



PhreeqcRM: A reaction module for transport simulators based on the geochemical model PHREEQC[☆]



David L. Parkhurst^{a,*}, Laurin Wissmeier^{b,1}

^a U.S. Geological Survey, Denver Federal Center, Box 25046, MS 413, Denver, CO 80225, USA

^b AF-Consult Switzerland Ltd, Groundwater Protection and Waste Disposal, Täferstrasse 26, CH-5405 Baden, Switzerland

ARTICLE INFO

Article history:

Received 29 January 2015

Revised 1 June 2015

Accepted 2 June 2015

Available online 6 June 2015

Keywords:

Reactive transport modelling

PHREEQC

PHAST

FEFLOW

Parallelization

MoMaS reactive transport benchmark

ABSTRACT

PhreeqcRM is a geochemical reaction module designed specifically to perform equilibrium and kinetic reaction calculations for reactive transport simulators that use an operator-splitting approach. The basic function of the reaction module is to take component concentrations from the model cells of the transport simulator, run geochemical reactions, and return updated component concentrations to the transport simulator. If multicomponent diffusion is modeled (e.g., Nernst–Planck equation), then aqueous species concentrations can be used instead of component concentrations. The reaction capabilities are a complete implementation of the reaction capabilities of PHREEQC. In each cell, the reaction module maintains the composition of all of the reactants, which may include minerals, exchangers, surface complexers, gas phases, solid solutions, and user-defined kinetic reactants.

PhreeqcRM assigns initial and boundary conditions for model cells based on standard PHREEQC input definitions (files or strings) of chemical compositions of solutions and reactants. Additional PhreeqcRM capabilities include methods to eliminate reaction calculations for inactive parts of a model domain, transfer concentrations and other model properties, and retrieve selected results. The module demonstrates good scalability for parallel processing by using multiprocessing with MPI (message passing interface) on distributed memory systems, and limited scalability using multithreading with OpenMP on shared memory systems. PhreeqcRM is written in C++, but interfaces allow methods to be called from C or Fortran. By using the PhreeqcRM reaction module, an existing multicomponent transport simulator can be extended to simulate a wide range of geochemical reactions. Results of the implementation of PhreeqcRM as the reaction engine for transport simulators PHAST and FEFLOW are shown by using an analytical solution and the reactive transport benchmark of MoMaS.

Published by Elsevier Ltd.

1. Introduction

During the past decade, one of the main driving forces in reactive transport modelling has been the ongoing global search for strategies of safe nuclear waste disposal [1–9]. Governmental agencies are taking great efforts to manage radioactive waste and to assess the risks of different disposal schemes. Reactive transport modelling is of interest because of (1) its potential to predict the evolution of contaminant plumes over long time scales, and (2) its capabilities to simulate near-field processes, which are used to improve safety in the design of confining structures and containers. Yet, the search for safe radioac-

tive waste disposal is only one of many environmental issues where reactive transport modelling has become an essential decision support and planning tool.

With the search for alternative forms of energy production, reactive transport modelling is increasingly used to predict long-term effects of geothermal energy usage [10–13]. The mining industry and its regulatory authorities routinely use reactive transport modelling to optimize exploitation strategies, assess the risk for groundwater contamination, and design rehabilitation schemes [14–18]. Another field of application is soil systems, where reactive transport modelling has been applied to issues in agriculture and food production [19–21], water treatment, and contaminant retention [22–25]. Particularly with respect to emerging technologies, such as hydraulic fracturing and shale-gas extraction, reactive transport modelling is a useful tool to estimate profitability and evaluate adverse effects on the environment. Furthermore, reactive transport modelling has been employed successfully to demonstrate the feasibility, dangers, and uncertainties of underground carbon dioxide storage [26–32].

[☆] Any use of trade, product, or firm names in this publication is for descriptive purposes only and does not imply endorsement by the U.S. Government.

* Corresponding author. Tel.: +1 303 236 5098; fax: +1 303 236 5034.

E-mail addresses: dlpark@usgs.gov (D.L. Parkhurst), laurin.wissmeier@afconsult.com (L. Wissmeier).

¹ Tel.: +41 (0)56 483 15 65.

All of these fields need an accurate description of flow and transport in combination with reaction processes, and couplings of existing transport simulators with established geochemistry packages have become increasingly popular [33–35] as the means to provide these simulation capabilities.

With the release of IPhreeqc [36]—a general purpose application programming interface (API) for the geochemical modelling framework PHREEQC [37]—the necessary functionality to use PHREEQC as a reaction engine for transport simulators became available. Since its release, IPhreeqc has been coupled to numerous transport codes in a wide variety of contexts, which indicates the scientific and industrial interest in using PHREEQC as a reaction module. Wissmeier and Barry [38] were the first to develop a coupling with COMSOL, which gave access to the full range of COMSOL's flow and transport capabilities together with the complete set of geochemical reactions in PHREEQC. In addition, they outlined a general strategy for couplings with different flow and transport simulators. On these grounds, Nardi et al. [39] published another COMSOL-IPhreeqc coupling. The free environmental flow- and transport-modelling platform OpenGeoSys [40] has an interface to IPhreeqc for coupled thermo-hydro-mechanical-chemical (THMC) simulations. Specialized tools using IPhreeqc have been developed by Takahashi and Ishida [41] for cementitious materials and Huber et al. [42] for the paper-making process. More recently, Patel et al. [43] implemented IPhreeqc as reaction engine for pore-scale multicomponent reactive transport that used a Lattice-Boltzmann approach.

However, IPhreeqc has limitations for couplings with transport simulators. Its API provides access to all of PHREEQC's reaction capabilities by reading and interpreting PHREEQC script commands from input strings or files. Although this approach provides a comprehensive interface, it requires substantial coding in the client software to compose PHREEQC scripts for exchanging data and running reactions. In this paper, we present a new reaction module based on IPhreeqc called PhreeqcRM that is specifically designed for couplings with environmental flow and transport simulators. It provides a high-level interface that allows multicomponent transport codes to implement geochemical reactions with a minimum amount of programming, while maintaining the full functionality of PHREEQC's reaction capabilities.

The paper has two main parts: In Section 2, we provide a generic program flow for the coupling of PhreeqcRM with flow and transport simulators together with detailed information on the key methods of its API. In Section 3, we present the implementation of PhreeqcRM as the reaction engine for the groundwater modelling packages PHAST [44,45] and FEFLOW [46]. The codes are verified through two test cases: a three-dimensional (3D) analytical solution by Sun et al. [47] and the MoMaS (Modeling, Mathematics and numerical Simulations related to nuclear waste management problems) reactive transport benchmark of Groupement de Recherche (GdR) [48].

2. The PhreeqcRM API

PHREEQC is a general-purpose geochemical reaction model that has capabilities to model interactions between water and minerals, gases, ion exchangers, surface complexers, and solid solutions. The nonlinear algebraic equations that define these equilibrium reactions are formulated as an optimization problem that is solved by Newton–Raphson iteration. The optimization method is used specifically to determine the stable equilibrium phase assemblage, where each mineral is constrained to be in equilibrium if present, or undersaturated (or saturated) if absent [49]. Kinetic reactions are described by ordinary differential equations (ODEs), which are solved by either an implicit or an explicit ODE solver. With its capabilities for user-defined reactants and kinetic rate expressions, PHREEQC can simulate a wide variety of biogeochemical processes.

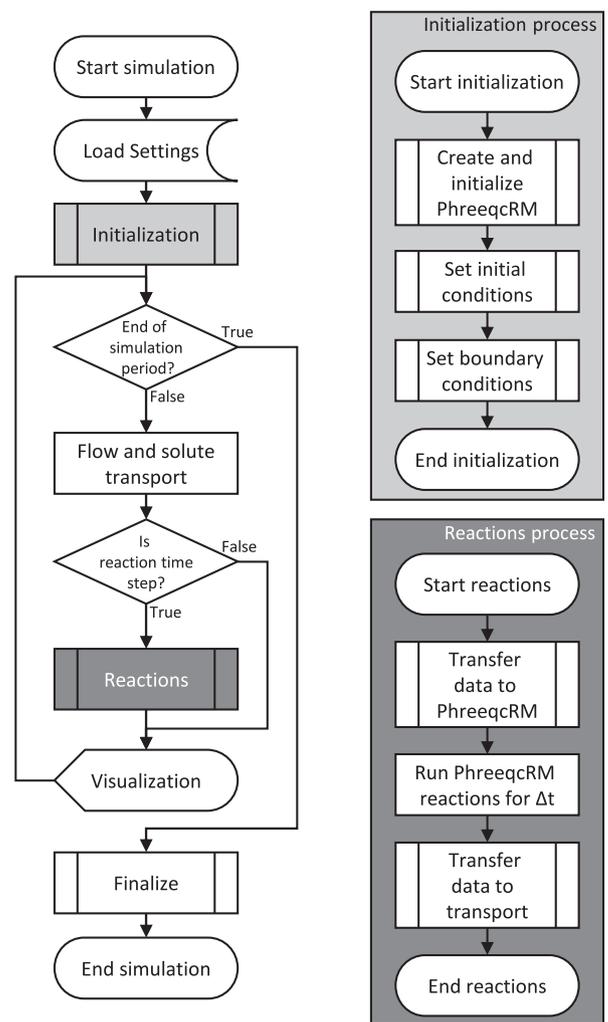


Fig. 1. Generic program flow for a reactive transport simulator using operator splitting.

PHREEQC relies on a set of keyword data blocks to define the types and compositions of the reactants. IPhreeqc [36] encapsulates the capabilities of PHREEQC in a C++ class (with wrappers for C and Fortran) to facilitate integration of PHREEQC into other computer programs. However, IPhreeqc relies primarily on processing strings that define keyword data blocks and arrays of selected-output to automate the use of PHREEQC. For reactive transport modelling, translating cell solutions to strings and processing output arrays item-by-item are tedious programming efforts and may be inefficient for running millions of calculations.

PhreeqcRM is a C++ class that encapsulates IPhreeqc and is designed specifically for performing reaction calculations in reactive transport simulators. The methods of the class allow data arrays to be transferred and reaction calculations to be performed without the use of keyword data blocks. Selected output data can be obtained from memory (e.g., for additional runtime calculations) and output to files. The methods in PhreeqcRM include all of the capabilities necessary to implement PHREEQC reaction calculations in a multicomponent transport simulator.

2.1. Implementation of PhreeqcRM in a multicomponent transport simulator

A generic program flow for the implementation of PhreeqcRM as a reaction engine for a transport simulator is presented in Fig. 1.

The main simulation process accounts for the storage of coupling settings and visualization of reaction results in the host application. The specific tasks for PhreeqcRM that the transport simulator needs to execute during the *Initialization*, *Reactions*, and *Finalize* processes are detailed in the following section.

PhreeqcRM is intended to be the reaction-calculation engine in an operator-splitting approach to reactive transport. The simplest operator-splitting approach is the sequential non-iterative approach (SNIA) [50,51], which separates the simultaneous processes of solute transport and reaction into alternating transport and reaction steps. The sequential iterative approach (SIA) could also be implemented with PhreeqcRM, but SNIA is assumed here for simplicity. Reviews on operator-splitting techniques in the field of reactive transport modelling can be found at [50–55]

PHREEQC uses moles of elements, including oxygen and hydrogen, as the definition of a solution. To account for charge imbalance (difference in equivalents between cations and anions) in boundary and initial solutions and to compensate for charge imbalances that may be induced by surface-complexation reactions, charge imbalance is also required for a complete solution definition. The elements plus charge imbalance are referred to as components. PhreeqcRM does not rely on a specific form of the transport equations. If the same form of transport equation applies for all dissolved species, and equation parameters are species-independent, then the aqueous solution may be transported in terms of components rather than solution species, which reduces the number of transported entities.

