PHREEQCRM on HPC

What is PHREEQC?

PHREEQC RM (Reactive Transport Model) is an extension of the PHREEQC program that allows for the simulation of reactive transport in porous media. It is a tool for simulating the movement of water and dissolved chemical species through porous rock and soil, and how they interact with the surrounding minerals and gases. PHREEQC RM can be used to simulate a wide range of geochemical processes, such as weathering, dissolution, precipitation, and adsorption. It can also be used to predict the fate and transport of contaminants in groundwater and surface water systems, and to evaluate the effectiveness of remediation strategies. It can be used to simulate the chemical processes in subsurface environments, including the movement of water, heat, and solutes through porous and fractured rock.

Links:

Official Paper

Versions Available:

The following versions are available on the cluster:

- PhreeqcRM library v3.4.0-12927

How to load PHREEQC?

To load PHREEQC, use the following commands:

```
#Load the PHREEQC module
module load physical/phreeqcrm/3.4.0
```

To verify if the module is loaded correctly, use the following command,

```
# List all the module loaded in the environment
module list
```

In a fresh environment, this should show gcc, openmpi and phreeqcrm module listed.

How to use PHREEQC?

A sample use case of using the PHREEQC-RM library in C++ could involve simulating the transport of dissolved contaminants in a porous aquifer. The following steps provide an overview of how this simulation could be implemented:

1. Prepare input data: Next, you will need to prepare input data for the simulation, including the initial solution composition, the minerals and gases present in the system, and the boundary conditions. This data can be provided in a text file in the form of a PHREEQC input file.
2. Write C++ code: Using C++, write a program that reads the input file and sets up the simulation. The program should include calls to the PHREEQC-RM library functions to initialize the simulation, set the boundary conditions, and advance the simulation in time.
3. Compile and run the code: Once the code is written, you can compile and run the program. The program will run the simulation and output the results at regular intervals.
4. Analyze the results: After the simulation is complete, you can analyze the output data to understand the behavior of the system. The output data includes information such as the concentrations of different species, the saturation indices of minerals, and the pH and electric conductivity of the solution.
5. Repeat and modify: You can repeat the simulation with different input data or parameters to further explore the system and modify the program as needed.

Here is a sample c++ layout,

```
#include <iostream>
#include <phreeqc_rm.h>

int main()
{
```

```
    // Initialize the simulation
    PhreeqcRM prm;
    prm.initialize();

    // Read input file
    prm.readFile("input.dat");

    // Set boundary conditions
    prm.setBoundary(1, 0.0, 0.0); // Fixed-concentration boundary at
left side
    prm.setBoundary(2, 0.0, 1.0); // Fixed-concentration boundary at
right side

    // Set initial conditions
    prm.setInitialPhreeqc();

    // Advance the simulation in time
    for (int t = 0; t < 100; t++) {
        prm.advanceTimeStep();
        double c = prm.getSpeciesConcentration(0, 0, 0);
        std::cout << "Time: " << t << "  Concentration: " << c <<
std::endl;
    }

    // Clean up
    prm.cleanup();

    return 0;
}
```

Here is a sample slurm script to compile and execute the script,

```
#!/bin/bash
#SBATCH --job-name=myprogram
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=01:00:00
#SBATCH --mem=1G
#SBATCH --p main
#SBATHC --qos main

# Load the necessary modules
module load physical/phreeqcrm/3.4.0
# Compile the program
g++ -o myprogram myprogram.cpp -lphreeqcrm -lphreeqc -lm
```

```
# Run the program
./myprogram
```

Where to find help?

If you are confused or need help at any point, please contact OIT at the following address.

https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp