ORIGINAL PAPER



A reactive transport simulator for variable porosity problems

Fabien Georget¹ · Jean H. Prévost¹ · Bruno Huet²

Received: 20 April 2015 / Accepted: 4 October 2016 © Springer International Publishing Switzerland 2016

Abstract ReactMiCP, a new reactive transport simulator was developed based on the semismooth speciation solver SpecMiCP. Its main feature is a sequential iterative operator splitting algorithm where macroscopic model parameters are explicitly included in the formulation. Its correctness, robustness, and efficiency are tested against the MoMaS benchmark and two sets of cement paste lab experiments. We show that a robust speciation solver is a key requirement for good performance of the reactive transport simulator. We also demonstrate that a sequential iterative solver should be preferred over non-iterative solvers when using operator splitting. The flexibility and the speed of the simulator are used to test the influence of the database, the initial condition, and the diffusion coefficient model for the cement paste simulations.

Keywords Reactive transport · Operator-Splitting · Upscaling · Cement · Benchmark

Fabien Georget fabieng@princeton.edu

> Jean H. Prévost prevost@princeton.edu

Bruno Huet bruno.m.huet@lafargeholcim.com

- ¹ Department of Civil and Environmental Engineering, Princeton University, Princeton, NJ08544, USA
- ² Lafarge Centre de Recherche, 95, rue du Montmurier, BP 70, 38291 Saint-Quentin Fallavier cedex, France

1 Introduction

Due to its wide range of applications, reactive transport has gained a lot of attention in the last decades. It has been studied and used numerous times, and many codes implement this method using various numerical strategies [8, 21, 27, 34, 36, 42]. A domain where reactive transport models are important is for the assessment of long-term durability properties of cementitious materials [2, 22, 34]. From every-day building to carbon capture and sequestration infrastructure, cement is ubiquitous. However, concrete is sensitive to its environment and it will undergo large-scale reactions if in contact with species such as CO₂ [2, 22], sulfates [32], chloride [24], or just pure water in large quantities [33, 44]. By tracking both the flux of reactants and products, and the transformation occurring inside the porous medium, reactive transport is an essential tool for understanding the mechanisms of these attacks and for developing mitigation strategies [2, 22, 33, 41].

The optimization field is also in continuous progression and, at the turn of the century, the complementarity problem and its variant were studied extensively (see, for example, the bibliography of Facchinei and Pang [14]). The complementarity problem gives us a mathematical framework to study equilibrium in many fields [14]. Using these newly developed tools, S. Kräutle and collaborators [21, 25] were the first to implement a reactive transport solver based on the complementarity condition to solve the equilibrium solid phase assemblage.

In a previous paper [17], we presented SpecMiCP, a new speciation solver based on the complementarity condition. Compared to the traditional trial and error process [4], this

formulation finds the solid phase assemblage more efficiently [17]. The speciation solver was developed to solve cement-related problems, where many solid phases can coexist at equilibrium, the water content is low and the ionic strength is high. The present article describes the work carried out to interface an enhanced version (see Appendix A) of this speciation solver with a reactive transport simulator.

Numerical methods are important but modelers need to trust their tools before using them. Therefore, developers need to test and demonstrate the correctness of their softwares. It is especially true for complex codes like reactive transport simulators. Benchmarking and comparison to simulations and experiments increase the confidence in the mathematical methods and their numerical implementations. In the present paper, we describe the tests used to develop and improve our simulator. In particular, we focus on the stability and robustness of the coupling process and the numerical benefits of our formulation and, of the semismooth speciation solver. The robust and efficient code is used to evaluate the sensitivity of the model to its input parameters (initial composition, initial porosity, diffusion coefficient law, ...). The second section explains the coupling algorithm used. Section 3 describes the system of mass balance equations solved in the examples ("saturated system"). Sections 4 and 5 present and discuss the simulations we used to test the solver.

2 Coupling algorithm

2.1 Component

Solving the mass conservation equations for every species in the system would be too expensive. To remedy this problem, the notion of component was introduced (see, for example, the reference books [4, 30]). In a chemical system at equilibrium, the chemical reactions for the set of species can be written as function of a finite set of component, the basis :

$$\{\mathbf{A}_j\} = \begin{bmatrix} \nu_{ij} \end{bmatrix} \{\mathbf{A}_i\} \qquad \qquad j = 1 \dots N_r, \quad i = 1 \dots N_c$$
(1)

where $[v_{ij}]$ is the matrix of stoichiometric coefficient and index *i* denotes a component. The size of the basis is N_c , and the number of species is N_r . Following the arguments in [4, 17], the components in this work are chosen as actual species. The total concentration of a component (C_i^{α}) in a phase is a linear combination of species concentrations (c_k^{α}) weighted by the stoichiometric coefficients.

$$C_i^{\alpha} = c_i^{\alpha} + \sum_j \nu_{ij} c_j^{\alpha} \tag{2}$$

2.2 Governing equations

We are interested in solving the following system of mass conservation equations for each component *i* in phase α :

$$\frac{\partial \mathscr{C}_i^{\alpha}}{\partial t} = \nabla \cdot \mathscr{F}_i^{\alpha} + \sum_{\beta}^{N_{\alpha}} \mathscr{R}_i^{\beta \to \alpha}$$
(3)

where \mathscr{C}_i^{α} is the total concentration of component i in phase α (per unit volume of material). \mathscr{F}_i^{α} is the total flux of component i in phase α . Several transport phenomena can be combined in this term, such as advection, dispersion, diffusion, $\ldots \mathscr{R}_i^{\beta \to \alpha}$ is a sink/source term due to the chemical reactions from phase β to phase α (hereafter called chemical exchange term). N_{α} is the number of phases. This term can describe precipitation and dissolution of solid phases, sorption, or evaporation, In the following work, it is assumed that coupling between phases and components are mainly due to the chemical exchange terms. In addition, the transport operator is the main factor driving the system out of equilibrium. The importance of these assumptions will be discussed in the next sections, when the coupling algorithm is presented.

2.3 Operator splitting

Solving the system of Eq. 3 is a complex task because of the coupling between phases and between components. The different mathematical properties of the operators is also a source of numerical difficulties. A common strategy is to apply the operator splitting method (e.g., [30, 50]). In this approach, each operator, or group of operators, is solved sequentially while considering the others fixed. This method is opposed to the global step approach where the entire system is solved all at once. Operator splitting offers a strong practical advantage. The best solver can be used for each operator. In addition, the flexibility of the simulator is greatly improved since it breaks the coupling between the equations.

In the operator splitting method for reactive transport, two sets of operators are usually defined [30, 50]: the transport operator $(\nabla \mathscr{F}_i^{\alpha})$ and the chemistry operator $(\sum_{\beta} \mathscr{R}_i^{\beta \to \alpha})$. We add a third operator: the upscaling operator. Its task is to solve the macroscopic parameters of the model. As for the transport and chemistry operators, the exact definition of the upscaling operator is problem dependent, but a common set of properties for each operator can be defined; they are presented in the next section.