Alternatively, PhreeqcRM provides functionality to retrieve the solution composition in terms of aqueous species, together with species charge and diffusion coefficient, to facilitate couplings that account for species-dependent transport processes (e.g., Nernst–Planck equation). In this case, it is necessary for the transport equations to account for each aqueous species, while maintaining a net current of zero [1,2]. In the following sections, we consider only component-based transport with species-independent transport parameters.

Numerical solutions of solute transport may lead to negative concentrations, typically due to spurious oscillations close to sharp concentration fronts. Because PHREEQC calculates reactions in terms of moles, negative element/species concentrations from the transport step have to be amended (set to zero) during the reaction step.¹ To avoid mass-balance errors, negative concentrations from the previous transport step have to be added to the result of the reaction step prior to updating concentrations for the following transport step. While conserving mass, reaction calculations with amended (inaccurate) transport concentrations will add to overall model errors. Other implicit or TVD (total variation diminishing) numerical methods may be used for the transport step to minimize spurious oscillations.

2.2. Key methods of the PhreeqcRM API

The use of PhreeqcRM entails a sequence of method calls that create and initialize the instance, set initial conditions and reactions for the model cells, determine concentrations for boundary conditions, run reaction calculations for a series of time steps, and finalize the instance. Although C++ methods are described here, equivalent C and Fortran subroutines, generally with the same names preceded by “RM_”, are available for transport codes written in these languages. [PhreeqcRM is available at http://wwwbrr.cr.usgs.gov/projects/GWC_coupled/phreeqc. Included in the distributions are source files, compilation files (CMake and configure), documentation of all methods of the API, and simple advection examples in C, Fortran, and C++. Tables 1–8 are excerpts from the C++ example.]

¹ This does not include the charge imbalance component whose codomain includes negative values.

Table 1
Create PhreeqcRM.

```
int nxyz = 40;
int nthreads = 3;
PhreeqcRM phreeqc_rm(nxyz, nthreads);
phreeqc_rm.SetComponentH2O(false);
phreeqc_rm.SetFilePrefix("Advect_cpp");
phreeqc_rm.OpenFiles();
// grid2chem is a mapping from transport cells to reaction cells
std::vector<int> grid2chem;
grid2chem.resize(nxyz, -1);
for (int i = 0; i < nxyz / 2; i++)
{
    grid2chem[i] = i;
    grid2chem[i + nxyz / 2] = i;
}
status = phreeqc_rm.CreateMapping(grid2chem);
std::vector<double> sat, por, rv;
sat.resize(nxyz, 1.0);
phreeqc_rm.SetSaturation(sat);
por.resize(nxyz, 0.2);
phreeqc_rm.SetPorosity(por);
rv.resize(nxyz, 1.0);
phreeqc_rm.SetPoreVolume(rv);
phreeqc_rm.SetUnitsSolution(2);
phreeqc_rm.SetUnitsPAssemblage(1);
phreeqc_rm.SetUnitsExchange(1);
phreeqc_rm.SetUnitsSurface(1);
phreeqc_rm.SetUnitsGasPhase(1);
phreeqc_rm.SetUnitsSAssemblage(1);
phreeqc_rm.SetUnitsKinetics(1);
```

Table 2
Initial conditions and reaction for each cell.

```
phreeqc_rm.LoadDatabase("phreeqc.dat");
bool workers = true
bool initial_phreeqc = true
bool utility = true
phreeqc_rm.RunFile(workers, initial_phreeqc, utility, "advect.pqi");
std::vector<int> ic1;
ic1.resize(nxyz * 7, -1);
for (int i = 0; i < nxyz; i++)
{
    ic1[i] = 1; // Solution 1
    ic1[2 * nxyz + i] = 1; // Exchange 1
}
status = phreeqc_rm.InitialPhreeqc2Module(ic1);
int ncomps = phreeqc_rm.FindComponents();
const std::vector<std::string> &components = phreeqc_rm.GetComponents();
std::vector<double> c;
c.resize(nxyz * components.size());
phreeqc_rm.SetTime(0.0);
phreeqc_rm.SetTimeStep(0.0);
phreeqc_rm.RunCells();
phreeqc_rm.GetConcentrations(c);
```

Table 3
Boundary conditions.

```
std::vector<double> bc_conc;
std::vector<int> bc1;
int nbound = 1;
// bc1 is solution 0
bc1.resize(nbound, 0);
phreeqc_rm.InitialPhreeqc2Concentrations(bc_conc, bc1);
```

Table 4
Transferring data to PhreeqcRM.

```
// spatial arrays
phreeqc_rm.SetPorosity(por);
phreeqc_rm.SetSaturation(sat);
phreeqc_rm.SetTemperature(temperature);
phreeqc_rm.SetPressure(pressure);
phreeqc_rm.SetConcentrations(c);
// set selected output
phreeqc_rm.SetSelectedOutputOn(selected_output_on);
```

Table 5
Time stepping and reaction calculations.

```

phreeqc_rm.SetTimeStep(time_step);
time += time_step;
status = phreeqc_rm.SetTime(time);
status = phreeqc_rm.RunCells();

```

Table 6
Transfer data to transport model.

```

std::vector<double> density;
phreeqc_rm.GetDensity(density);
const std::vector<double> &volume = phreeqc_rm.GetSolutionVolume();
std::vector<double> so;
phreeqc_rm.GetSelectedOutput(so);

```

Table 7
Finalize PhreeqcRM.

```

phreeqc_rm.CloseFiles();

```

Table 8
Example code for an MPI worker.

```

int nxyz = 1;
PhreeqcRM phreeqc_rm(nxyz, MPI_COMM_WORLD);
MP_Type comm = MPI_COMM_WORLD;
int mpi_myself;
if (MPI_Comm_rank(MPI_COMM_WORLD, &mpi_myself) != MPI_SUCCESS)
{
    exit(4);
}
if (mpi_myself > 0)
{
    phreeqc_rm.MpiWorker();
    return EXIT_SUCCESS;
}

```

2.2.1. Create PhreeqcRM

As a C++ class, a constructor creates an instance of PhreeqcRM. Assuming that the source code has been compiled for use with multiple threads (with `USE_OPENMP` as pre-processor definition), the number of threads (*nthreads*) and the number of transport cells (*nxyz*) that represent the nodes or cells in the transport simulator are parameters for the constructor. For multiple processes (with `USE_MPI` as pre-processor definition), the constructor requires an MPI (message passing interface) communicator and the number of transport cells.

The transport simulator must transport each component and provide new concentrations to PhreeqcRM at each time step. The minimum set of chemical components for PhreeqcRM is H (hydrogen), O (oxygen), and charge (imbalance). However, because the number of moles of water is large compared to all other solutes, the non-water H and O concentrations are several orders of magnitude smaller than the total concentrations, yet PHREEQC requires accurate accounting of the non-water H and O. It is numerically advantageous to consider water as a separate component; therefore, the default set of components is water, total H minus H in water, total O minus O in water, and charge. Water can be included or excluded as a separate component by the use of the method `SetComponentWater`. Including water as a component will be numerically more robust; whereas excluding water as a component requires one less component transport calculation, which could save calculation time. Often the calculation time for a transport calculation is small relative to the reaction calculations, so the expense of one extra transport component for water (the default) may be minimal.

The number of cells in the reaction module may differ from the number of cells in the transport simulator because of inactive zones or symmetry. Inactive zones are common in rectangular finite difference block meshes, where some grid cells may be inactive based on

topography or geology. The consideration of symmetry is motivated by PHAST, which requires a minimum of two nodes in each coordinate direction. In a logically 1D problem, chemistry need only be run on one quarter of the PHAST nodes, resulting in a substantial savings in calculation time. A many-to-one mapping can be defined that translates each transport-cell number to a reaction-cell number, such that the number of reaction cells is less than or equal to the number of transport cells (`CreateMapping` method). Regardless of the mapping, PhreeqcRM returns results for every transport cell.

The reaction cells are defined to have a representative volume, *RV*, of 1 L by default. A solution defined in PhreeqcRM input is adjusted (maintaining the same concentrations) to a volume equal to the water content, *l_C* (L), according to

$$l_C = S \times n \times RV, \quad (1)$$

where *S* is the liquid phase saturation (unitless), and *n* is porosity (unitless). The saturation, porosity, and representative volume are set with the methods `SetSaturation`, `SetPorosity`, and `SetRepresentativeVolume`, respectively.

The concentration units of dissolved constituents, *c_{i,j}*, used in the transport simulator can be *c_i* (mg/L), *c^{''}_i* (mol/L), or *c^{'''}_i* (kg/kg solution); the method `SetUnitsSolution` specifies the concentration units of the transport simulator. The concentrations from the transport simulator are converted in PhreeqcRM to moles per cell (*m_{i,C}*) by using the following equations:

$$m_{i,C} = \frac{c'_i l_C}{g_i}, \quad (2)$$

$$m_{i,C} = c''_i l_C, \quad (3)$$

$$m_{i,C} = \frac{c'''_i \rho l_C}{g_i}, \quad (4)$$

where *g_i* is gram formula weight of component/species *i* (g/mol) and *ρ* is the solution density (kg/L) as set by the `SetDensity` method. Note, that the charge imbalance component is always expressed in terms of equivalents independent of the unit settings.

To convert from solid-phase moles in the PhreeqcRM input, *s_p*, to the moles of solid phases in the representative volume of the reaction cell, *s_C* (mol), one of the following definitions can be selected:

$$s_C = s_p RV, \quad (5)$$

$$s_C = s_p n RV, \quad (6)$$

$$s_C = s_p (1 - n) RV. \quad (7)$$

With Eq. (5), the solid-phase input in PhreeqcRM (*s_p*) is mol/L of porous media; with Eq. (6), *s_p* is mol/L of pore volume (fully saturated water volume); and with Eq. (7), *s_p* is mol/L of solid volume. The conversion from the definition in a PHREEQC input file to moles in the cell using one of these three definitions is specified for each reactant type using the methods `SetUnitsPPassemblage`, `SetUnitsExchange`, `SetUnitsSurface`, `SetUnitsGasPhase`, `SetUnitsSSassemblage`, and `SetUnitsKinetics`.

Kinetic reactions in PHREEQC are defined as the transfer of components to or from the solution in terms of moles. Care must be taken to convert rate expressions (PHREEQC RATES data block) from change in concentration per time ($\frac{dc_i}{dt}$) to molar change per time ($\frac{dn_i}{dt}$) by multiplying by the current solution volume (PHREEQC Basic function `SOLN_VOL`).

A sample of C++ code illustrates the creation of a PhreeqcRM instance and setting some of its properties (Table 1).

2.2.2. Set initial conditions

Initial conditions for solution compositions and solid and gas phase reactants can be read from one or more PHREEQC input files or strings and then distributed to the model cells. An initial solution composition has to be assigned to each cell. In addition, sets of reactants can be assigned to each cell to define the types of reactions

that can occur in the cell and the initial number of moles of each reactant, for example the initial amount of a mineral or the number of ion-exchange sites.

