

## **Appendix B: Interface and IO**

The ACCRETE batch geochemistry code is at present used in two different settings: (1) In stand-alone batch simulations to study the temporal evolution of single grid volumes (as in paper III); and (2) as a batch code that is used as an external object in reactive transport simulations (as in paper V). Both types of simulations employ a C++ interface which calls ACCRETE as an external object. The principal difference between the two simulation types is how the interface C++ code works. The Appendix presents some details about how simulations are solved by ACCRETE/ATHENA.

## ACCRETE used in stand-alone batch simulations

The ACCRETE batch version was used in this work to simulate CO<sub>2</sub>-water-rock interactions in paper III. The C++ interface works in such simulations as a control center doing three tasks in sequential order: (1) Reading information from an input file; (2) running the simulation according to the directives given in the input file, i.e. number of timesteps, sizes of timesteps etc.; and (3) writing the results to an output file.

Below is the `glauconite.dat` input file listed which defines the glauconite sand system. The information listed after `//` is usually not listed but is included here to explain the content of the file. This is an example file and should not be confused with the input used for the glauconitic sand in paper III.

```
SIM_PAR (T,P,V)
327 // Temperature (K)
260 // Pressure (bar)
100000.00 // Volume (m^3)
10000 // Number of timesteps
1 // Size of timesteps

MIN_DATA (volume fractions)
MIN_STAT (1: primary or secondary; 0: not used in sim)
0.013 // Volume fraction calcite
1 // Reaction ON
0.005 // Volume fraction magnesite
0 // Reaction OFF
0.01 // Siderite
1 // Reaction ON
0.0 // Dawsonite
1
0.0 // Albite
1
0.02 // Microcline
1
0.72 // Quartz
1
0.0 // Chalcedony
1
0.02 // Kaolinite
1
0.0 // Clinocllore
1
0.0 // Daphnite
1
0.0 // Muscovite
1
0.04 // Phlogopite
1
0.04 // Annite
1
0.01 // Labradorite
1
0.0 // Gibbsite
1

SOL_DATA (mole fractions)
3e-9 // Mole fraction H+
1.33e-4 // xCa2+
0.0 // xCO2,aq
```

```

4.14e-5 // xHCO3-
8.07e-3 // xNa+
9.30e-3 // xCl-
2.32e-10 // xAl3+
0.0 // xCO32-
3.24e-4 // xMg2+
3e-9 // xOH-
2.97e-4 // xSiO2,aq
9.50e-5 // xK+
0.0 // xFe2+
0.0 // NOT DEFINED
0.982 // xH2O

SOL_DATA2 (CO2 volume fraction of pore space)
0.2 // Volume fraction (VCO2/V $\phi$ ) of CO2

```

After stand-alone batch simulations finish, results that are temporarily stored in buffers during the simulation are written to an output file. The output file contains information about the system ( $T$ ,  $P$ ,  $V$ , time, etc) in addition to temporal evolution of the aqueous phase (mole fractions listed), minerals (volume fractions listed), the volume fraction of CO<sub>2</sub> ( $\text{vol}_{\text{CO}_2}/\text{vol}_{\phi}$ ), and porosity.

## ACCRETE used as an external object in reactive transport simulations

The interface code that is used when ACCRETE works as an external module for the ATHENA flow simulator is much simpler than for stand-alone batch simulations. In reactive flow simulations the interface defines a linkage directive to call the ACCRETE F77 object, and calls the geochemistry solver as an external function. Data required for the simulations, i.e. system definitions, are sent back and forth between the interface and ATHENA.

### *Linking of the ACCRETE code to the ATHENA simulator*

To call the ACCRETE F77 as an external function from the C++ interface a linkage directive is necessary. This linkage directive is added to the external functions prototype in the start of the interface. The 'extern' keyword declares a variable or function and specifies that it has external linkage. The 'extern' keyword used with a string 'C' means that a linkage convention of another language is being used for the declarator. Below is the linkage directive and function-call used in the interface for simulations in papers III and V listed.

```

\\ --- Linkage directive ---
extern "C" {void accrete_(double *, double *,
                        double [], int *, int *, double
                        [], double *, double *, double *,
                        double *, int *, double *, double
                        *, double [], double *, double*,
                        double *, double [], int []);}

/* --- Body of interface code preparing information from
ATHENA to be sent to ACCRETE --- */

```

```

\\ --- Call external fortran object ---
      accrete_(&P, &T, nw, &miner, &tau, nm, &gridVol,
              &por, &nCO2, &volH2O, &nloop, &denw, &denCO2,
              xw, &dDenWdP, &KH, &dDenCO2dP, xmin, xminOn);