2.4 Operator Staggers

Each operator is solved inside a stagger. A stagger is a selfcontained module solving a set of equations which defines the corresponding operator. Three staggers are defined: transport, chemistry, and upscaling.

2.4.1 Transport stagger

The transport stagger is in charge of solving the system of governing equations (3). During its execution, it assumes that the chemistry exchange terms are fixed. Since a transport stagger defines the governing equation of a system, it defines the main assumptions about the models. As a consequence, different chemistry staggers (e.g., equilibrium or kinetics controlled equations) and/or upscaling staggers (different models for the macroscopic models) can be attached to the same transport stagger. On the contrary, the same chemistry stagger should not be connected to different transport staggers since fundamental assumptions about the physical systems and the dominant phenomena vary.

Two transport staggers, i.e., two systems of equations, are currently implemented: a saturated advection-diffusion system and an unsaturated system. Only the saturated system is presented in this article for brevity (see Section 3), but it should be noted that the coupling algorithm is strictly the same for both cases. Other systems of governing equations could be implemented to broaden the scope of the simulator (e.g., three phases H_2O-CO_2 transport).

The transport stagger solves the problem using a finite element method. A new finite element solver was implemented using the techniques and the algorithms of Dynaflow [38]. Since we are interested in small 1D problems (~ 100 nodes), the implementation is relatively easy and straightforward. The main advantage of a new implementation over an existing code is the increased flexibility in the choice of variables and of the method to exchange data between the modules.

2.4.2 Chemistry stagger

The chemistry stagger solves the chemistry exchange terms for every node of the mesh. It assumes that the total concentrations are fixed at the macroscopic scale. One of the common task of the chemistry stagger is to solve the aqueous phase speciation. Solid phases precipitation and dissolution, as well as fluid phase separation can be added if the model requires it.

SpecMiCP [17] is the speciation solver used by the chemistry staggers implemented in ReactMiCP. The main features of this speciation solver are to formulate the solid phase equilibrium assemblage as a Mixed Complementarity problem and to solve the resulting problem using a semismooth Newton solver [17]. The robustness of this method contributes directly to the efficiency of our reactive transport simulator as it will be shown in the examples. Some modifications were necessary to the speciation solver

in order to optimize the coupling. They are described in Appendix A.

One of the consequences of the semismooth approach is that the basis of components is fixed during a computation. It has important repercussions on the implementation details. For example, the staggers share the same fixed basis. Thus, indexing the species, checking the consistency of the variables, and transferring data from the transport and chemistry staggers are easy tasks in our code. It removes the possibility of numerous hard-to-debug errors.

2.4.3 Upscaling stagger

In addition to the numerical coupling introduced in equation (3), there is a strong physical coupling between the transport and chemistry operators in variable porosity problems. Precipitation and dissolution of phases change the microstructure of the porous medium (in term of porosity, connectivity, tortuosity, ...). This transformation affects the macroscopic transport properties such as the diffusion coefficients or the permeabilities. Modification of the pore structure is usually modeled using hard-coded laws such as the Kozeny-Carman relationships for the intrinsic permeability or power law for the effective diffusion coefficient. Such laws are only theoretically valid under strong assumptions (monodisperse spherical particles, laminar flow, ...) and they are not generally applicable (e.g., [46]). Since we are interested in problems where transport fluxes are the main driving forces out of equilibrium, the quantitative abilities of our simulator are limited by the accuracy of the prediction of the transport parameters.

Improving the prediction of these parameters is only possible if the modeler can test and adapt its models. The solution we propose is to separate the upscaling stagger from the transport stagger. The user defines a customized upscaling stagger which computes the necessary information and provides them to the other staggers. The definition of "necessary information" is presented in the next section on variables. This approach is possible due to the operator splitting approach. In practice, it works with an application programming interface (API) that the modeler uses to provide its data and functions to the algorithm.

2.5 Variables

Three sets of variables and one set of parameters are defined by system. The variables are separated by their roles and by the staggers used to find their values.

The main set of variables, called primary variables ({x}), is used to solve the governing equations. They are computed in both the transport and the chemistry staggers. The number of primary variables is $N_c \times N_{\alpha}$. However, some equations and/or variables may be trivial. For example, a component may not exist in gas form, or solid phases can be considered impermeable, thus making the flux operator null in the mass balance equations of the solid phases.

The secondary variables $(\{y\})$ are variables dependent on the primary variables, and they are computed in the chemistry stagger. The total concentrations (\mathscr{C}_i^{α}) can be written as a combination of primary and secondary variables. For example, the total concentration of water in the liquid phase per amount volume of material is $\mathscr{C}_w^l = \phi S_l C_w$. The liquid saturation (S_l) is used as a primary variable, and the secondary variables are the porosity (ϕ) and the total concentration of water in the liquid phase (C_w) . The gaseous total concentration for a component who evaporates as an ideal gas is $\mathscr{C}_{i}^{g} = \phi(1 - S_{l})\frac{p_{i}}{RT}$ where the partial pressure (p_i) is a primary variable. The choice of primary and secondary variables is dependent upon the model solved. Solving the secondary variables consistently is important, in particular in variable porosity problem. It was demonstrated for the porosity by Lagneau et al. [27].

The set of tertiary variables is composed of the concentrations, volume fractions, fugacities, ..., of the chemical species. These tertiary variables are computed in the chemistry stagger. They are used to obtain the primary and secondary variables in the chemistry stagger. Keeping these values in memory is important to warmstart the speciation solver [17].

To close the system, we also need the upscaling parameters ($\{w\}$). They are solved in the upscaling stagger. They are the macroscopic parameters related to the microstructure and appearing in the transport and chemistry operators. The diffusion coefficients, intrinsic, and relative permeabilities, capillary pressure, specific surface area are all examples of such parameters. These parameters are computed inside the upscaling stagger. Depending on the model, these parameters can be constants, variables, or function of the primary variables. For example, the relative permeabilities and the capillary pressure are function of the liquid saturation in the unsaturated system. Although not presented in this paper, this feature is crucial to model the coupling between reactive transport and drying of cementitious material [16, 20]. The set of parameters and their dependence needed for a simulation are defined by the governing equations and thus by the transport stagger.

2.6 Residuals based formulation

2.6.1 Operator splitting

Operator splitting exists in two main flavors, sequential noniterative algorithms (SNIA) and sequential iterative algorithms (SIA) (e.g., see the reference book [30]). As the name suggests, SIA iterate over the operators until convergence is obtained.