A reaction module (an instance of `PhreeqcRM`) has at least three `IPhreeqc` instances, one or more workers, an input-processing instance called the `InitialPhreeqc` instance, and a utility instance, which is optionally used for special calculations (for example calculating the pH of a mixture of water extracted from a well). If using MPI, each process has a reaction module with one worker and a total of three `IPhreeqc` instances. If using OpenMP, the reaction module has a number of workers equal to the number of threads specified in the constructor.

The `LoadDatabase` method loads a PHREEQC database for all of the `IPhreeqc` instances. The PHREEQC database defines a set of elements and corresponding thermodynamic data for the aqueous species, gas components, and mineral phases derived from these elements. The database determines which type of aqueous model will be used—the WATEQ ion-association, LLNL (Lawrence Livermore National Laboratory) ion-association, Pitzer, or SIT (specific interaction theory) model. The database may also contain thermodynamic data for ion-exchange and surface-complexation processes and rate expressions for kinetic reactions. Like PHREEQC, it is possible to modify or augment the definitions of the database with other input files; however, the thermodynamic definitions must be consistent for all of the `IPhreeqc` instances.

The `RunFile` method reads and executes a PHREEQC input file by any combination of `IPhreeqc` instances: worker, `InitialPhreeqc`, and utility. If only initial or boundary conditions are included in the input file, then only the `InitialPhreeqc` instance needs to run the file. If additions to the database are part of an input file, then all of the `IPhreeqc` instances should run the file so that all instances use the same thermodynamic data and aqueous model. Selected output definitions (combinations of the PHREEQC data blocks `SELECTED_OUTPUT` and `USER_PUNCH`) need to be read by the worker instances to allow extraction of selected data for visualization or output to files.

The strategy for assigning initial conditions to the reaction cells is to transfer definitions from the `InitialPhreeqc` instance to the worker instances. Once the `InitialPhreeqc` instance reads a PHREEQC input file, a set of solutions and reactants is present within the `InitialPhreeqc` instance, each identified by a user number (integer), as specified in the input file. These solutions and reactants can be distributed to the reaction cells by use of the `InitialPhreeqc2Module` method. The argument to the method is an array of integers that specifies the solutions and reactants, as identified by user numbers that are assigned to each transport cell. [Table 2](#) displays sample code for the initialization of reaction cells and initial solution conditions.

Instead of defining all reactants in all cells with `InitialPhreeqc2Module`, an alternative way to distribute initial conditions is to run a PHREEQC input file with the `InitialPhreeqc` instance and select one user number to assign to each cell in a list of transport cells. All of the solutions and reactants with that user number will be distributed to each transport cell in the list. It is an error if a reaction cell has not been initialized with at least a solution definition.

It is convenient to get a list of components that have been defined in the solutions and reactants of the `InitialPhreeqc` instance. This list is the set of components that needs to be transported by the transport simulator. The `FindComponents` method accumulates a list of components that have been used in the `InitialPhreeqc` instance. The method can be called multiple times, once after each call to `RunFile` for the `InitialPhreeqc` instance. The method `GetComponents` returns a list of all components that have been accumulated, including water, hydrogen, oxygen, charge, and any other element that has been defined in a solution, equilibrium phase assemblage, surface, gas phase, solid solution, or kinetic reaction definition.

Once the initial conditions for solutions and reactions have been defined, it is useful to ensure that the equilibrium reactants (all except kinetic reactants) are equilibrated with the aqueous solutions in each cell by using the `RunCells` method. By setting the time step for the `RunCells` calculation to 0 s (`SetDeltaTime`), no kinetic reactions are run, but all other reactants equilibrate with the solution, perhaps resulting in revised initial conditions for the solutions and the equilibrium reactants. The revised solution compositions can be retrieved with the `GetConcentrations` method for use in the first transport step.

2.2.3. Set boundary conditions

Boundary-condition aqueous concentrations also are defined through PHREEQC input files or strings. Component concentrations can be extracted from the `InitialPhreeqc` instance for use as boundary conditions in transport calculations. The method `InitialPhreeqc2Concentrations` takes a list of user numbers corresponding to solution definitions in the `InitialPhreeqc` instance and generates an array of concentrations for each component for each solution. These concentrations may then be applied to the set of boundary conditions, cells or nodes, in the transport simulator. Sample code for the calculation of component boundary concentrations is displayed in [Table 3](#).

2.2.4. Transfer data to PhreeqcRM

`SetTemperature`, `SetPressure`, `SetSaturation`, `SetPorosity`, and `SetDensity` can be used to define the corresponding properties in the reaction cells after a transport step. `SetConcentrations` transfers the transported concentrations to the reaction module, but uses the current saturation, porosity, representative volume, and possibly density (for mass-fraction concentrations) to set the number of moles of elements in each of the reaction cells.

Selected output, as defined by `SELECTED_OUTPUT` and `USER_PUNCH` data blocks in PHREEQC input, is the primary means for obtaining geochemical output for a time step. Selected output can be activated or inactivated with the `SetSelectedOutputOn` method. Sample code for the transfer of data from the transport simulator to `PhreeqcRM` is displayed in [Table 4](#).

2.2.5. Run reactions

The current simulation time is set with the method `SetTime` and the time step for the reaction calculations is set with the method `SetDeltaTime`. Equilibrium and kinetic reactions for the time step are run with the `RunCells` method. Sample code for running a typical reaction step in `PhreeqcRM` is provided in [Table 5](#).

2.2.6. Transfer data to the transport model

After the reaction calculations, the newly calculated solution volume, solution density, and saturation may be retrieved with `GetSolutionVolume`, `GetDensity`, and `GetSaturation` methods. When the volume of solution increases as result of reactions, the saturation (solution volume divided by the pore volume) may be larger than one. Similarly, if the solution volume decreases, the calculated saturation may be less than one, even for a saturated flow system.

Selected output results can be retrieved with the `GetSelectedOutput` method (provided it has been enabled with `SetSelectedOutputOn`) for additional runtime calculations (e.g., translation of changes in mineral composition to changes in porosity and conductivity) or visualization. Sample code to transfer data to the transport model after a reaction step is provided in [Table 6](#).

2.2.7. Finalize PhreeqcRM

After time-stepping and reaction calculations are completed, the output and log files can be closed with the `CloseFiles` method ([Table 7](#)). In C++, the module is destroyed and memory is freed when the destructor is invoked. In Fortran and C, the `RM_Destroy` subroutine will destroy the module.

2.3. Parallelization

PhreeqcRM is designed for parallel reaction calculations, where the user only needs to select the parallelization method at compile time and the number of threads or processes at runtime.

Operator-splitting in reactive transport calculations is ideally suited to parallelization because the reaction calculation for each cell is independent of all other cells. Thus, the reaction calculations for the cells can be distributed to multiple processors by using multiple threads, on a shared memory system, or to multiple processes, on a distributed memory system. By using preprocessor definitions (either `USE_OPENMP` or `USE_MPI`), PhreeqcRM can be compiled to use OpenMP, which enables multiple threads, or MPI, which enables multiple processes. For OpenMP the number of threads is defined when creating a PhreeqcRM instance. For MPI, the number of processes and the set of host computers are defined through arguments to `mpixec`, the command that launches an MPI job. PhreeqcRM has been successfully compiled with OpenMP implementations for Windows® (Visual Studio® 2010/2012/2013) and Scientific Linux and with MPI implementations for Windows® (MSMPI from Microsoft®) and Linux (OPENMPI, version 1.5.4).

With the implementation of multiple threads, each thread has access to all of the data of the reaction module. Several loops within PhreeqcRM are parallelized by using OpenMP directives, most notably a loop in `RunCells`, where each thread runs reaction calculations on an assigned set of cells.

The parallelization for multiple processes is complicated by the fact that the data necessary to perform a reaction calculation for a cell must be distributed to the process that is to run reaction calculations for that cell. Particularly, the solution concentrations from the transport simulation must be distributed to the processes at each time step so that the new concentrations are used in the reaction calculations. PhreeqcRM uses a client-server model to perform a variety of distributed tasks, including reaction calculations. The root process is the client (manager), and the non-client processes are the servers (workers). For example, when the method `RunCells` is called, the manager sends a message to the workers that a `RunCells` task must be done. The workers interpret the message in `MpiWorker` and run the `RunCells` method. Within the `RunCells` method, the manager and workers do the reaction calculations for the set of cells for which each is responsible. The workers then wait for the next task message in `MpiWorker`. This same manager-worker structure is used by each method that requires action by the workers, including methods that distribute data from the manager to the workers, such as concentrations, saturations, and porosities, and methods that retrieve data from the workers back to the manager, such as concentrations, solution volumes, densities, and selected output.

An example of code that allows an MPI worker to participate in MPI calculations is shown in Table 8. First, the MPI process creates a reaction module. The number of user grid cells (`nxyz`) is not important for the workers because the manager sets its value when the worker PhreeqcRM instance is created. If the MPI process number (`mpi_myself`) is greater than zero, the process is a worker, and `MpiWorker` is called, which is a loop that waits for a task message from the manager. The worker then processes tasks from the manager until the manager calls `MpiWorkerBreak`, which indicates that processing by the worker is complete. The worker returns from `MpiWorker` and exits.

PhreeqcRM implements the algorithm described in Parkhurst et al. [45] to balance the computational load among the number of threads or processes. A list of M cells is divided into k tasks, whereby the first m_1 cells in the list are assigned to the first task, the next m_2 cells are assigned to the second task, and so on, until the final m_k cells are assigned to the k th task, with the provision that $\sum_{i=1}^k m_i = M$. The load-balancing algorithm adjusts the set of m_i to try to achieve an equal calculation time for each of the tasks. Preliminary indications

are that the algorithm works well for small numbers of tasks, but is not effective for large numbers of tasks on a cluster of computers. Methods to set the load balancing are `SetRebalanceByCell` and `SetRebalanceFraction`.

3. Code verification

PhreeqcRM is implemented as the reaction engine for reactive transport calculations in PHAST [45] and FEFLOW [46]. Both simulators use SNIA for the coupling of transport and reactions. A 3D reactive transport analytical solution [47] and a series of MoMaS reactive transport benchmarks [48] are used to test the two simulators.

3.1. Implementation of PhreeqcRM as the reaction engine in PHAST

For PHAST, PhreeqcRM replaces the old coupling between transport and PHREEQC; however, there are no changes in the usage of the program—input, output, and documentation are the same. The previous version of PHAST (version 2) relied on the source code of PHREEQC version 2 with additional coding to define initial and boundary conditions, transfer concentrations from transport to PHREEQC, run reaction calculations, and return concentrations for transport. PhreeqcRM is designed to perform these functions for PHAST, and the functions of the previous PHAST version correspond closely with the methods of PhreeqcRM.

PHAST's finite-difference nodes are linked to reaction cells in PhreeqcRM by a mapping that accounts for inactive zones and removes redundancies based on symmetry. PHAST assumes constant fluid properties and ignores changes in density and fluid volume produced by reactions. Concentrations are transferred to the module before reaction calculations and retrieved from the module after reaction calculations.

New code is introduced to PHAST to retrieve the selected-output data, and write files in the same format as the previous version, either as text or in HDF (hierarchical data format). PHAST has another output format that is used to save solution definitions for a specified set of cells for use as boundary conditions in subsequent simulations on a more refined grid. To recreate this file format, concentrations are extracted from the concentration array used for transport calculations and transferred to the utility `IPhreeqc` instance to create PHREEQC solutions (`Concentrations2Utility` method). These solution definitions in the utility instance are then speciated and written to file by the `RunString` method using the PHREEQC keywords `RUN_CELLS` and `DUMP`.

