stagnant zone, and mobile colloids, in terms of thermodynamic components
- **Elements**: Bulk composition, in terms of elements
- **Activity**: Aqueous species, end members, surfaces species; gas fugacity
- **Activity coefficients**: Aqueous species, end members, and surface species; fug. coefficients

- **Saturation**: *Q/K* of the fluid with respect to minerals and solid solutions
- **Nernst potentials**: Theoretical redox potentials for independent redox couples
- **Solid solutions**: Amount and composition of solid solutions
- **Surfaces**: Charge density, potential, sorbing surface area, and so on
- **Sorption**: Sorbed fraction and Apparent Kd's of sorbing components
- **Gases**: Partial pressure and other properties of gas species; dissolved gas concentrations
- **Isotopes**: Composition of the bulk system, fluid, rock, sorbate, and vapor, as well as individual species, for active isotope systems

The selection of a variable type displays a list of possible variables. Click on a variable you wish to select. Use the following to specify the way the variables are plotted:

- **Variable type** to control the type of variables to appear along the plot's axis
- **Subtype** to list subsets of the chosen variable type
- **Filter** to list, for species or mineral saturation plots, the set of just those that contain a given basis species
- **Auto-scale** to set the data range for the axis to span the data to be plotted
- **Minimum**, **Maximum**, and **Tick increment** to set the data range for the axis
- **Reverse axis** to reverse the sense of the axis
- **Units** to select alternative units, if any, for the axis
- **As** to plot values as elemental equivalents, or as protonated or deprotonated species equivalents
- **Type** to set the axis to a linear, log, or delta scale. A delta scale shows change in a variable's value from the initial point in any particular step
- **First**, **Previous**, **Next**, and **Last** to cycle through the variable choices


Cross-section Plot Configuration — Select X-axis Variable

X Axis | Y Axis | Step

Variable type: System

Subtype: Any

Filter:

Orientation: Along scanning paths

Progress
- Rxn progress (x)
- Rxn progress (y)
- Mass reacted, total (x)
- Mass reacted, total (y)

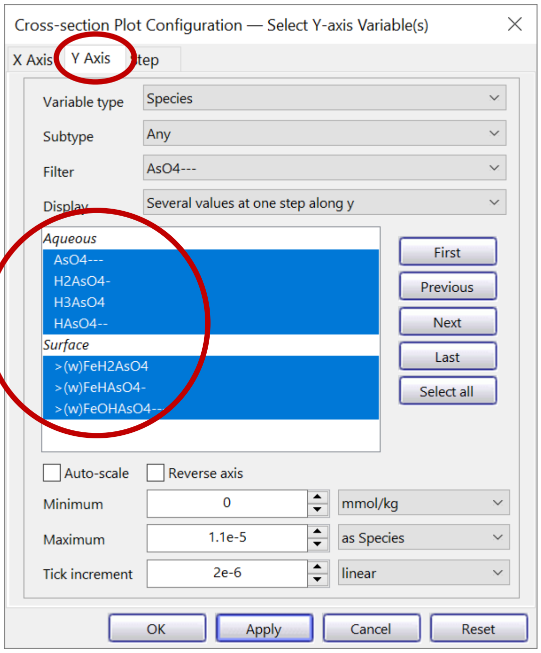
Chemistry
- Temperature
- Pressure
- Vapor pressure
- pH

□ Auto-scale □ Reverse axis

Minimum: 2

Maximum: 12

Tick increment: 2 linear


Cross-section Plot Configuration — Select Y-axis Variable(s)