The main cited advantage of the non-iterative algorithms is the simplicity of their implementations [6, 30]. However, they lack the robustness of the iterative algorithms. They usually require smaller timesteps and finer meshes to be numerically stable and accurate [8]. In addition, the integration of adaptive timestep algorithm is harder since a non-obvious metric must be defined to track the success of the algorithm. These downsides are especially important in our case since we want to allow user-defined models. The solver needs to provide a stronger guarantee of stability. In addition, it will be demonstrated using the MoMaS benchmark that fixed coefficient problems also benefit from iterative algorithms (see Section 4.3.2). As a consequence, a sequential iterative algorithm was chosen. It should be noted that a non-iterative algorithm can be trivially obtained by stopping prematurely the iterative algorithm after the first iteration.

2.6.2 Convergence criterion

The convergence criterion for any iterative algorithm is crucial for the robustness of the method. We adopted a residual-based formulation using fixed-point iterations (see Fig. 1. The residuals of the weak form of Eq. 3 over an element is

$$\|R_{i}^{\alpha}\| = \int_{\Omega} \left(\frac{\partial \mathscr{C}_{i}^{\alpha}}{\partial t} - \nabla \cdot \mathscr{F}_{i}^{\alpha} - \sum_{\beta} \mathscr{R}_{i}^{\beta \to \alpha}\right) w_{i} \,\mathrm{d}\Omega \qquad (4)$$

The choice of the test function w_i and the method to perform the integration will define the finite element method used to solve the transport operator (e.g., Galerkin, finite volume, ...). Since the residuals contain all the information about the system (transport, chemistry, and upscaling variables/parameters are present), they provide a good metric about the state of the solution. The goal of the coupling



Fig. 1 Sequential iterative algorithm of ReactMiCP

algorithm, and of the transport stagger, is to minimize the residuals.

We also define an initial residual to provide a reference value :

$$\|R_i^{\alpha}\|^0 = \int_{\Omega} \left(\nabla \cdot \mathscr{F}_i^{\alpha}\right)^0 w_i \,\mathrm{d}\Omega \tag{5}$$

This value is computed at the beginning of the timestep. The system is considered solved when

$$\|R_i^{\alpha}\| \le \operatorname{tol} \|R_i^{\alpha}\|^0 \tag{6}$$

where tol is a user-defined tolerance. The initial residual is defined to represent the main perturbation driving the system out of equilibrium. Equation 6 means that the system is solved when the error on the mass balance is much smaller than the initial perturbation. In practice, tol = 1×10^{-3} to 1×10^{-4} was found to offer a good compromise between accuracy and convergence speed. Since fixed-point iterations are used, the number of iterations increases steeply with stricter tolerances.

In our simulator, we assume that transport is the main perturbation phenomena. If a chemical reaction is responsible for driving the system out of equilibrium (e.g., isotope fractionation), it must be included in the initial residual and must be solved in the transport stagger.

The same residuals are used for the coupling algorithm and for the transport stagger. However, a different convergence criterion must be used. In particular, the stagger tolerances are much stricter. High precision is needed in each stagger in order to reach convergence of the coupling algorithm in a reasonable number of iterations.

A more common convergence condition is the minimization of the difference between the total concentrations given by the transport and the chemistry staggers [7, 43]. Our residual formulation is governed by our goal of a flexible algorithm. The impact of the upscaling stagger can be captured by the residuals independently of the model provided by the user, and the variables are always consistent. Our formulation also directly minimizes the mass balance error. However, computing the residuals requires an implicit formulation. This is not a restriction for our applications.

Although the residuals are the main metric for convergence, other criteria must be introduced for limit cases (e.g., when steady-state is reached). The complete algorithm is presented in the code manual (see Appendix B).

2.6.3 Updating the variables

A predictor-multipredictor method is used to update consistently the primary and secondary variables through the staggers. The scope of each stagger is well defined due to the operator splitting approach. The transport stagger only modifies the primary variables. The chemistry stagger modifies both the primary and secondary variables. It also sets the tertiary variables. The upscaling stagger only modifies the upscaling parameters.

At the beginning of the timestep, the predictors are saved from the initial condition $\{z\}^{n,t+\Delta t} = \{z\}^t$, where z is a primary or secondary variables (z = x, y). When the rate of variation is computed in a stagger (e.g., transport stagger and kinetic reactions), the variables are updated using the predictors :

$$\{z\}^{n,t+\Delta t} = \{z\}^{0,t+\Delta t} + \Delta t \cdot \{\dot{z}\}^{n,t+\Delta t}$$
(7)

where n is current iteration. When the variables are solved directly (e.g., chemical equilibrium), the rate of variation is updated using the backward scheme :

$$\{\dot{z}\}^{n,t+\Delta t} = \frac{\{z\}^{n,t+\Delta t} - \{z\}^{0,t+\Delta t}}{\Delta t}$$
(8)

In this way, the information is complete and consistent at every step. For each variables, at least three values must be stored by nodes (z, z_0, \dot{z}) . In addition, the tertiary variables, the upscaling parameters, fluxes, and chemistry exchange terms must also be stored. However, as noted before, we are interested in small 1D problems, so the memory requirement is not limiting. The highest resident set size measured during a computation was 14 Mb (unsaturated system, 9 components, 19 solid phases, and 100 nodes).

2.6.4 Iterations

A timestep is solved by performing fixed-point iterations over every stagger, as shown in Fig. 1. First, the governing (3) is solved using the transport stagger, assuming that the chemical exchange terms are constant. Then, the chemical exchange terms are computed by the chemistry stagger. Finally, the upscaling parameters are updated in the upscaling stagger. The last step is to check the convergence. The residuals needed for this step are computed by the transport stagger. This scheme works because the residuals are minimized in the first step. Then the solution is perturbed in the chemistry and upscaling stagger. If the perturbation is small enough then the solution is still good enough at the end of the iteration and, the residuals respect the convergence condition (6). The meaning of "good enough" is defined by the tolerances chosen by the modeler. Also, since the weak form is used, only one implementation is required for the global convergence of the coupling, and the local convergence in the transport solver. Since a finite difference algorithm is used to compute the Jacobian, the governing equations are implemented only once per system.

The initialization of the iterations is necessary for the robustness and efficiency of the method (e.g., [27]). However, it is heavily dependent on the problem solved. For example, using relaxed values from the previous timestep to initialize, the system is useful to avoid negative concentrations. Many failures can be avoided if this algorithm is correctly implemented. The upscaling stagger has access to the initialization step, thus the user can apply specific rules for his model. Readers interested in the implementation can refer to the code for further details (see Appendix B).