The other major coding effort for the new PHAST version involves parallelizing the transport calculations for each component. Fortran modules are written to contain all of the data necessary for a transport calculation, and methods are added to perform these calculations in parallel. For multithreading, a single loop is parallelized with OpenMP directives that assign component-transport calculations to individual threads. For multiprocessing, PHAST makes use of the same processes used to parallelize the reaction calculations. To do the transport calculations with MPI, the method `SetMpiWorkerCallbackFortran` is called by each worker process to register a Fortran subroutine. This Fortran subroutine has a single integer argument, and calls other Fortran subroutines (depending on the argument value) that transfer data, perform transport calculations, and retrieve data from those processes that do the transport calculations. The manager process sends messages to the worker processes that determine the sequence of subroutines run by the workers.

3.2. Implementation of PhreeqcRM as the reaction engine for FEFLOW

PhreeqcRM is implemented as an optional plugin for the finite-element groundwater modelling software FEFLOW [46]. By default,

FEFLOW employs a predictor–corrector method with second-order accurate explicit predictor based on the Adams–Bashforth method and the second-order accurate trapezoid rule as corrector for time discretization. Reactions are calculated at predefined time steps or at every n th transport step, in which case the length of the time step is determined by the automatic time-stepping algorithm.

FEFLOW's interface manager (IFM) provides the API for the coupling as well as plugin methods that execute at certain events during program flow. Similar to PHAST, these event-based methods are implemented as callbacks for the FEFLOW main program. For the problem definition, result visualization, and result storage, FEFLOW's existing infrastructure and GUI (graphical user interface) elements are used as much as possible. New GUI elements are designed for specific coupling settings and for the association of PHREEQC input files with FEFLOW boundary and initial conditions.

Every node in FEFLOW's finite-element mesh corresponds to a reaction cell in PhreeqCRM. Component concentrations, solution temperature and pressure, and liquid phase saturation are transferred from FEFLOW's transport nodes to the reaction cells in PhreeqCRM before each reaction step. Saturation is transferred from PhreeqCRM to FEFLOW after each reaction step, which allows FEFLOW to simulate saturation-modifying chemical reactions.

The starting point of a coupled simulation is a fully functional FEFLOW model for flow and transport with at least one (placeholder) mass transport species. Transport properties of this species (e.g., diffusion coefficient, porosity) are used for all geochemical components. Geochemical components are determined from the associated PHREEQC input files and added automatically at the start of the simulation. Furthermore, the FEFLOW model requires a time series (“power curve” in FEFLOW terminology) that defines reaction steps and changes in boundary conditions and a named value distribution (“nodal user data” in FEFLOW terminology) that identifies nodes with the same geochemical initial conditions. Initial liquid phase saturation and temperatures are taken from the FEFLOW model.

In contrast to PHAST, the FEFLOW plugin uses separate PHREEQC input files to define the geochemical boundary and initial conditions. A file selection GUI with immediate error checking is used to associate PHREEQC files with FEFLOW nodes that share the same boundary and initial conditions (that is, nodes with the same value for constant concentration boundary or the nodal user data). For initial conditions, the solution with the highest user number in the file is transferred as the initial solution for all of the specified transport cells; any reactants with the same user number also are transferred. For boundary conditions, the solution with the highest user number in each PHREEQC file is used to define the solution composition at the associated nodes. All coupling-relevant information can be saved in the FEFLOW *fem*-file through the serialization functionality in the IFM.

Output from SELECTED_OUTPUT and USER_PUNCH keywords in the PHREEQC files that are used to define boundary and initial conditions is transferred to FEFLOW as an additional nodal user data definition, which is updated after each reaction step. Nodal user data definitions can be saved together with FEFLOW's result files (*dac*- and *dar*-files). This allows for the visualization of geochemical parameters during and after a coupled simulation using FEFLOW's built-in post-processing tools.

3.3. Kinetic decay-chain test case

The analytical solution of Wexler [56] in combination with the methodology for multi-species transport of Sun et al. [47] is used to verify the calculation of simple kinetic reactions in PhreeqCRM and its implementation as the reaction engine for FEFLOW and PHAST. Results from PHAST and FEFLOW are nearly identical, and the FEFLOW

Table 9
Domain properties.

Property	Value
Domain extent in x -direction (m)	100
Domain extent in y -direction (m)	25
Domain extent in z -direction (m)	15
Flow velocity in x -direction (m/d)	0.2
Longitudinal dispersivity (m)	1.5
Transverse dispersivity (m)	0.15
Source patch extent in y -direction (m)	0–5
Source patch extent in z -direction (m)	0–2.5
Simulation period (d)	400
Operator-splitting time step (d)	10

Table 10
Error norms (unitless).

	C_1	C_2	C_3	C_4
L_1	0.0004	0.0009	0.0008	0.0009
L_2	0.0038	0.0081	0.0024	0.0030
L_∞	0.0929	0.1583	0.0336	0.0385

plugin results are presented here. The example is adapted from Example 2 of the PHAST manual [45] and simulates a decay chain of four artificial species ($C_1 - C_4$) according to the first-order rate expressions

$$\frac{dC_i(t)}{dt} = \begin{cases} -k_1 C_1(t), & i = 1 \\ k_{i-1} C_{i-1} - k_i C_i(t), & 2 \leq i \leq 4 \end{cases} \quad (8)$$

with species concentrations C_i (mol/L) and rate constants $k_1 = 0.05$ 1/d, $k_2 = 0.02$ 1/d, $k_3 = 0.01$ 1/d, and $k_4 = 0.005$ 1/d. C_1 is introduced over a specified area at the center of the y - z plane with flow in x -direction at 0.2 m/d, and the species form and decay in accordance with the specified reaction rates.

The analytical solution is derived for a semi-infinite half-space with Cartesian coordinates where the x -axis is chosen to be the normal vector of the boundary surface with the source patch at $x = 0$. Domain properties for the numerical approximation using the FEFLOW plugin are given in Table 9. Due to symmetry along the x - y and x - z planes, only the top left sector looking in the direction of flow is simulated. At the source patch, a constant concentration boundary with 0 mol/L is set for all species except species C_1 , which has a concentration of 1 mol/L. At all other boundaries, a zero-gradient condition is employed for solute transport.

Results in Fig. 2 demonstrate excellent agreement between the FEFLOW plugin and the analytical solution. In addition, Table 10 provides the integrated error norms $L_{1,2,\infty}$ for the entire domain and all species. The employed error norms are defined as

$$L_1 = \frac{1}{N\varphi_{\max}} \sum_{i=1}^N |\varphi_i - \hat{\varphi}_i|, \quad (9)$$

$$L_2 = \sqrt{\frac{1}{N\varphi_{\max}^2} \sum_{i=1}^N |\varphi_i - \hat{\varphi}_i|^2}, \quad (10)$$

$$L_\infty = \frac{1}{\varphi_{\max}} \max |\varphi_i - \hat{\varphi}_i|, \quad (11)$$

where φ_i is the analytical solution at node i , $\hat{\varphi}_i$ is the numerical approximation, φ_{\max} is the maximum domain value, and N is the number of nodes.

By using simulations with shorter operator-splitting time step, it can be shown that differences relative to the analytical solution result mainly from operator splitting. The operator-splitting error is most evident close to the source patch, where advection-dominated inflow leads to an overestimation of concentrations of C_1 , while the

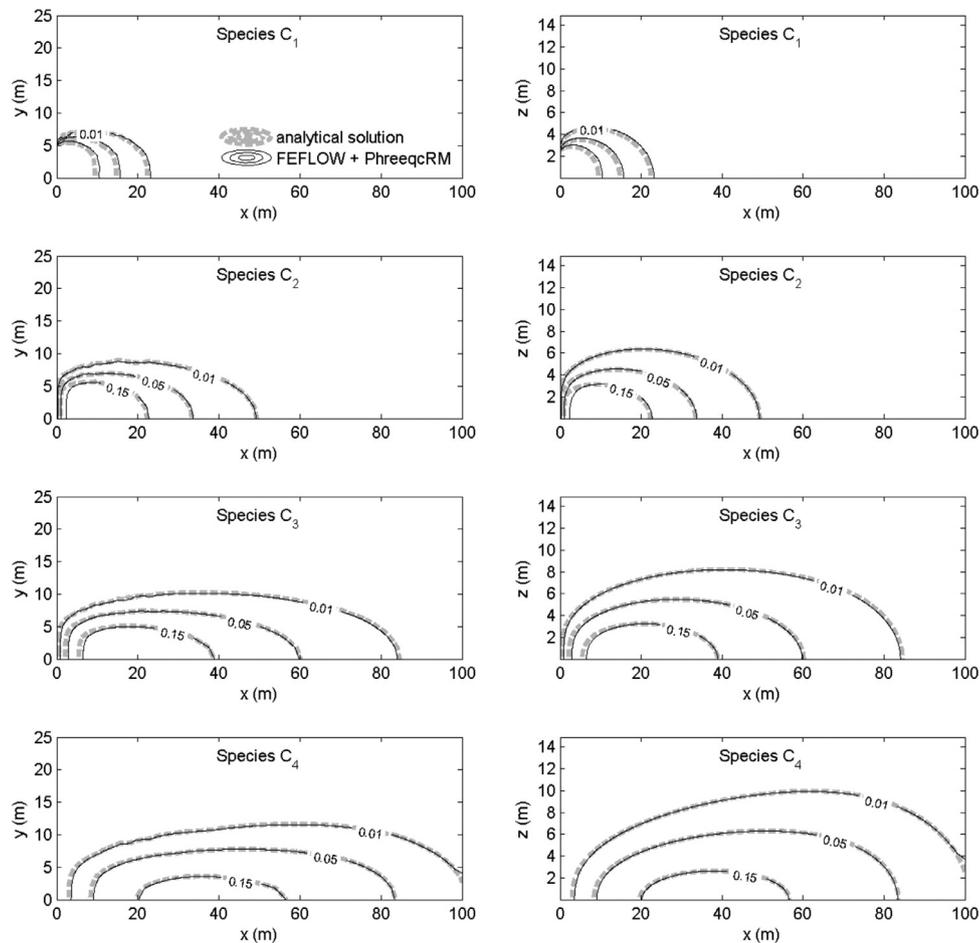


Fig. 2. Species concentrations (mol/L) for a first-order decay chain in a steady flow field as calculated by the FEFLOW plugin compared to the analytical solution of Wexler [56] and Sun et al. [47].

rest of the decay-chain concentrations, C_2 – C_4 , are underestimated. Minor differences of C_4 at the downstream boundary result from the approximation of the semi-infinite half space by a finite domain of 100 m.

Previously, a coupling has been developed for FEFLOW that uses IPhreeqc instead of PhreeqcRM as the reaction engine. Although the IPhreeqc coupling is optimized and uses multi-threading, the computer calculation time for the new version with PhreeqcRM is a factor of 4.8 faster for the reaction calculation in the kinetic decay-chain test case. Given that the underlying numerical method of the two couplings is identical, any advantage of PhreeqcRM is related to data transfer and problem setup. For simulations that require longer calculation times for the nonlinear and ODE equations, the data transfer and problem setup will be relatively smaller, and the two approaches are expected to require similar calculation times. Therefore, although preliminary timings indicate PhreeqcRM is faster relative to IPhreeqc in some cases, the primary advantages of PhreeqcRM are the ease of implementation and maintenance, and the built-in parallelization by both OpenMP and MPI.