X Axis | Y Axis | Step

Variable type: Species

Subtype: Any

Filter: AsO4---

Display: Several values at one step along y

Aqueous
- AsO4---
- H2AsO4-
- H3AsO4
- HAsO4--

Surface
- >(w)FeH2AsO4
- >(w)FeHAsO4-
- >(w)FeOHAsO4--

□ Auto-scale □ Reverse axis

Minimum: 0 mmol/kg

Maximum: 1.1e-5 as Species

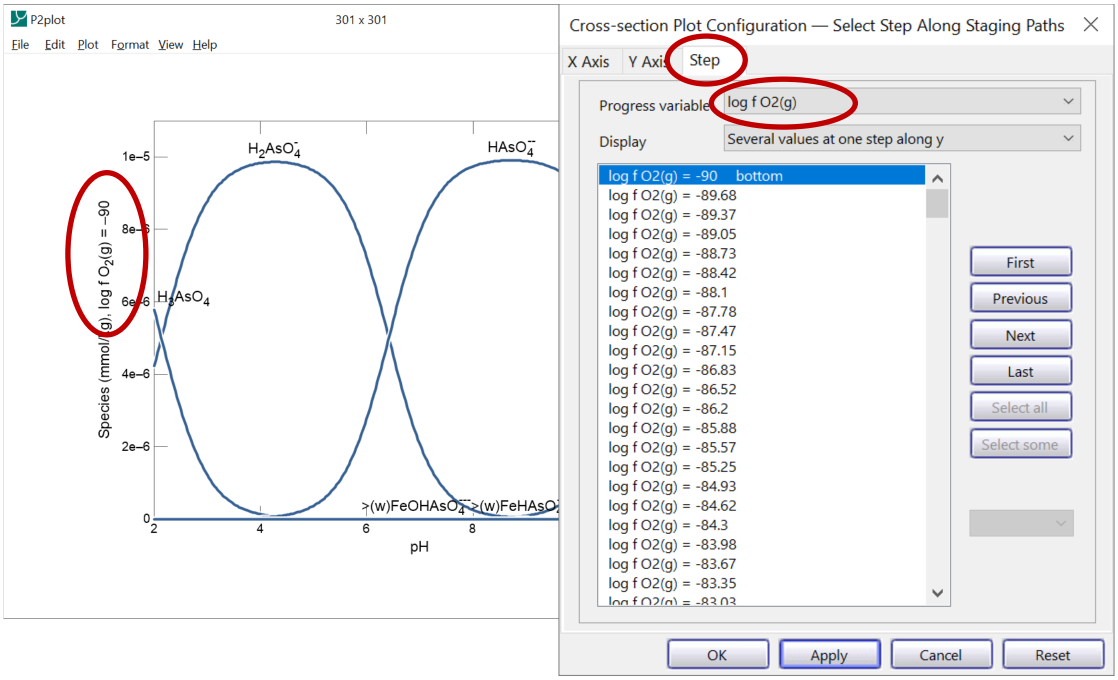
Tick increment: 2e-6 linear

The **Y Axis** tab lets you select the variables to plot on the *y* axis and specify the way they are plotted. Choose a **Variable type**, **Subtype**, and variable just as you do on the *x* axis. Use the **Filter** menu to set, for species or mineral saturation plots, a basis species to consider. In a typical calculation, so many dissolved species are considered, and saturation states are determined for so many minerals, that the data fit poorly on a single plot. For this reason, **P2plot** can group species and minerals into sets of just those that contain a given basis species. You can choose to plot, for example, the species containing Na$^+$, or the saturation indices of minerals containing Al^{+++}. Alternatively, you may choose to consider all species and minerals.

The selection of a variable type displays a list of possible variables. Click on a variable you wish to select. Use **Ctrl+click** to select multiple variables. Use **Shift+click** to select a range of variables.

The **Step** tab lets you select the position in the diagram for which data should be rendered. If you are plotting a horizontal section (a scanning path), the step indicates reaction progress along the vertical direction (the staging path). If you are plotting a vertical section (the staging path, or parallel sections), the step indicates progress along the horizontal direction (the scanning path).

Depending on how the calculation is configured, one or more descriptive progress variables may be available. Use the **Progress variable** pulldown to choose from the available options. The progress variable aids in step selection and is used to label cross-section plots.



Progress variable: log f O2(g)

Display: Several values at one step along y

log f O2(g) = -90 bottom
log f O2(g) = -89.68
log f O2(g) = -89.37
log f O2(g) = -89.05
log f O2(g) = -88.73
log f O2(g) = -88.42
log f O2(g) = -88.1
log f O2(g) = -87.78
log f O2(g) = -87.47
log f O2(g) = -87.15
log f O2(g) = -86.83
log f O2(g) = -86.52
log f O2(g) = -86.2
log f O2(g) = -85.88
log f O2(g) = -85.57
log f O2(g) = -85.25
log f O2(g) = -84.93
log f O2(g) = -84.62
log f O2(g) = -84.3
log f O2(g) = -83.98
log f O2(g) = -83.67
log f O2(g) = -83.35
log f O2(g) = -83.03

Use the **First**, **Previous**, **Next**, and **Last** buttons to cycle the plot through the steps.

You can choose from the **Y Axis** or **Step** tab to display several values from a single step, or one value from several steps.

8.3 Plot types

Use the **Plot** menu to choose between the 2D Diagram View and the Cross-section Plots. Each selection invokes a dialog box specific to the type of plot in question (see **Using Gtplot** in the ***GWB Essentials Guide***).

8.4 Editing plot appearance

P2plot allows you to interactively modify many aspects of the diagram. The details of using active items, aspect menus, and dialogs to edit the plot appearance are given in the **Using Gtplot** section of the ***GWB Essentials Guide***.

8.5 Reaction traces

P2plot can overlay onto a 2D diagram the trace of a reaction path (a “.gtp file”) calculated with the **React** program.