2.6.5 Adaptive timestep

In variable porosity problems, the coupling is highly dynamic. The stiffness of the numerical and physical coupling varies with time. For better or worse, it results in a change of the amount of work required to reach a given tolerance. In nonlinear problems, especially with a userdefined upscaling stagger, it is difficult to estimate the required work and to adjust the parameters manually. A method must be found to let the solver regulates itself. A common solution is to use an adaptive timestep algorithm to select an appropriate timestep duration [26, 36]. To drive these algorithms, a metric is required to measure the performance of the coupling algorithm. For SIA, the obvious choice is the number of fixed-point iterations needed to reach convergence. If the coupling strength increases, so does the number of iterations. The adaptive timestep algorithm tracks the number of iterations, and it keeps them in a given range. If the upper limit is reached, the timestep is decreased. Conversely, if the lower limit is reached the timestep is increased. In the tests and examples presented below, the range [1, 15] was found to work quite well. To track the number of iterations, an exponential moving average is used. A single value keeps the history of previous timesteps while smoothing out accidents. Since the rate of variations (\dot{x}) , rather than the variables themselves (x), is used to drive the algorithms no special procedure is needed to initialize the system when the timestep is changed.

2.6.6 Restart procedure

Since the values at the beginning of the timestep are saved in the predictors, the system can be easily restarted in case of failures. Although failures should be avoided if possible, it might still happen if the conditions in the system change abruptly (e.g., if the solid phase assemblage change at the reaction front). In this case, a new timestep duration is chosen (smaller or bigger, depending on the error). The system is cleaned by resetting the values of variables, setting the chemical exchange terms to zero, and recomputing the upscaling parameters. Then, a new timestep is started.

2.6.7 Parallel computations

Modern computers have multiple CPUs, and efficient algorithms should use them as much as possible. Operator splitting algorithms for reactive transport are well-known for their abilities to be parallelized since the chemistry problems for each nodes are independent [30]. The strategy adopted should depend on the dimensionality of the problem. In our case, the main targets are small 1D problems (~ 100 nodes). Data transfer latency is a major bottleneck for parallel algorithms. For these reasons, we constrain ourselves to a single machine. Therefore, the shared-memory multiprocessing OpenMP API [10] was chosen to implement the parallelization. It allows to perform tasks in parallel in just a few lines of code, without perturbing the logic of the main algorithms and it leaves the implementation details to the compiler. As a consequence, running tests to find the best parallelization strategies are easy. The impact of the parallel computations is described in the examples.

3 Presentation of the saturated system

The formulation presented in the preceding section is general and can be adapted to many cases describing advection, diffusion, dispersion, ... in one or multiphase systems governed by chemical equilibrium or by kinetics. However, calibration and/or validation is only relevant when targeting a specific case. Therefore, we focus on the case of the saturated system. This system models the advection-diffusion of aqueous components in a saturated porous medium. The governing transport equation for the aqueous component i is :

$$\frac{\partial \phi \tilde{C}_i}{\partial t} = -\nabla \cdot \left(\tilde{C}_i \underline{q} - D \nabla \tilde{C}_i \right) - \frac{\partial \bar{\mathscr{C}}_i}{\partial t}$$
(9)

The primary variables for component *i* are (1) \tilde{C}_i , the aqueous concentration (per unit volume of solution), and (2) \mathscr{C}_i the immobile concentration (per unit volume of porous medium). The immobile concentration is composed of both the sorbed species and the solid phases. The only secondary variable is the porosity (ϕ) . The pressure equation of the liquid phase is not solved. The transport equations for the immobile concentrations are trivial, but the immobile concentrations should still be considered in the stopping criteria. In this system, it is needed to distinguish between apparent equilibrium, due to the chemical buffering, and true steady-state. The system of governing equations is solved using a vertex-centered finite volume method [29, 38] with the upwinding scheme to solve the advection operator. A chemistry stagger for this system must compute the aqueous speciation, the sorption of species and the precipitation/dissolution of solid phases. Only chemical reactions at equilibrium are considered in this article. By identifying Eqs. (9) and (3), the chemistry exchange term for component i is:

$$\mathscr{R}_{i}^{s \to l} = -\frac{\partial \tilde{\mathscr{C}_{i}}}{\partial t} \approx -\frac{\tilde{\mathscr{C}_{i}^{t+\Delta t}} - \tilde{\mathscr{C}_{i}^{t}}}{\Delta t}$$
(10)

The upscaling parameters are the Darcy's velocity (\underline{q}) and the effective diffusion coefficient (D). Any upscaling stagger for this system must define these two parameters. The diffusion coefficient is assumed to be the same for all species.

A better classification for the Darcy's velocity would be as a simulation input parameter since we do not solve explicitly the pressure equation. However, including it in the upscaling stagger allows space and time dependence without any supplementary cost in the implementation. In addition, since both parameters are set in the upscaling stagger, the diffusion term can model dispersion too. In this case, the equivalent effective diffusion coefficient is a function of the Darcy's velocity and the dispersivity α , $D = \alpha |q|$. The effective diffusion coefficient includes all effect of the porosity. The more common formulation $\nabla \left(\phi \tilde{D} \nabla \tilde{C} \right)$ is not used in this work for two main reasons: (1) our formulation matches the common empirical laws used in the cement and concrete community and (2) the modeler can distinguish between the different porosities, for example, the connected and non-connected porosities, in the upscaling stagger.

This system is used in the examples presented in the next two sections.

4 MoMaS reactive transport benchmark

4.1 Context

The MoMaS research group developed a reactive transport benchmark to compare the performance and accuracy of reactive transport simulators [9]. We use this benchmark to demonstrate the correctness and efficiency of our algorithm. Although each test cases tried are presented here, we refer the readers to the original publication [9] for the full description of the benchmark. The results are presented in Carrayrou et al. [8] and in the following individual contributions [6, 21, 26, 36]. The solutions are compared to a reference solution found by increasing the number of nodes and decreasing the timestep until convergence [6]. No analytical solution nor experimental data is available.

Three cases, named *easy, medium*, and *hard* cases, exist in the benchmark. Only results from the 1D *easy* test case are presented in this paper. Since this benchmark is artificial and cannot be compared to analytical solution or experimental data, we preferred to present real test cases to show the properties of our simulator in real-life applications. These effects and their importance are discussed in the next section. In this section, we first provide a presentation of the benchmark then the results for the two sub-cases are discussed.

4.2 The benchmark

The benchmark is unitless. The 1D *easy* test case includes four components (X_1 to X_4), one sorption site (S), five secondary aqueous species (C_1 to C_5), and two sorbed species (CS_1 and CS_2). The sample of length L = 2.1 is initially at equilibrium with a solution of X_2 and X_4 . At time t = 0, a solution containing X_1 , X_2 , and X_3 is injected at x = 0. At time t = 5000, the system is flushed with a solution containing X_2 and X_4 . The sample is inhomogeneous. Between x = 1 and x=1.1, the material is more porous ($\phi = 0.5$ instead of $\phi = 0.25$) but less permeable ($K = 10^{-5}$ compared to 10^{-2}). The concentration of surface sorption site is also greater (S = 10 instead of S = 1).