3.4. Reactive transport benchmarks of MoMaS

The MoMaS reactive transport benchmarks [48], referred to here as MoMaS, define test cases for steady-state flow with advection- or dispersion-dominated transient solute transport in 1D and 2D domains together with three versions (easy, medium, and hard) of an

artificial reaction network. The general applicability of the SNIA to the MoMaS exercise has been demonstrated for the SPECY code in Carrayrou [57] and Carrayrou et al. [58]. Details of the MoMaS definitions are provided by Carrayrou et al. [48] and Bourgeat et al. [59]. Previous results for the MoMaS benchmark and the influence of different coupling schemes are discussed in de Dieuleveult and Erhel [60] and Hoffmann et al. [61] among others.

Arbitrarily, results of the 1D cases with MPI and OpenMP timings are presented for PHAST, and results of the 2D cases with OpenMP timings are presented for the FEFLOW plugin. Because MoMaS does not define units for dimensional properties, model input and results are presented in terms of unitless time, length, and concentration. Figures showing the complete set of 2D results are provided as additional material.

3.4.1. Reaction network

The hard test case defines equilibrium speciation reactions with five primary solution species and seven secondary solution species, surface complexation without electrical-double-layer calculations that involve one primary surface species and two secondary surface species, and two equilibrium and two kinetic reactions for solid phases. The medium and easy test cases are progressive simplifications of the reaction network from the hard case.

The implementation of the reaction networks in PHREEQC is complicated by the fact that PHREEQC requires positive concentrations of components and balanced chemical reactions, whereas the MoMaS

Table 11
Correspondence between MoMaS and PHREEQC chemical formulas [S and B represent surface-complexation sites].

	Reactant name	MoMaS	PHREEQC
Aqueous species		X1	[X1]
		X2	H ⁺
		X3	[X3] ⁻³
		X4	H ₄ [X4] ⁺³
		X5	[X5] ⁻
Surface species		S	SOH or BOH, depending on site concentration
Equilibrium phases	CP1	X ₂ X ₃	H ₃ [X3]
	CP2	X ₂ X ₅	H[X5]
Kinetic reactants	Cc	X ₂ X ₄	H[X4] gain to solution
	X5 decomposition	X ₂ X ₃ X ₂ X ₅	H ₃ [X3] gain to solution H[X5] loss from solution

reaction network has components that may have negative concentrations and partial mole-balance definitions. However, balanced chemical reactions can be obtained for PHREEQC by using specific definitions of species charge and associated hydrogen ions as shown in Table 11. The negative MoMaS component concentrations can be calculated by the appropriate combination of PHREEQC component and species concentrations.

Because MoMaS reaction networks ignore activity correction, activity coefficients were effectively set to unity by defining the parameters a_0 and b of the extended Debye–Hückel equation to 10^{12} and 0 respectively [49]. The activity of water was assumed to be 1.0 in the MoMaS, so the mass-action expression for OH⁻ was adjusted to remove the effects of nonunity activity of water.

MoMaS defines mass-action equations for bidentate surface complexation in terms of molarity, which has been recognized as problematic [62], whereas PHREEQC uses the equivalent fraction of sites occupied for activity of surface species. To be consistent with the MoMaS definitions, it was necessary to define different equilibrium constants for different surface site concentrations. The relation for a bidentate surface species between K_{eq} for an equivalent fraction mass-action equation and K_{mol} for a molarity mass-action equation is

$$K_{eq} = 2 \times S_{tot} \times K_{mol}, \quad (12)$$

where S_{tot} is the total number of surface sites. Thus, two surfaces were defined that had different equilibrium constants for the bidentate surface species depending on the total number of surface sites in the subdomains of the 1D and 2D MoMaS problems. All definitions used standard PHREEQC input; no modification of the Phreeqrm code was necessary.

3.4.2. 1D MoMaS calculations with PHAST

The 1D easy, advection-dominated benchmark, in which dispersivities are relatively small, requires a fine spatial discretization and small time steps to resolve a small concentration peak in a surface species at early times. Only the advection-dominated benchmarks are presented because the dispersive benchmarks are less restrictive in terms of spatial and temporal discretization. PHAST was discretized with 1760 nodes for comparison with Mayer and MacQuarrie [63] and a 0.05 unit time step (Courant number 0.92) to provide adequate resolution of the surface-species concentration peak. The time step was maintained for the entire 6000-unit simulation time in part because PHAST lacks an automatic time-stepping algorithm, but also to provide a well-defined number of cell calculations that does not depend on variable time stepping. The total number of cell calculations for each simulation was approximately 2.1×10^8 . Backward-in-time and upstream-in-space weighting, which is unconditionally stable but introduces numerical dispersion, was used for all simulations. The estimate of numerical dispersivity (in distance units) due to spatial discretization is $\frac{\Delta x}{2} = 0.60 \times 10^{-3}$ and due to time discretization is $\frac{v\Delta t}{2} = 0.55 \times 10^{-3}$, where v is velocity [64]. The sum of these values is approximately equal to the

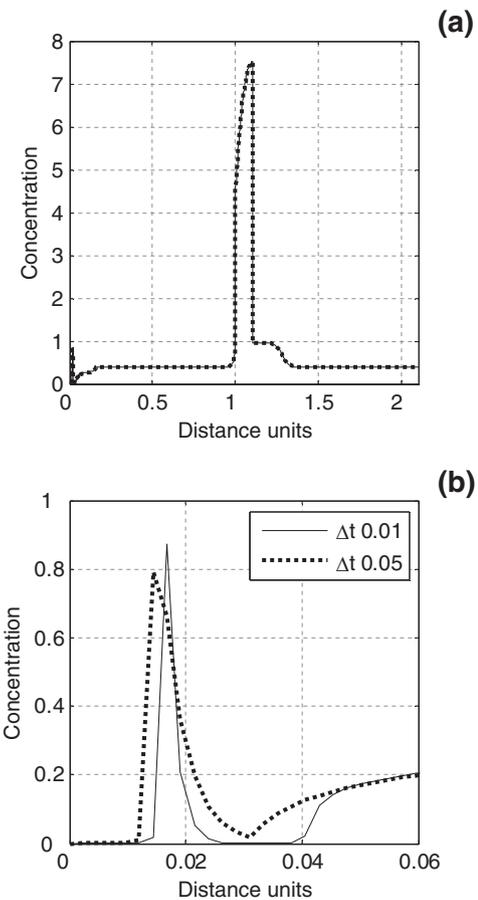


Fig. 3. Concentration of the primary surface species, S, at 10 time units as calculated with 0.05- and 0.01- time-unit steps displayed for (a) the entire domain and (b) the first 0.06 distance units from the inflow boundary.

dispersivity specified by the problem definition (1.0×10^{-2}), which indicates that dispersivity in the calculation (numerical plus specified) is at least twice the specified dispersivity. Operator-splitting dispersivity is not estimated, but, based on spreading in the results compared to results for smaller time steps, may also be of the same order of magnitude as the specified dispersivity.

Fig. 3 shows the resolution of the surface-species peak for 0.05-unit and 0.01-unit time step runs. The smaller time step produces a sharper peak and a deeper valley at distances beyond the peak, indicating that grid and time convergence is not completely achieved. Although, smaller time steps for this simulation are feasible, a better approach is to implement an automatic time-stepping algorithm [63], which would allow for variable time steps that achieve a specified accuracy. The height and location of the peak are similar to the

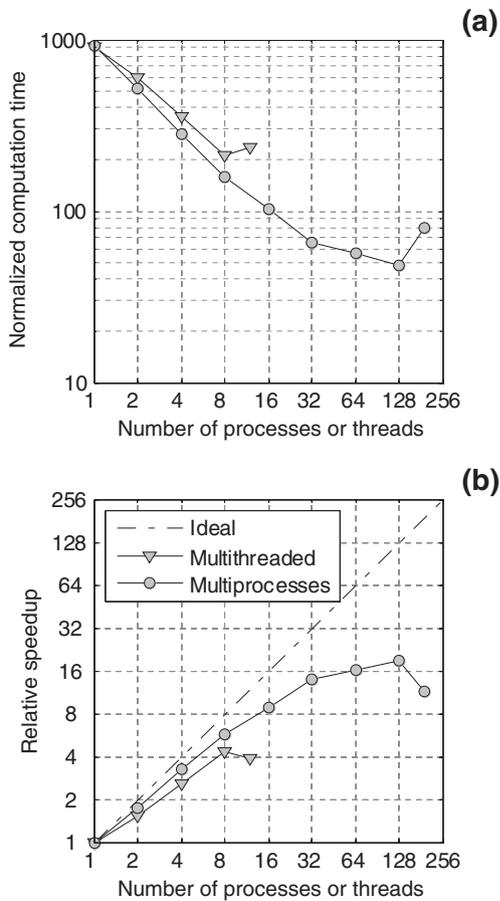


Fig. 4. Log-log plots of (a) chemistry calculation time (including communication) and (b) speedup against number of threads (multithreaded) or number of processes (multiprocessing) for the easy 1D MoMaS reactive transport benchmark.

results of Mayer and MacQuarrie [63] and Amir and Kern [65]. All other concentration results, which are resolved at a coarser scale, are consistent between PHAST and the other simulators.

The speedup of parallelization, defined as of the ratio of the calculation time for a single processor to the calculation time for multiple processors, has been evaluated for the MoMaS easy benchmark by using the multithreaded (OpenMP) and the multiprocessing (MPI) versions of PhreeqcRM in PHAST. Two series of simulations using up to 192 processes or threads were run on a Silicon Graphics International (SGI) computer (256 cores, Intel 3.3 GHz processors, 410 TB RAM, one MoMaS time unit [48] equals 8.81 s) using a Linux operating system. The SLURM (simple Linux utility for resource management) job control software was used to schedule the jobs. For multiprocessing with MPI, the times for the chemistry calculation diminished from about 915 to 48.5 MoMaS time units (Fig. 4), with a maximum speedup of 18.9 using 128 processes. For multithreading, calculation times diminished from 906 to 212 time units, with a maximum speedup of 4.3 using 8 threads. Increasing the number of threads beyond 8 caused a drop in performance, presumably because the data needed for a calculation were not associated with the memory of the core doing the calculation. Thus, PhreeqcRM using MPI provides good scaling to a large number of cores on multiple nodes, whereas PhreeqcRM using OpenMP is appropriate for use with multiple cores on a single node.

A series of runs was made for the easy, medium, and hard benchmarks using 4–192 processes on the SGI computer. Timings were obtained for the chemistry calculations, including the

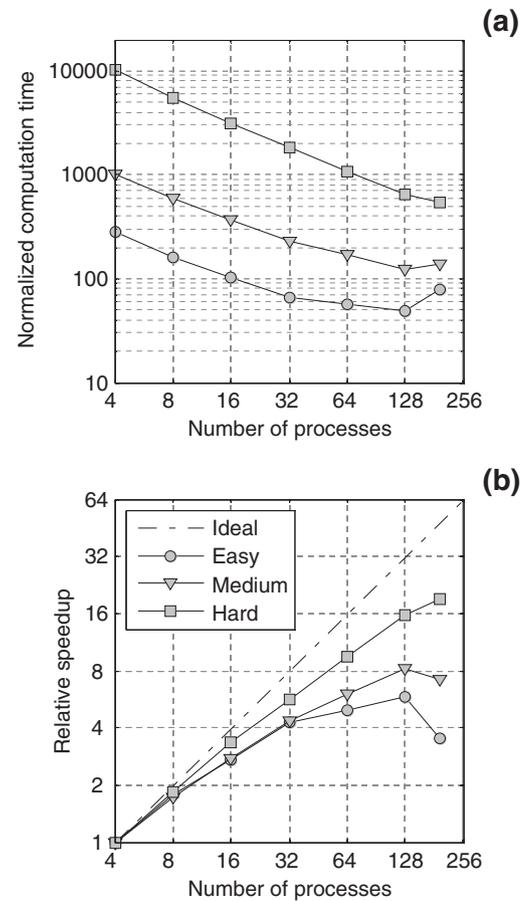


Fig. 5. (a) Normalized reaction-calculation times for PhreeqcRM, including communication time, for the easy, medium, and hard MoMaS 1D-advective benchmarks, and (b) speedup relative to a 4-process base case, as a function of number of MPI processes.

communication time to send and receive data among the MPI processes (Fig. 5).