To read a .gtp file into **P2plot**, select **File → Open → Reaction Trace…**. Clear a trace from a diagram on the **Open → Reaction Trace…** dialog, with the OFF button. The **Reaction Traces** section of the **Using Act2** chapter in the ***GWB Essentials Guide*** gives details on plotting traces of **React** simulations. Unlike in **Act2** and **Tact**, you can currently load only one trace into **P2plot**.

8.5 Scatter data

P2plot can overlay the data in a **GSS** data sheet (a .gss file) as "scatter data" on a plot of the results of a **Phase2** calculation. The program can add scatter data to any of the 2D diagrams it makes, as well as the cross-section plots. To coordinate the plotting of sample dates and times in the **GSS** data sheet on a cross-section plot, set an explicit starting date and time in **Phase2**.

To read a .gss file into **P2plot**, select **File → Open → Scatter Data…**. Clear scatter data from a diagram on the **Open → Scatter Data…** dialog, with the **OFF** button. The **Scatter data** section of the **Using Gtplot** chapter in the ***GWB Essentials Guide*** gives details on plotting scatter data from .gss files.

In GWB releases 7.0 and earlier, the program took scatter data from a specially formatted text file, rather than a .gss data sheet. Legacy scatter files are still supported, and are described in the **Scatter Data** chapter of the ***GWB Reference Manual***.

8.6 Loading and saving plot configuration

Upon finishing, **P2plot** writes into the user's working directory a file, "p2plot_conf.p2c", containing the configuration of the current plots. When the program starts again in the same directory, it reads the file and assumes the same configurations.

Choosing **File → Reset Configuration** or the `reset` option from the command line (see P2plot command line) returns the configuration for each plot type to its default state.

You can also save plot settings in ".p2c" configuration files. To do so, select **File → Save As…**. You can then specify that file as the configuration for a later **P2plot** run from the command line (the `-c` flag; see P2plot command line) or read it into **P2plot** by selecting **File → Open → Configuration…**.

Exiting the program by choosing **File → Abort (No Save)** causes an immediate exit from the program; the plot configuration is lost.

8.7 Exporting the plot

P2plot makes it convenient to use the plots you create in articles, reports, presentations, and databases. You can copy the current plot to the clipboard and then paste it into a variety of applications, in a format meaningful to the application.

To copy a plot, use **Edit → Copy** or **Ctrl+C**. If you paste the plot into MS PowerPoint, it will appear as an EMF (an MS Enhanced Metafile) graphic object. Pasting into Adobe Illustrator places a native AI graphic.

If you paste the plot into MS Excel or a text editor, such as Notepad or MS Word, the numerical values of the data points that make up the lines on the plot will appear in spreadsheet format.

You can control the format in which the plot is copied to the clipboard by selecting **Edit → Copy As**. You can choose to copy the plot as an AI object, an EMF object, or a bitmap, or to copy the data points in the plot as tab delimited or space delimited text. Use the tab delimited option to paste the data into a spreadsheet program like MS Excel. For examining the data in a text file created with an editor like Notepad or MS Word, copying from **P2plot** using the space delimited option and then pasting into the editor writes a nicely aligned table.

8.8 P2plot command line

To run **P2plot**, click on the **P2plot** icon on the GWB dashboard, or open a ".p2p" or ".p2c" file. The program can also be initiated from the Windows command prompt by typing `p2plot.exe`. Starting the program in this way allows you to make use of the command line arguments described below.

P2plot accepts a number of arguments from the command line. For example, the command

```
p2plot -i Phase2_plot1.p2p -c Config1.p2c
```

causes **P2plot** to read as input the file "Phase2_plot1.p2p", and to use the plot configuration stored in "Config1.p2c".

The following options are available from the command line:

`-i <input_data>`	Set the "Phase2_plot.p2p" dataset, produced by a Phase2 run, which contains the data to be plotted. The program, by default, looks for file Phase2_plot.p2p in the user's working directory.
`-c <config_file>`	Set the configuration file to be read at startup. By default, the program reads the file p2plot_conf.p2c if it exists in the working directory.
`-trace <trace_file>`	Set the "React_plot.gtp" dataset, produced by a React run, to plot as a reaction trace. By default, P2plot does not plot reaction traces.

`-scat <scatter_file>`	Take scatter data from the named dataset. By default, P2plot does not plot scatter data.
`-graph <plot_type>`	Set the type of plot displayed at startup. By default, it is set from the configuration file. Choose from: 2d and cross-section.
`-reset`	Set the default configuration at startup; do not read p2plot_conf.p2c.

Appendix: Further Reading

The following literature references, from the many hundreds that have been published, provide a starting point for further reading on various aspects of geochemical modeling and its applications. Additional references are available in the ***GWB Essentials Guide*** and the ***GWB Reactive Transport Modeling Guide***.

A.1 Reaction modeling

Bethke, C.M., 2022, *Geochemical and Biogeochemical Reaction Modeling*, 3rd ed. Cambridge University Press, New York, 520 p.