In addition to the advective flux, dispersion is also considered. In the 1D problem, the flow field is not modified during the experiment. Therefore, the dispersivity coefficient is constant with time, and the dispersivity term is mathematically equivalent to a diffusion term. As a consequence, our saturated system is equivalent to the one presented in [9]. The benchmark can be modeled with our saturated system and its equilibrium chemistry stagger. For the advective case, the dispersivity between x = 1 and x=1.1 is 6×10^{-2} , and it is 1×10^{-2} elsewhere.

Although this exercise is artificial and called *easy*, it presents many of the difficulties of real problems such as appearing or disappearing components (due to the injection and flushing fluxes), inhomogeneous mesh, large range of equilibrium constant... However, the ionic strength model and the solid phase equilibrium are not tested. In these tests, it is especially interesting to compare our performance to HYTEC [26], the other code of the benchmark based on iterative operator splitting, and SPECY [6] which uses a non-iterative operator splitting algorithm. The other codes use a global-step approach [8].

For a better comparison, a normalization method for the CPU time was introduced [9]. The normalization process involves a matrix-matrix multiplication. Many methods exist to implement the multiplication. Given the reference value of 12.3 s, the straightforward way was assumed. On the two computers used during the tests, the results are (1) 8.3 s (Intel® CoreTM i7-3612QM (8 cores) used for development) and (2) 8.4 s (Intel® Xeon® CPU X5550 (8 cores) used for the long tests). When the normalization was implemented using the Eigen library [19], this time drops to approximately 0.2 s and drops further to 0.06 s when OpenMP [10] is enabled. It shows that the first step to optimize a code is to use a powerful linear algebra library which can use the features of modern CPUs. ReactMiCP is implemented using Eigen [19].

Two sub-cases are defined depending on the relative values of the advective and dispersive fluxes. The next subsections present the results for both cases.

4.3 Advective case

4.3.1 Accuracy

As suggested by the name, advection is the dominant transport phenomena in this case. The solution is characterized by a relative sharp reaction front moving through the mesh. In the original benchmark, the accuracy of the simulators is tested using the concentration of free sorption sites at time t = 10.0 [9]. The Figs. 2 and 3 present a comparison between our results and the reference solution provided. Our solution is similar to the reference solution and in the range of the solutions presented in [8, Fig. 5].

Accuracy is quantified by the amplitude and position of the first peak of free sorption site (Fig. 3). This peak is delayed from the advection front due to the chemical equilibrium buffering. In our tests, for $\Delta x = 0.001$ the first peak is at x = 0.0175, as close as possible to the reference value of x = 0.0174 [8, Table 4] due to the resolution. However the amplitude is lower (0.692 instead of the expected value of 1.0). Most implicit numerical scheme to discretize the



Fig. 2 Concentration profiles of solid component S at time 10 for the MoMaS benchmark 1D easy advective test case. Reference curve is from Carrayrou et al. $\Delta x = 0.002$ [8]



Fig. 3 Concentration profiles of solid component S at time 10 for the MoMaS benchmark 1D easy advective test case. Zoom on the first peak. Reference curve is from Carrayrou et al. $\Delta x = 0.002$ [8]

advection operator introduce numerical diffusion ([29]). A common solution is to use explicit high order methods (e.g. [28, 29] and to be at the optimum timestep. This optimum timestep is given by the Courant-Friedrich-Levy criterion (CFL) [29] :

$$CFL = \frac{u\Delta t}{\Delta x}$$
(11)

where *u* is the pore velocity. The optimum timestep is for CFL = 1.0. However due to the coupling with the chemistry operator, the optimum timestep for the advection operator does not lead to a stable scheme and a lower timestep must be used. The reference solution was computed with CFL = 1×10^{-2} [6].

4.3.2 SIA / SNIA comparison

Since our simulator uses the predictor-multicorrector method, we are able to choose between an explicit and implicit scheme, and an iterative or non-iterative algorithm.¹ Figures 4 and 5 present the results of these simulations. Figure 4 shows the amplitude of the first peak as function of the CFL criterion (For $\Delta x = 0.001$). Figure 5 presents the normalized CPU time for the same simulations. The explicit point corresponds to the maximum timestep for which the system is stable in our formulation (CFL = 1.1×10^{-2}). Using such a small timestep leads to a high execution time as shown in Fig. 5 (× 4.2 the CPU time needed by the

¹SIA is only available with the implicit scheme.

Fig. 4 Amplitude of the first peak of the concentration of free sorption site at t = 10 as function of the CFL criteria for the MoMaS benchmark 1D easy advective test case. S(N)IA: sequential (non-)iterative algorithm. $\Delta x = 0.001$



adaptative timestep algorithm). A longer timestep can be used with the implicit solver. Thus, it decreases the overall amount of work needed to solve the system. As the timestep increases, the SNIA solutions requires less and less time. However, the amplitude is not stable and it increases with the timestep. Although the solution becomes closer to the reference solution, we interpret this variation as instability. A good numerical solution should be as invariant as possible of the timestep. This is a requirement for the implementation of an adaptive timestep algorithm. SIA offers this guarantee as seen on Fig. 4. It should be noted that even with the implicit scheme, the maximum timestep is such that CFL < 1.0. Higher timestep is usually not numerically stable.

Figure 5 also demonstrates that no cheating is possible. The same amount of work is required for all timestep durations to obtain the same quality of solution. The amount of work is governed by the stiffness of the coupling, which is related to the disequilibrium caused by the transport. For the iterative algorithm, increasing the timestep implies



Fig. 5 Normalized CPU time until t = 10 as function of the CFL criteria for the MoMaS benchmark 1D easy advective test case. $\Delta x = 0.001$.CFL = 1 corresponds to $\Delta t = 0.0455$

increasing the number of fixed-point iteration. This relation leads to a stable total execution time with respect to the timestep for the iterative algorithm. Figures 4 and 5 form a very strong argument in favor of using an iterative sequential algorithm instead of the non-iterative approach. Only a SIA solver guarantees a consistent solution across all nodes and all timesteps. In the next examples, it is the only solver used.

In addition, it should be noted that the amplitude of the peak depends strongly on the space discretization. The results presented are for $\Delta x = 0.001$. Figure 3 presents the results for $\Delta x = 0.002$ and the amplitude of the peak is only 0.495. Due to the lack of analytical solution, or experiments, we can only assert that our results are sufficiently close to the solutions obtained with the other simulators.