The addition of more processes resulted in faster calculation times and substantial speedups for all three benchmarks. The speedup is expected to scale best when the communication times are small relative to the reaction-calculation times. For the MoMaS benchmarks, communication times are similar for all three benchmarks, whereas the reaction-calculation times increase in order from the easy to the hard benchmark. Consequently, the number of processors that produce substantial speedups increases from about 32 for the easy benchmark, to 128 for the medium benchmark, to at least 192 for the hard benchmark.

Another consideration in the speedup results is that PhreeqcRM distributes the calculation load by assigning a number of cells to each process. If there are many cells and few processes, a difference of one cell among the processors may have a small effect. With 192 processes and only 1760 reaction cells, the number of cells per process is about nine; even a difference of one cell results in about an 11% difference in load for a processor (assuming all cell calculations are equal).

The MoMaS 1D problem has relatively few cells (1760) and relatively few data items defined for selected output (22), although data are retrieved and written to file often (3000 times). If more data are retrieved from PhreeqcRM or larger output files are written, the communication times will increase relative to the reaction-calculation time, resulting in a decrease in the overall speedup.

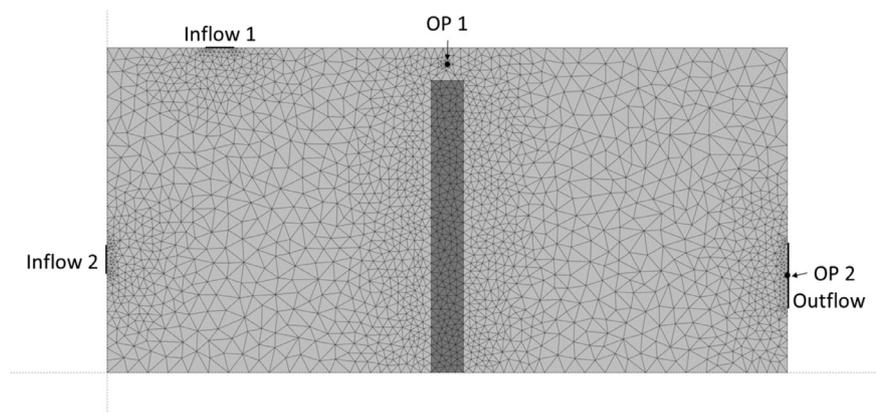


Fig. 6. Finite-element mesh for the MoMaS 2D cases with high conductivity (light gray) and low conductivity (dark gray) subdomains, inflow and outflow zones, and observation points (OP).

Table 12

Calculation times for MoMaS in CPU units.

Reaction network	Transport type	2D FEFLOW	2D fine FEFLOW
Easy	Advective	310	1702
	Dispersive	246	1447
Medium	Advective	356	2073
	Dispersive	296	1937
Hard	Advective	731	5837
	Dispersive	1091	9297

3.4.3. 2D MoMaS calculations with FEFLOW

The irregular finite-element mesh with 3753 elements and 1970 nodes used by the FEFLOW plugin for the 2D cases is displayed in Fig. 6.

For the advective cases, reactions were calculated with an operator-splitting time step of 1 for 100 time units at the beginning of the simulation and after the boundary change at time 5000; otherwise, a time step of 5 time units was used. Despite the suggestion to use the maximum Courant number as an indicator for the appropriate time step with the SNIA [58], the specified time stepping was chosen because FEFLOW uses automatic time stepping solute transport, and the stability of the method is not limited by the Courant condition. Because of the greater solute fluxes in the dispersive cases, these cases were simulated with an operator-splitting time step of 0.1 for the first 20 time units of the simulation and after the boundary change; otherwise, an operator-splitting time step of 1 time unit was used. As an indicator of grid convergence, additional runs were conducted for a fine mesh consisting of 14,930 elements and 7651 nodes with the same operator-splitting time step as the dispersive cases.

The streamline upwinding option was selected for the stabilization of solute transport. For the solution of the non-symmetric matrices from the transport equations, a standard iterative solver was used for the advective cases, and the direct, more stable but slightly slower PARDISO solver was used for the dispersive cases. A L_{∞} error norm of 10^{-4} was employed as an overall convergence criterion for flow and transport.

For the computer used for the calculations (Intel® Core™ i7-4770, 3.40 GHz, 16 GB RAM), one MoMaS time unit [48] equals 7.0 s. Reaction calculations used OpenMP parallelization with eight threads. Timing of the 2D cases in CPU units is presented in Table 12.

The computer calculations in the implementation of PhreeqCRM in FEFLOW are not as fast as other published reactive transport codes (e.g. [66,67]). In the coupling to PhreeqCRM, FEFLOW's automatic time stepping algorithm greatly reduces the transport time step after each reaction step due to abrupt changes and discontinuities

in the concentration fields that result from the chemical reactions. Therefore, overall simulation times are strongly influenced by the frequency of reaction calculations, which in the presented cases has been selected in favor of accuracy over speed.

Selected results of the 2D advective MoMaS cases are presented in Fig. 7. According to Carrayrou et al. [58] the concentration of component X3 at time 1000 offers a suitable means for comparison. Plotted with a similar color map as in Carrayrou et al. [58], Fig. 7a can be directly compared to Carrayrou et al. [58]. Given the bandwidth of results in Carrayrou et al. [58], results from the FEFLOW plugin fit well with results from MIN3P [66,68], RICHY2D [69] and HYTEC [70,71]. Apart from slightly higher concentrations, the simulation with the refined mesh (not shown) yields a very similar concentration pattern.

Fig. 7b shows the concentration of X2 with a similar color map as in Mayer and MacQuarrie [63] to facilitate direct comparison to their Fig. 6c. Simulations with the fine mesh (not shown) yield a smoother reaction edge but otherwise the same general concentration pattern.

Fig. 7c shows the equilibrium mineral species CP1 at time 2000 for the hard advective case. Qualitatively, results are similar to those in Mayer and MacQuarrie [63]; however, maximum concentrations are lower (~ 20 versus ~ 24). Using the fine mesh, Fig. 7d reveals the discretization dependence of maximum CP1 concentrations. With the fine mesh, maximum concentrations and their general distribution are comparable to Mayer and MacQuarrie [63]. In addition to the figures presented here, all results that are requested by the MoMaS definition [48] compare well to results from Hoffmann [69] using RICHY2D.

Fig. 8 verifies the results from the dispersive cases relative to published data in Carrayrou et al. [58] and Mayer and MacQuarrie [63]. For comparability with Carrayrou et al. [58], a similar color scale is used in Fig. 8a and b. Results obtained with the fine mesh show only minor differences compared to results for the coarse mesh (Fig. 8b versus Fig. 8a). Compared to results from Mayer and MacQuarrie [63], component X2 at time 10 (Fig. 8c) has a sharper transition from low to high concentrations and has less lateral spread at Inflow 2 (Fig. 6). Compared to Mayer and MacQuarrie [63], the mineral phase CP1 at time 1000 (Fig. 8d) is removed from a larger part of the domain. Furthermore, the large amounts of this phase at the edge of the low-permeability zone are absent. Both effects may be caused by the employed upwind stabilization of solute transport. Results with the operator-splitting time step of the coarse-meshed advective case show substantial differences compared to the results in Fig. 8. These differences show that when using SNIA, it is necessary to test simulations for time step and mesh convergence in order to demonstrate the reliability of results.

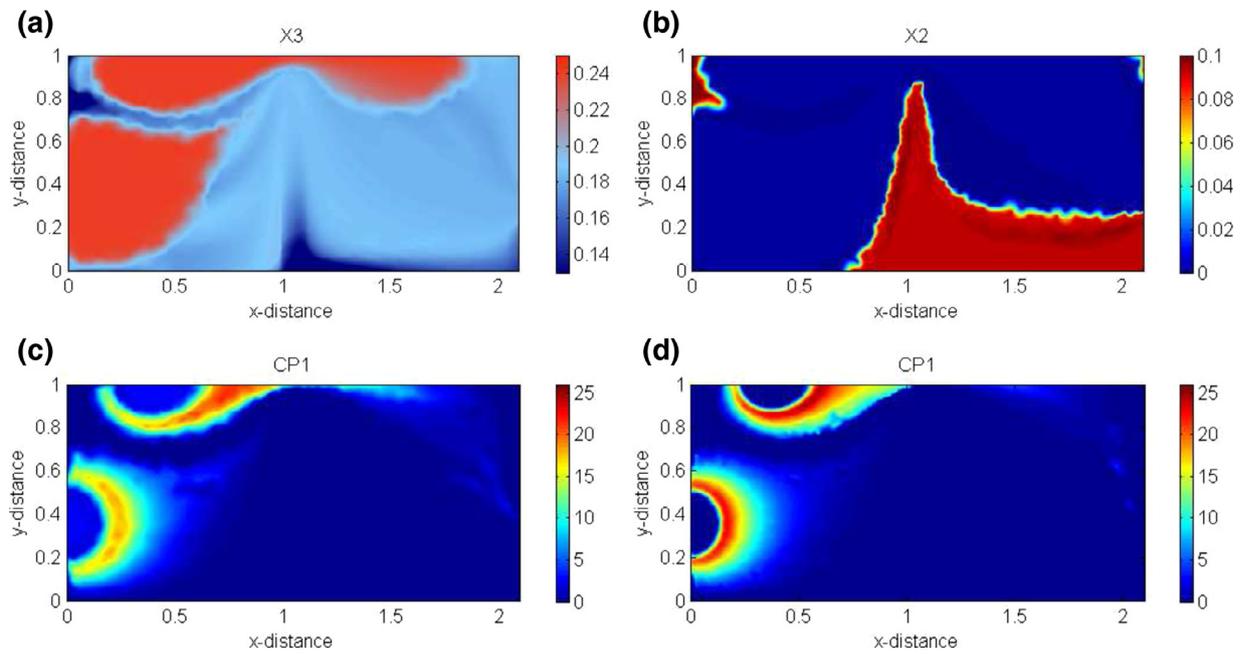


Fig. 7. FEFLOW concentration results for the MoMaS 2D advective cases: (a) X3, easy case at time 1000; (b) X2, medium case at time 1000; (c) CP1, hard case at time 2000; and (d) same as (c) calculated with fine mesh.