Garrels, R.M. and F.T. MacKenzie, 1967, Origin of the chemical compositions of some springs and lakes. In Equilibrium Concepts in Natural Waters. *American Chemical Society, Advances in Chemistry Series* **67**, 222–242.

Helgeson, H.C., 1968, Evaluation of irreversible reactions in geochemical processes involving minerals and aqueous solutions—I.Thermodynamic relations. *Geochimica et Cosmochimica Acta* **32**, 853–877.

Helgeson, H.C., R.M. Garrels and F.T. Mackenzie, 1969, Evaluation of irreversible reactions in geochemical processes involving minerals and aqueous solutions—II. Applications. *Geochimica et Cosmochimica Acta* **33**, 455–481.

Helgeson, H.C., T.H. Brown, A. Nigrini and T.A. Jones, 1970, Calculation of mass transfer in geochemical processes involving aqueous solutions. *Geochimica et Cosmochimica Acta* **34**, 569–592.

Parkhurst, D.L., D.C. Thorstenson and L.N. Plummer, 1980, PHREEQE—A computer program for geochemical calculations. *U.S. Geological Survey Water-Resources Investigations* 80–96, 210 p.

Plummer, L.N., D.L. Parkhurst and D.C. Thorstenson, 1983, The development of reaction models for groundwater systems. *Geochimica et Cosmochimica Acta* **47**, 665–686.

Plummer, L.N., D.L. Parkhurst, G.W. Fleming and S.A. Dunkle, 1988, PHRQPITZ—A computer program incorporating Pitzer's equations for calculation of geochemical reactions in brines. *U.S. Geological Survey Water-Resources Investigations Report* 88–4153, 310 p.

Reed, M.H., 1982, Calculation of multicomponent chemical equilibria and reaction processes in systems involving minerals, gases and an aqueous phase. *Geochimica et Cosmochimica Acta* **46**, 513–528.

Reed, M. and N. Spycher, 1984, Calculation of pH and mineral equilibria in hydrothermal waters with application to geothermometry and studies of boiling and dilution. *Geochimica et Cosmochimica Acta* **48**, 1479–1492.

Wolery, T.J., 1979, Calculation of chemical equilibrium between aqueous solution and minerals: the EQ3/6 software package. *Lawrence Livermore National Laboratory Report* UCRL52658, 41 p.

Wolery, T.J., 1983, EQ3NR, a computer program for geochemical aqueous speciation-solubility calculations: user's guide and documentation. *Lawrence Livermore National Laboratory Report* UCRL53414, 191 p.

A.2 Geochemical kinetics

Lasaga, A.C., 1984, Chemical kinetics of water-rock interactions. *Journal of Geophysical Research* **89**, 4009–4025.

Lasaga, A.C. and R.J. Kirkpatrick (eds.), 1981, *Kinetics of Geochemical Processes*. Mineralogical Society of America, Washington, 398 p.

Jin, Q. and C.M. Bethke, 2002, Kinetics of electron transfer through the respiratory chain. *Biophysical Journal* **83**, 1797–1808.

Jin, Q. and C.M. Bethke, 2003, A new rate law describing microbial respiration. *Applied and Environmental Microbiology* **69**, 2340–2348.

Jin, Q. and C.M. Bethke, 2005, Predicting the rate of microbial respiration in geochemical environments. *Geochimica et Cosmochimica Acta* **69**, 1133–1143.

Jin, Q. and C.M. Bethke, 2007, The thermodynamics and kinetics of microbial metabolism. *American Journal of Science* **307**, 643–677.

A.3 Boiling and condensation

Affandi, M., N. Mamat, S. N. A. M. Kanafiah, and N. S. Khalid, 2013, Simplified Equations for Saturated Steam Properties for Simulation Purpose. *Procedia Engineering* **53**, 722–726.

Wagner, W. and H. J. Kretzschmar, *International Steam tables*, 2ed., Springer-Verlag: Berlin, 2008

A.4 Isotopic modeling

Bowers, T.S., 1989, Stable isotope signatures of water-rock interaction in midocean ridge hydrothermal systems: sulfur, oxygen, and hydrogen. *Journal of Geophysical Research* **94**, 5775–5786.

Bowers, T.S. and H.P. Taylor, Jr., 1985, An integrated chemical and stable isotope model of the origin of midocean ridge hot spring systems. *Journal of Geophysical Research* **90**, 12583–12606.

A.5 Compilers and interpreters

Aho, A.V., R. Sethi and J.D. Ullman, 1986, *Compilers, Principles, Techniques, and Tools*. Addison Wesley, Reading, Massachusetts, 796 p.

Donnelly, C. and R. Stallman, 1995, *Bison, The YACC-compatible Parser Generator*. Free Software Foundation, Boston, 89 p.