4.3.3 Efficiency

Obtaining a good answer is the first step, but obtaining it quickly is better. Figure 6 shows the normalized CPU time of ReactMiCP compared to the results presented by Carrayrou et al. [8]. The execution time from ReactMiCP is in the range of the other codes. In particular, it is faster than HYTEC [26] and similar to MIN3P [36]. The slope is also similar to the other code : CPUtime $\propto N_{cell}^{2.30}$. In the benchmark [8], these slopes range from 1.81 (GDAE1D) to 2.62 (HYTEC). These values are an indication of the complexity of each coupling algorithm and the relative complexity of the transport and chemistry solvers. The quadratic behavior prohibits very fine mesh and highlights the need for a solver to be able to solve accurately coarser meshes. This is another argument for using an iterative solver. Figure 6 also displays the effect of the parallelization. Parallel computing is implemented by some of the other reactive transport solvers, but this feature was disabled for the benchmark. Therefore, only ReactMiCP results are provided. Without OpenMP, 67 % is spent in the chemistry stagger and 30 % in the transport stagger. When OpenMP is enabled, the trend is inverted: 28 % for the chemistry and 68 % for the transport. Due to the importance of the serial transport stagger, the speed up is relatively small. This is not a common case, the chemistry stagger is usually more expensive, even where OpenMP is enabled, so the parallelization becomes more efficient.

4.4 Dispersive case

In the dispersive case, the magnitude of the dispersion is increased by a factor 1000, making dispersion the dominant transport phenomena [9]. Operator splitting methods are supposed to be less efficient for this test case according the original comparison [8]. Since the dispersivity is high, more nodes are perturbed. Therefore, the coupling is stiffer and more iterations are required (e.g., see [Fig. 15] [26]).

However, due to the care taken on the chemistry solver and its integration, our implementation proved to be quite efficient as shown by Figs. 7 (accuracy) and 8 (execution time). Figure 7 presents the elution curve of species C2 as function of time, as presented in the original publication [8]. SPECY uses an explicit SNIA algorithm and therefore do not control the error as seen on Fig. 7. As a consequence, small timesteps and a fine mesh are required to keep the error sufficiently small, leading to high execution

Fig. 6 Normalized computing times as a function of discretization for the MoMaS benchmark 1D easy advective test case. Results of the other codes are from Carrayrou et al. [9]



Fig. 7 Elution curve for species C_2 at x = 2.1 for the MoMaS benchmark 1D easy diffusive test case. Results of the other codes are from Carrayrou et al. [9]



times. Also, the fixed timestep does not adapt to the different characteristic times of the injection and leaching steps. The difference between the execution time of HYTEC and ReactMiCP highlights the need for a very efficient chemistry solver. Indeed, the chemistry solver has a large impact on the simulation. In this test case, the solver spends in average 80 % of its time in the chemistry stagger (40 % when OpenMP is enabled). Therefore, it is important to keep the overhead as small as possible. This is why ReactMiCP is built has a collection of modules that can be customized and interfaced together in a very efficient way. The slope of the normalized CPU time with respect to the log of the number of cells is similar to the other codes and the advective case, with CPU time $\propto N_{\rm cell}^{2.17}$ (Fig. 8).

Using the MoMas benchmark, we demonstrated the correctness and efficiency of ReactMiCP for this special case. However, these problems are artificial, and we need to guarantee these properties on real life problems.





5 Cement paste simulations of laboratory experiments

In this section, we present some simulation results relevant to cement pastes. We use previous experimental works to validate our simulator and test its performance and sensitivity.

The thermodynamic database used in this section is the CEMDATA'07 database [31, 35] developed for cement applications. It is based on the Nagra-PSI database for the aqueous species [23]. No solid solution model is included in SpecMiCP; therefore, only the end-member compositions are taken into account. The basis used in the computations is H2O, HO⁻, Ca²⁺, SiO(OH)₃⁻, Al(OH)₄⁻, Fe(OH)₄⁻, Cl⁻, K⁺, and Na⁺.

5.1 Leaching in pure water

5.1.1 Presentation of the problem

Leaching is an experiment where a sample of cement paste is dissolved in a bath of undersaturated water [15, 33]. The solid phases are dissolved progressively, increasing the porosity in the leached area and leading to further damages in the sample [15, 33]. This is one of the simplest simulation for cement paste since no new component is introduced to the system.

In this section, we use the data from Mainguy et al. [33]. In this paper, the authors present both experimental data and a simplified reactive transport model where only the amount of calcium is taken into account, as a proxy for the entire state of the material. They obtain a good agreement between the model and their data [33]. The objective of this section is to reproduce both the experimental and the simulation results with a complete reactive transport model. Although more computationally expensive, a complete model provides more information especially about the chemistry. Such insight is important when distinguishing mechanisms and when extrapolating experimental data.

The cement paste samples were prepared from OPC cement with a water to cement ratio of 0.4 [33]. They were cured under lime-saturated water for 6 months. Then the samples were immersed in water maintained at pH 7 by addition of nitric acid. The total loss of calcium were recorded [33, Fig. 2]. Reaction fronts propagates into the samples [33, Fig. 1]. These fronts are caused by the dissolution of cement paste in contact with the acid solution.

The initial conditions used in the simulation were obtained by hydrating the OPC oxide composition given in [33, Table 1]. Since the samples were cured, complete hydration is assumed. Then, the volume fraction of solids were scaled in order to obtain a total concentration in calcium equal to the reference value of 15,000 mol.m⁻³. The

final porosity of the system is $\phi = 29.5$ %, higher than the reference value of 25 %. The molar volumes used to compute this value are those given by the CEMDATA'07 database [31]. We note that this value is the total porosity including the non-connected porosity. The correct determination of the transport properties should include a correction to only take into account the effective porosity [3]. A 1D axisymmetric mesh is used to represent the experimental samples of cylindrical shape (diameter d = 7 cm). The pH at the boundary condition is fixed at 7 to simulate the experimental conditions.

The advection velocity is set to 0. The effective diffusion coefficient is given by the following exponential law [33] :

$$D_e = \exp(9.95\phi - 29.08) \tag{12}$$

if $\phi < 0.92$, else $D_e = 2.2 \times 10^{-9} \,\mathrm{m}^2 \,\mathrm{s}^{-1}$, the diffusion coefficient in water [33]. In practice, the porosity is not higher than 0.6 in the original publication since it is assumed that the amorphous silica gel does not dissolve in the timeframe of the experiment [33]. As shown by Dove et al. [12], the kinetics of dissolution of the silica gel are slower than expected and it still exists even if the thermodynamic model predicts its dissolution. In the profiles at 25 days (9), we see that the pH is too high to allow for the precipitation of the silica gel. In the experiment, the pH was maintained at neutrality with the addition of nitric acid. Without the amorphous silica gel, the total loss of calcium is 2.5 times higher than the experiments. To model the gel, the coefficient of diffusion is taken as the minimum between the exponential law and $1 \times 10^{-10} \text{ m}^2 \text{ s}^{-1}$. This value is close to the maximum value allowed in the original model for a porosity of 0.6 [33]. To force the precipitation of silica gel, silicates may be added in the boundary conditions. However, in our simulations, these boundary conditions lead to the clogging of the pores due to the mass precipitation of tobermorite at the surface of the sample.