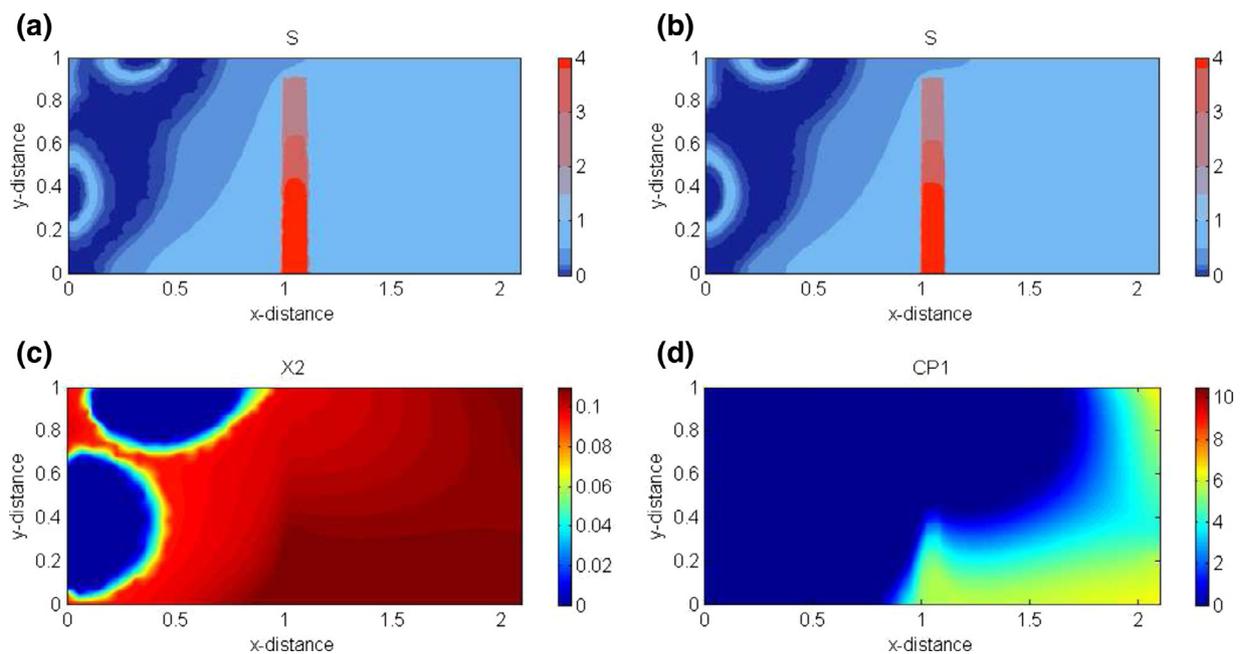


Fig. 8. FEFLOW concentration results for the MoMaS 2D dispersive cases; (a) and (b) S, easy case at time 10 with coarse and fine mesh, respectively; (c) X2, medium case at time 10; (d) CP1, hard case at time 1000.

4. Summary

PhreeqcRM is a module based on the geochemical model PHREEQC that is designed to perform reaction calculations for reactive transport simulators that use an operator-splitting approach. PhreeqcRM is written in C++, but interfaces allow methods to be called from C or Fortran. By using the PhreeqcRM reaction module, an existing multicomponent transport simulator can be extended to simulate a wide range of geochemical reactions. PhreeqcRM relies on IPhreeqc, the previous encapsulation of PHREEQC, but specializes its use for reactive transport. The module has methods to set initial conditions, set boundary conditions, transfer concentrations and

other model properties, run reactions, and retrieve post-reaction results. The module can provide component concentrations for simulators that use species-independent transport parameters or aqueous species concentrations for multicomponent-diffusion simulators. PhreeqcRM has been implemented in two simulators, PHAST and FEFLOW, which have been used to demonstrate the validity and efficiency of the approach on a test problem with an analytical solution and on the MoMaS reactive transport benchmarks. PhreeqcRM is parallelized for OpenMP or MPI, depending on compilation options. Good scalability has been demonstrated for MPI on a multiprocessor computer using up to 192 cores; whereas scalability for OpenMP was limited to a single node.

Acknowledgments

We thank Jeffrey Falgout for his help in using the U.S. Geological Survey computer cluster (yeti) in Lakewood, Colorado. The paper was improved by the reviews of Gary Curtis, Diederik Jacques, and four anonymous reviewers. The contribution of Laurin Wissmeier was partially funded by AF-Consult Switzerland Ltd.

Supplementary Materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.advwatres.2015.06.001.

References

- Appelo CAJ, Wersin P. Multicomponent diffusion modeling in clay systems with application to the diffusion of tritium, iodide, and sodium in Opalinus clay. *Environ Sci Technol* 2007;41(14):5002–7. <http://dx.doi.org/10.1021/es0629256>.
- Appelo CAJ, van Loon LR, Wersin P. Multicomponent diffusion of a suite of tracers (HTO, Cl, Br, I, Na, Sr, Cs) in a single sample of Opalinus clay. *Geochim Cosmochim Acta* 2010;74(4):1201–19. <http://dx.doi.org/10.1016/j.gca.2009.11.013>.
- Arnold J, Kosson DS, Garrabrants A, Meeussen JCL, van der Sloot HA. Solution of the nonlinear Poisson–Boltzmann equation: application to ionic diffusion in cementitious materials. *Cement Concrete Res* 2013;44:8–17. <http://dx.doi.org/10.1016/j.cemconres.2012.10.013>.
- Cochebin B, Trotignon L, Bildstein O, Steefel CI, Lagneau V, van der Lee J. Approaches to modelling coupled flow and reaction in a 2D cementation experiment. *Adv Water Resour* 2008;31(12):1540–51. <http://dx.doi.org/10.1016/j.advwatres.2008.05.007>.
- Kletskova T, Czerwinski K, Gelbard E, Yip S. Modeling cation exchange in zeolitic nuclear waste form. *J Comput-Aided Mater Des* 1999;6(2–3):363–8. <http://dx.doi.org/10.1023/A:1008780522789>.
- Montarnal P, Mügler C, Colin J, Descostes M, Dimier A, Jacquot E. Presentation and use of a reactive transport code in porous media. *Phys Chem Earth* 2007;32(1–7):507–17. <http://dx.doi.org/10.1016/j.pce.2006.01.009>.
- Trotignon L, Devallois V, Peycelon H, Tiffreau C, Bourbon X. Predicting the long term durability of concrete engineered barriers in a geological repository for radioactive waste. *Phys Chem Earth* 2007;32(1–7):259–74. <http://dx.doi.org/10.1016/j.pce.2006.02.049>.
- Zhang GX, Spycher N, Sonnenthal E, Steefel C, Xu TF. Modeling reactive multiphase flow and transport of concentrated solutions. *Nucl Technol* 2008;164(2):180–95.
- Perko J, Mayer KU, Kosakowski G, De Windt L, Govaerts J, Jacques D, et al. Decalcification of cracked cement structures. *Comput Geosci* 2015;1–21. <http://dx.doi.org/10.1007/s10596-014-9467-2>.
- Bozau E, van Berk W. Hydrogeochemical modeling of deep formation water applied to geothermal energy production. In: Proceedings of the fourteenth international symposium on water–rock interaction, WRI 14; 2013. <http://dx.doi.org/10.1016/j.proeps.2013.03.006>.
- Oldenburg CM, Pruess K. Simulation of propagating fronts in geothermal reservoirs with the implicit Leonard total variation diminishing scheme. *Geothermics* 2000;29(1):1–25. [http://dx.doi.org/10.1016/S0375-6505\(99\)00048-6](http://dx.doi.org/10.1016/S0375-6505(99)00048-6).
- Kim J, Sonnenthal E, Rutqvist J. A sequential implicit algorithm of chemo-thermoporomechanics for fractured geothermal reservoirs. *Comput Geosci* 2015;76:59–71. <http://dx.doi.org/10.1016/j.cageo.2014.11.009>.
- Diamond LW, Alt-Epping P. Predictive modelling of mineral scaling, corrosion and the performance of solute geothermometers in a granitoid-hosted, enhanced geothermal system. *Appl Geochem* 2014;51:216–28. <http://dx.doi.org/10.1016/j.apgeochem.2014.09.017>.
- Amos RT, Mayer KU, Blowes DW, Ptacek CJ. Reactive transport modeling of column experiments for the remediation of acid mine drainage. *Environ Sci Technol* 2004;38(11):3131–8. <http://dx.doi.org/10.1021/es0349608>.
- Jurjovec J, Blowes DW, Ptacek CJ, Mayer KU. Multicomponent reactive transport modeling of acid neutralization reactions in mine tailings. *Water Resour Res* 2004;40(11). <http://dx.doi.org/10.1029/2003WR002233>.
- Liu WC, Yang JK, Xiao B. Review on treatment and utilization of bauxite residues in China. *Int J Miner Process* 2009;93(3–4):220–31. <http://dx.doi.org/10.1016/j.minpro.2009.08.005>.
- Wissmeier L, Barry DA, Phillips IR. Predictive hydrogeochemical modelling of bauxite residue sand in field conditions. *J Hazard Mater* 2011;191(1–3):306–24. <http://dx.doi.org/10.1016/j.jhazmat.2011.04.078>.
- Xenidis A, Harokopou AD, Mylona E, Brofas G. Modifying alumina red mud to support a revegetation cover. *J Miner Met Mater* 2005;57(2):42–6. <http://dx.doi.org/10.1007/s11837-005-0214-y>.
- Šimunek J, Hopmans JW. Modeling compensated root water and nutrient uptake. *Ecol Modell* 2009;220(4):505–21. <http://dx.doi.org/10.1016/j.ecolmodel.2008.11.004>.
- Hanson BR, Šimunek J, Hopmans JW. Evaluation of urea–ammonium–nitrate fertigation with drip irrigation using numerical modeling. *Agric Water Manage* 2006;86(1–2):102–13. <http://dx.doi.org/10.1016/j.agwat.2006.06.013>.
- Bailey RT, Gates TK, Halvorson AD. Simulating variably-saturated reactive transport of selenium and nitrogen in agricultural groundwater systems. *J Contam Hydrol* 2013;149:27–45. <http://dx.doi.org/10.1016/j.jconhyd.2013.03.001>.
- Langergraber G. Modeling of processes in subsurface flow constructed wetlands: a review. *Vadose Zone J* 2008;7(2):830–42. <http://dx.doi.org/10.2136/vzj2007.0054>.
- Langergraber G, Rousseau DPL, Garcia J, Mena J. CWM1: a general model to describe biokinetic processes in subsurface flow constructed wetlands. *Water Sci Technol* 2009;59(9):1687–97. <http://dx.doi.org/10.2166/wst.2009.131>.
- Toscano A, Langergraber G, Consoli S, Cirelli GL. Modelling pollutant removal in a pilot-scale two-stage subsurface flow constructed wetlands. *Ecol Eng* 2009;35(2):281–9. <http://dx.doi.org/10.1016/j.ecoleng.2008.07.011>.
- Langergraber G, Šimunek J. Reactive transport modeling of subsurface flow constructed wetlands using the HYDRUS wetland module. *Vadose Zone J* 2012;11(2). <http://dx.doi.org/10.2136/vzj2011.0104>.
- Graupner BJ, Li D, Bauer S. The coupled simulator ECLIPSE–OpenGeoSys for the simulation of CO₂ storage in saline formations. In: 10th international conference on greenhouse gas control technologies; 2011. <http://dx.doi.org/10.1016/j.egypro.2011.02.314>.
- Kvamme B, Liu S. Reactive transport of CO₂ in saline aquifers with implicit geomechanical analysis. *Energy Proc* 2009;1(1):3267–74. <http://dx.doi.org/10.1016/j.egypro.2009.02.112>.
- Nordbotten JM, Kavetski D, Celia MA, Bachu S. Model for CO₂ leakage including multiple geological layers and multiple leaky wells. *Environ Sci Technol* 2009;43(3):743–9. <http://dx.doi.org/10.1021/es801135v>.
- Xu TF, Sonnenthal E, Spycher N, Pruess K. TOUGHREACT—a simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media: applications to geothermal injectivity and CO₂ geological sequestration. *Comput Geosci* 2006;32(2):145–65. <http://dx.doi.org/10.1016/j.cageo.2005.06.014>.
- Xu T, Li J. Reactive transport modeling to address the issue of CO₂ geological sequestration. In: Proceedings of the fourteenth international symposium on water–rock interaction, WRI 14; 2013. <http://dx.doi.org/10.1016/j.proeps.2013.03.153>.
- Zhang Y, Oldenburg CM, Finsterle S, Bodvarsson GS. System-level modeling for economic evaluation of geological CO₂ storage in gas reservoirs. *Energy Convers Manage* 2007;48(6):1827–33. <http://dx.doi.org/10.1016/j.enconman.2007.01.018>.
- Beyer C, Li D, De Lucia M, Kühn M, Bauer S. Modelling CO₂-induced fluid–rock interactions in the Altensalzvedel gas reservoir. Part II: coupled reactive transport simulation. *Environ Earth Sci* 2012;67(2):573–88. <http://dx.doi.org/10.1007/s12665-012-1684-1>.
- Meeussen JCL. ORCHESTRA: an object-oriented framework for implementing chemical equilibrium models. *Environ Sci Technol* 2003;37(6):1175–82. <http://dx.doi.org/10.1021/es025597s>.
- van der Lee J. Thermodynamic and mathematical concepts of CHES. LHM/RD/98/39. Ecole des Mines de Paris; 1998.
- Šimunek J, Jacques D, van Genuchten MT, Mallants D. Multicomponent geochemical transport modeling using HYDRUS-1D and HP1. *J Am Water Resour Assoc* 2006;42(6):1537–47. <http://dx.doi.org/10.1111/j.1752-1688.2006.tb06019.x>.
- Charlton SR, Parkhurst DL. Modules based on the geochemical model PHREEQC for use in scripting and programming languages. *Comput Geosci* 2011;37(10):1653–63. <http://dx.doi.org/10.1016/j.cageo.2011.02.005>.
- Parkhurst D.L., Appelo CAJ. Description of input and examples for PHREEQC Version 3—a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. U.S. Department of the Interior, U.S. Geological Survey Techniques and Methods 6-A43; 2013.
- Wissmeier L, Barry DA. Simulation tool for variably saturated flow with comprehensive geochemical reactions in two- and three-dimensional domains. *Environ Modell Softw* 2011;26(2011):210–18. <http://dx.doi.org/10.1016/j.envsoft.2010.07.005>.
- Nardi A, Ildiart A, Trinchero P, de Vries LM, Molinero J. Interface COMSOL-PHREEQC (iCP), an efficient numerical framework for the solution of coupled multiphysics and geochemistry. *Comput Geosci* 2014;69:10–21. <http://dx.doi.org/10.1016/j.cageo.2014.04.011>.
- Kolditz O, Bauer S, Bilke L, Böttcher N, Delfs JO, Fischer T, et al. OpenGeoSys: an open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media. *Environ Earth Sci* 2012;67(2):589–99. <http://dx.doi.org/10.1007/s12665-012-1546-x>.
- Takahashi Y, Ishida T. Modeling of coupled mass transport and chemical equilibrium in cement-solidified soil contaminated with heavy-metal ions. *Constr Build Mater* 2014;67(Part A):100–7. <http://dx.doi.org/10.1016/j.conbuildmat.2013.10.030>.
- Huber P, Nivelon S, Ottenio P, Nortier P. Coupling a chemical reaction engine with a mass flow balance process simulation for scaling management in papermaking process waters. *Ind Eng Chem Res* 2012;52(1):421–9. <http://dx.doi.org/10.1021/ie300984y>.
- Patel RA, Perko J, Jacques D, De Schutter G, Van Breugel K, Ye G. A versatile pore-scale multicomponent reactive transport approach based on lattice Boltzmann method: application to portlandite dissolution. *Phys Chem Earth* 2014;70–71:127–37. <http://dx.doi.org/10.1016/j.pce.2014.03.001>.
- Parkhurst D.L., Kipp K.L., Engesgaard P., Charlton S.R. PHAST—a program for simulating ground-water flow, solute transport, and multicomponent geochemical reactions. U.S. Geological Survey; 2004.
- Parkhurst D.L., Kipp K.L., Charlton S.R. PHAST Version 2—a program for simulating groundwater flow, solute transport, and multicomponent geochemical reactions. U.S. Geological Survey Techniques and Methods 6-A35; 2010.