```

### *Information sent back and forth between the interface and ACCRETE*

The present version of the ACCRETE routine uses 19 input parameters that are updated during a run. 11 of these parameters are required for the ACCRETE to work properly, whereas the remaining 8 is updated and can be utilized by external programs. Examples of the latter are phase densities and pressure derivatives of the phase densities that can prove important for flow simulations. The parameters are listed below with an explanation following the ‘\’ symbol.

```

double P           \\ *Pressure (bar).
double T           \\ *Temperature (K).
double nw[15]      \\ *Mole vector of aqueous components.
int  miner         \\ *Mineral reactions On/Off [1/0].
int  tau           \\ *Size timestep (seconds).
double nm[16]      \\ *Mole vector of minerals.
double gridVol     \\ *Size of grid volume (m^3).
double por         \\ *Porosity (1 - Xmin).
double nCO2        \\ *Moles of supercritical or gaseous
                  CO2.
double volH2O      \\ **Volume of aqueous phase (m^3).
int  nloop         \\ *Iteration number.
double denw        \\ *Density of aqueous phase.
double denCO2      \\ *Density of CO2 phase.
double xw[15]      \\ *Aqueous phase mole fractions.
double dDenWdP     \\ *Pressure differential for aqueous
                  density.
double KH          \\ *Henrys law coefficient for CO2.
double dDenCO2dP   \\ *Pressure differential for CO2
                  density.
double xmin[16]    \\ *Mineral volume fractions.
int  xminOn[16]    \\ *Individual mineralreactions
                  On/Off [1/0].

```

\*required for a simulation. Data not required are calculated and updated in ACCRETE and can be used by an external program.

\*\*A positive value is required at input. This can be any positive value since the ACCRETE code updates the volume according to the density and mole fractions in the aqueous phase. An updated value is sent back.

### *Compiling and linking*

The ACCRETE code and the interface are compiled and linked by the following commands:

```

>f77 -c -w ACCRETE.f
>g++ -c interface.cpp
>g++ -o accrete.exe accrete.o interface.o -lg2c

```

The executables run in this study (papers III and V) are compiled and linked by GNU Fortran and C++ compiler version GCC 3.2.2 under Linux 3.2.2-5.

## Log files

Two log-files are at present used for informing users of possible errors: (1) `error.log`; and (2) `nwneg.log`. The former is always created if not present in the program folder, and information is appended whenever some errors occur. The latter is formed if any calculations produce negative moles of the water components. Possible contents of the files and explanations are listed below:

`error.log`:

```
"Volume CO2 larger than porespace"
```

The error occurs if volume of porespace calculated from  $V_{\phi} = V_{tot} \left( 1 - \sum_i x_i \right)$ , where  $V_{tot}$  is total volume of system and  $x_i$  denotes volume fraction of mineral specie  $i$ , is smaller than the volume of  $\text{CO}_2$  that is calculated from the moles and density of the  $\text{CO}_2$  phase. If this error occurs the volume fraction of the aqueous phase is set to zero and the program returns without any further mineral or speciation calculations.

```
"Not charge neutral. Sumcharge = #"
```

This error occurs after the speciation calculations if the sum of charges defined as  $\eta = \sum_i m_i z_i$  is larger than some predefined limit. The current version has a limit set to  $10^{-8}$ . The program continues even if the sum of charges is higher than the predefined limit.

```
"Temperature outside temperature range <#min, #max>"
```

and

```
"Pressure outside pressure range <#min, #max>"
```

These two error messages are written if the simulation has predefined a pressure or temperature that is outside the limit of the code. The program terminates if this error occur. The present limits for temperature and pressure are currently  $\langle 283.15, 383.15 \rangle$  and  $\langle 50, 410 \rangle$  K and bar respectively.

`nwneg.log`:

The `nwneg.log` contains a report if the mole vector (mole fractions) of the aqueous phase contains negative values. Both the input values received to the code prior to the problem and the values after the problem has occurred are written to the report. This to ease eventual debugging needed when this problem occur. The report is written as:

```
NEGATIVE VALUES OUT FROM CODE!
```

```
xw(i) INTO code = #  
xw(i+1) INTO code = #  
..  
..  
xw(i+14) INTO code = #  
  
xw(i) OUT from code = #  
xw(i+1) OUT from code = #  
..  
..  
xw(i+14) OUT from code = #
```