5.1.2 Results

Figure 9 presents a typical profile of solid phases obtained by our simulator. The profile is a succession of reaction fronts where the solid phases dissolve one after the other. The first phase to dissolve is the portlandite. The pH is maintained high (>12) in the core of the sample but decreases in contact with the bath.

The solutions obtained (Fig. 10) are close to the value of the experiments and the model of Mainguy et al. [33]. This figure presents the cumulative flux of calcium lost by the sample. If this amount is closer to the experimental data than the simplified model, the degraded depth is lower at 90 days: 1.23 mm against the value of 1.45 mm for the experiment [33, Table 2]. The differences may be explained by the

Fig. 9 Solid phase and pH profiles for the leaching problem at 25 days. Maximum space discretization is $\Delta x = 25 \,\mu$ m. Results obtained with a non-uniform mesh, crosses on the pH curve indicate the position of the nodes



different relationships between the total aqueous concentration of calcium, the total solid concentration of calcium, and the porosity. It is investigated further below.

The short-term behavior is highly dependent on the mesh size. Figure 11 shows the total loss of calcium for different space discretizations. A very fine mesh ($\Delta x = 5 \,\mu$ m) is required to study the short term behavior. From the physical point of view, it may be smaller than the representative elementary volume. For the coarser mesh ($\Delta x = 50 \,\mu$ m), the time to obtain a linear slope is the time to dissolve the portlandite in the first cell. The complete dissolution of portlandite induces a decrease of the total aqueous concentration

of calcium in the first cell, and it creates a gradient of concentration of calcium inside the material. when the first node is depleted of calcium, a pseudo-stationary flux is obtained and its amount is governed by the damaged area of the sample. As a consequence, after the dissolution of the first cells, the slope of the leaching curve are similar for all simulations, about $1.15 \text{ molm}^{-1} \text{ s}^{-1/2}$. This value is mainly dependant on the diffusion coefficient Eq. 10. Finer meshes require more work and lead to many numerical difficulties. One way to solve the problem is to use a non-uniform mesh, with finer cells at the surface of the material and larger cells at the core of the samples. The results for such a mesh

Fig. 10 Leaching curve as function of the upscaling law. Experiments data is from Mainguy et al. [33]. Inset layer represents the coefficient of diffusion as function of the porosity. The mesh size was $\Delta x = 10 \,\mu m$







is presented in Fig. 11. The non-uniform mesh starts with cells of 5 μ m and end with cells up to 50 μ m. The solution obtained with this mesh is less accurate than the 5 μ m mesh but is still acceptable and is 32 times faster (see Table 1 and Fig. 11). To take full advantage of the non-uniform mesh an adaptive timestep algorithm is required, otherwise, the timestep would be governed by the smallest cell for the entire duration of the computation.

5.1.3 Influence of the simulation parameters

Table 2 tabulates the sensitivity of the solution to the initial condition. When the solid concentration of calcium is lower, the porosity is higher. Therefore, the coefficient of diffusion and the leaching flux are higher. This is in partial disagreement with the results presented by Mainguy et al. [33, Table 4] because they considered that the parameters where independent in their sensitivity analysis, so the porosity had no

Table 1 CPU time for the leaching simulation until t = 25 days as function of the mesh discretization. Non-uniform mesh contains cells from 5 to 50 μ m

Cell size (µm)	CPU time (s)	
50	28	
25	108	(×3.9)
10	440	(×16)
5	2149	(×76)
Non-uniform	67	(×2.4)

Corresponding leaching curve are presented in Fig. 11

feedback on the concentration of calcium. In their study, the most influencing parameter is the porosity, due to the diffusion coefficient exponential law. The effect of the diffusion law can be easily tested due to our formulation. We replaced this law with the common power law

$$D_e = D_{e,0} \left(\frac{\phi - \phi_r}{\phi_0 - \phi_r}\right)^m \tag{13}$$

where $\phi_0 = 0.25$, $\phi_r = 0.02$, and $D_{e,0} = 2.76 \times 10^{-12} \text{ m}^2 \text{ s}^{-1}$ is the value given by the exponential law for $\phi = \phi_0$. The results are presented in Fig. 10. The exponential law and the power law with m = 3.32 give very similar answers. The value of 3.32 was chosen to be close to the first law and to agree with the next section and Huet et al. [22]. The two laws are very similar in the range 0.25–0.5. For higher porosity, they are also similar since the effective diffusion coefficient is capped at $D_e = 1 \times 10^{-10} \text{ m}^{-2} \text{ s}^{-1}$. When no maximum is imposed for the power law, the

 Table 2
 Leaching : flux of calcium for different initial solid concentration of calcium

Calcium concentration (mol.m ⁻³)		Initial porosity	Calcium molm ⁻²	Calcium flux molm ⁻² d ^{-1/2}	
15000		29.5	1.095		
14500	-3.33 %	31.9	1.131	+3.29 %	
15500	+3.33 %	27.2	1.059	-3.29 %	
16000	+6.67 %	24.8	1.025	-6.39 %	

Results obtained with a mesh discretization of $\Delta x = 10 \,\mu\text{m}$

amount of calcium lost is higher, but not 2.5 times higher as the exponential law due to the shape of the power law. When the exponent m is higher, the flux of calcium is higher; when the exponent is lower the amount of calcium leached is lower. These variations are important and highlight the need for a representative effective diffusion coefficient model. In particular, the gel diffusivity is important since the diffusion through the amorphous gel is the driving phenomena for the leaching. It should be noted that using a single law for the three areas (silica gel, damaged paste, and core) is probably not representative of the changes in connectivity and pore shapes that occur during the experiment. Being able to incorporate complex upscaling laws to model the system is a step towards quantitative prediction; our three staggers approach are the steps toward reaching this goal.

This test shows that ReactMiCP is able to solve the leaching problem correctly, and it is robust with respect to the simulation parameters. The semismooth approach and the work done to optimize the solver allows us to be very efficient, solving 25 days in less than an hour, even for fine mesh. The conclusion to take out of this test is that the transport properties have the most impact on the solution, and it should be chosen carefully. These properties depend mainly on two independent variables, the total concentration of calcium (or the Ca/Si ratio) and the porosity [5]. Also, the mesh has to be very fine and requires a speciation solver which scales well.

5.2 Leaching in CO2-rich brines

5.2.1 Presentation of the problem

This set of simulation is similar to the previous one except for the inflow of CO_2 in the sample. CO_2 acidifies the pore solution. It also reacts with calcium to precipitates as calcite. The precipitation may result either in clogging of the pores or in accelerating the degradation process [40].