- [46] DHI-WASY Software. FEFLOW® 6.2 user manual. 2012.
- [47] Sun Y, Petersen JN, Clement TP. Analytical solutions for multiple species reactive transport in multiple dimensions. *J Contam Hydrol* 1999;35(4):429–40. [http://dx.doi.org/10.1016/S0169-7722\(98\)00105-3](http://dx.doi.org/10.1016/S0169-7722(98)00105-3).
- [48] Carrayrou J, Kern M, Knabner P. Reactive transport benchmark of MoMaS. *Comput Geosci* 2010;14(3):385–92. <http://dx.doi.org/10.1007/s10596-009-9157-7>.
- [49] Parkhurst D.L., Appelo C.A.J. User's guide to PHREEQC (version 2): a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. U.S. Geological Survey; 1999.
- [50] Barry DA, Bajracharya K, Crapper M, Prommer H, Cunningham CJ. Comparison of split-operator methods for solving coupled chemical non-equilibrium reaction/groundwater transport models. *Math Comput Simul* 2000;53(1–2):113–27. [http://dx.doi.org/10.1016/S0378-4754\(00\)00182-8](http://dx.doi.org/10.1016/S0378-4754(00)00182-8).
- [51] Carrayrou J, Mose R, Behra P. Operator-splitting procedures for reactive transport and comparison of mass balance errors. *J Contam Hydrol* 2004;68(3–4):239–68. [http://dx.doi.org/10.1016/S0169-7722\(03\)00141-4](http://dx.doi.org/10.1016/S0169-7722(03)00141-4).
- [52] Barry DA, Miller CT, Culligan-Hensley PJ, Bajracharya K. Analysis of split operator methods for nonlinear and multispecies groundwater chemical transport models. *Math Comput Simul* 1997;43(3–6):331–41. [http://dx.doi.org/10.1016/S0378-4754\(97\)00017-7](http://dx.doi.org/10.1016/S0378-4754(97)00017-7).
- [53] Simpson MJ, Landman KA. Theoretical analysis and physical interpretation of temporal truncation errors in operator split algorithms. *Math Comput Simul* 2008;77(1):9–21. <http://dx.doi.org/10.1016/j.matcom.2007.01.001>.
- [54] Jacques D, Šimůnek J, Mallants D, van Genuchten MT. Operator-splitting errors in coupled reactive transport codes for transient variably saturated flow and contaminant transport in layered soil profiles. *J Contam Hydrol* 2006;88(3–4):197–218. <http://dx.doi.org/10.1016/j.jconhyd.2006.06.008>.
- [55] Valocchi AJ, Malmstead M. Accuracy of operator splitting for advection-dispersion-reaction problems. *Water Resour Res* 1992;28(5):1471–6. <http://dx.doi.org/10.1029/92WR00423>.
- [56] Wexler E.J. Analytical solutions for one-, two-, and three-dimensional solute transport in ground-water systems with uniform flow. U.S. Department of the Interior, U.S. Geological Survey, Techniques of Water-Resources Investigations 3-B7; 1992.
- [57] Carrayrou J. Looking for some reference solutions for the reactive transport benchmark of MoMaS with SPECY. *Comput Geosci* 2010;14(3):393–403. <http://dx.doi.org/10.1007/s10596-009-9161-y>.
- [58] Carrayrou J, Hoffmann J, Knabner P, Krätzle S, de Dieuleveult C, Erhel J, et al. Comparison of numerical methods for simulating strongly nonlinear and heterogeneous reactive transport problems—the MoMaS benchmark case. *Comput Geosci* 2010;14(3):483–502. <http://dx.doi.org/10.1007/s10596-010-9178-2>.
- [59] A. Bourgeat, S. Bryant, J. Carrayrou, A. Dimier, C.J.V. Duijn, M. Kern, et al. GDR MoMaS benchmark reactive transport. Centre National de la Recherche Scientifique; 2008.
- [60] de Dieuleveult C, Erhel J. A global approach to reactive transport: application to the MoMas benchmark. *Comput Geosci* 2010;14(3):451–64. <http://dx.doi.org/10.1007/s10596-009-9163-9>.
- [61] Hoffmann J, Krätzle S, Knabner P. A parallel global-implicit 2-D solver for reactive transport problems in porous media based on a reduction scheme and its application to the MoMaS benchmark problem. *Comput Geosci* 2010;14(3):421–33. <http://dx.doi.org/10.1007/s10596-009-9173-7>.
- [62] Wang Z, Giammar DE. Mass action expressions for bidentate adsorption in surface complexation modeling: theory and practice. *Environ Sci Technol* 2013;47(9):3982–96. <http://dx.doi.org/10.1021/es305180e>.
- [63] Wang Z, Giammar DE. Mass action expressions for bidentate adsorption in surface complexation modeling: theory and practice. *Environ Sci Technol* 2013;47(9):3982–96. <http://dx.doi.org/10.1021/es305180e>.
- [63] Wang Z, Giammar DE. Mass action expressions for bidentate adsorption in surface complexation modeling: theory and practice. *Environ Sci Technol* 2013;47(9):3982–96. <http://dx.doi.org/10.1021/es305180e>.
- [63] Wang Z, Giammar DE. Mass action expressions for bidentate adsorption in surface complexation modeling: theory and practice. *Environ Sci Technol* 2013;47(9):3982–96. <http://dx.doi.org/10.1021/es305180e>.
- [64] Fletcher CAJ. *Computational techniques for fluid dynamics*. New York: Springer-Verlag; 1991.
- [65] Amir L, Kern M. A global method for coupling transport with chemistry in heterogeneous porous media. *Comput Geosci* 2010;14(3):465–81. <http://dx.doi.org/10.1007/s10596-009-9162-x>.
- [66] Mayer KU, Frind EO, Blowes DW. Multicomponent reactive transport modeling in variably saturated porous media using a generalized formulation for kinetically controlled reactions. *Water Resour Res* 2002;38(9):131–1321. <http://dx.doi.org/10.1029/2001WR000862>.
- [67] Steefel CI. *CrunchFlow, software for modeling, multicomponent reactive flow and transport: user's manual*. Lawrence Berkeley National Laboratory; 2006.
- [68] Mayer K.U. A numerical model for multicomponent reactive transport in variably saturated porous media. Ph.D. thesis. Canada: University of Waterloo; 1999.
- [69] Hoffmann J. *Results of the GdR MoMaS reactive transport benchmark with RICHY2D*. Institut für Angewandte Mathematik; 2008.
- [70] van der Lee J, De Windt L, Lagneau V, Goblet P. Presentation and application of the reactive transport code HYTEC. In: Majid S, Hassanizadeh RJSWGG, George FP, editors. *Developments in water science*. Elsevier; 2002. p. 599–606. [http://dx.doi.org/10.1016/S0167-5648\(02\)80114-9](http://dx.doi.org/10.1016/S0167-5648(02)80114-9).
- [71] Lagneau V, van der Lee J. HYTEC results of the MoMas reactive transport benchmark. *Comput Geosci* 2010;14(3):435–49. <http://dx.doi.org/10.1007/s10596-009-9159-5>.