The simulations of this problem provide more demonstration of the correctness and efficiency of our code. In particular, it gives some results about the stability with respect to the chemistry modeling choices and a simple profiling, i.e., analysis of the performance, of the code.

Duguid and Scherer [13] presented experiments of leaching in CO₂-saturated water of cement paste samples. Huet et al. [22] developed a reactive transport model for one of this experiment, where the cement paste sample is subjected to sandstone-like condition, at pH= 3.7 (adjusted with H₂CO3) and T= 25°C, (T= 20°C in the experiments) [22]. The sample is a cylindrical shape of diameter d = 7 cm obtained by hydration of a class H cement paste [13]. The final porosity is $\phi = 0.4$. Huet et al. used a global step approach to solve the problem [22]. In this section, we run the same simulation with our simulator. The thermodynamic database and the effective diffusion coefficient law are the main differences in the previous and this set of simulations as further explained below.

Huet et al. [22] used a modified version of the EQ3/6 database [49] while we used the CEMDATA'07 database [31]. In our version, less secondary species are included and different solid phases may precipitate. No iron nor magnesium are included in the simulation at first. The iron is assimilated to the aluminum since it forms similar solid phases [11]. The effect of iron is further discussed below.

The effective diffusion coefficient law used by Huet et al. [22] is a standard power law (13) at low porosity. However, for $\phi = 0.7$, the diffusion coefficient is set to $n \times 2 \times 10^{-10} \text{ m}^2 \text{ s}^{-1}$ ($n \in [1, 2, 4]$). The main problem is that this law is not continuous and may lead to convergence problem, especially for an iterative operator splitting solver. Instead, we used the same upscaling law as for the pure water leaching simulation. The impact of the diffusion coefficient is discussed below and in the original publication [22].

5.2.2 Results

Figure 12 presents the profiles of solid fraction as function of the radius. The four layers defined in the experiments [13] and the simulations [22] are also visible here and are denoted by the roman numerals on the figure. The four layers are (I) non-damaged sample; (II) C-S-H layer, where the portlandite is dissolved; (III) carbonated layer, where calcite is the predominant phase; and (IV) an amorphous gel primarily composed of silica and aluminum. The silica gel does not span the entire length of the fourth layer unlike the previous results [22]. Adding silicates in the boundary condition can reduce the dissolution of amorphous silica at the outer edge of the sample while conserving a similar profile in the core [22]. The pH curve is also represented in Fig. 12. As expected the pH is high in the core of the sample but low in the carbonated and the gel layers. The drop occurs in zone (II) and (III) where the carbonation takes place. The total concentration of HO⁻ approximately follows a linear profile in the amorphous gel.

Figure 13 compares the diameter of these layers to the experimental values [13]. For this simulation, the initial porosity was matched to the initial experimental porosity of 47 %. In the original simulation, the porosity was 40 %; they respected the initial composition more closely [22]. The porosity and the total concentration in calcium are two independent variables [5] and a choice must be made. It was observed that the numerical solution is closer to the experiments when the porosity is matched. It is another indication that a better modeling of the microstructure is required and that our three-staggers operator splitting approach is relevant for these problems.





As in the leaching case, the front tracking is only approximate. If the gel and calcite layers are roughly represented, the dissolution of the portlandite is not correct. A better match can be obtained by modifying the upscaling law (see Fig. 15). The mismatch is probably not due to the kinetic of dissolution since the experimental fronts are sharp [13], indicating a local equilibrium. However, it may be due to the kinetics of precipitation of the new phases which may not be at equilibrium at $T=25^{\circ}$ [39]. It may also be due to an error of prediction of the relative volume fractions of the solid phases, in particular for the portlandite and the C-S-H. Another possible reason is that it is supposed that every component has the same diffusion coefficient. In practice, some species migrate faster. However, it is already quite difficult to fit one upscaling law, one upscaling law per component does not seem reasonable. Therefore, the

Fig. 13 Leaching in CO₂-saturated water simulation. Comparison of the simulated and experimental degradation depths for the saturated carbonation simulation. Results were obtained with $\Delta x = 25 \,\mu m$. Experimental values were initially reported by Duguid and Scherer [13] Initial porosity of 47 %. The *double arrows* present the extent of the layers in the experiment



Fig. 14 Leaching in CO₂-saturated water simulation. Effect of the components and solid phases included in the thermodynamic database for the saturated carbonation simulation. Profiles of the porosity and the pH at 30.0 days. Results were obtained with $\Delta x = 50 \,\mu m$ and $D_{e,max} = 1 \times 10^{-10} \,\mathrm{m^2 \, s^{-1}}$



discrepancy between the experiments and the simulation is more a modeling problem that a numerical issue and does not invalidate our solver.

5.2.3 Influence of the database

The database used in this work is different from the previous study. Therefore, the set of solid phases is different. To investigate the impact of this choice on the simulation results, we run a simulation where the precipitation of straetlingite were disabled and another simulation where the speciation of iron is included. Few perturbations is expected since aluminum and iron are minor species, and they form similar phases in cementitious materials [11]. Figure 14 shows the results of this experiment. The main conclusion is that the differences are very small and mainly located in area (II) (portlandite dissolution) and (III) (calcite layer) where the carbonation takes place. The iron does not introduce any major perturbation. Although the inclusion of iron seems to slightly delay the front, it should be noted that





the uncertainty linked to the diffusion coefficient law is higher. A change of the gel diffusivity has more impact on the carbonation depth has shown in Fig. 15. The removal of straetlingite is even less significant for a finer mesh. However, for very fine mesh, the precipitation of calcite, thaumasite, straetlingite, or ettringite may clog the porosity and stop the carbonation process. This was observed for a fine mesh ($\Delta x \leq 10 \ \mu m$). Disabling the precipitation of straetlingite and thaumasite only delay the apparition of this problem. No reference solution could be obtained, and the damaged depths are slightly dependent upon the mesh. However, once again the uncertainty of the diffusion coefficient has a higher impact than the mesh dependency. Nevertheless, this set of simulations demonstrates the numerical stability of our solver. It also demonstrates the relevance of our flexible simulator for cement pastes.

5.2.4 Profiling

For this problem, ReactMiCP is also quite efficient. It solves the problem in 12 to 18 min for 50 μ m cells (76 cells) and for a simulated time of 30 days. As a comparison, the global step solver [22] take approximately 6 h to solve the same problem with $\Delta x = 75 \mu$ m (50 cells). As before, a very efficient speciation solver makes the difference since the solver spent 60 (with OpenMP on 8 cores) to 95 (without OpenMP) % of the total execution time in the chemistry stagger (see Fig. 16).

To analyze the impact of the semismooth approach, a rudimentary profiling was run using Valgrind with the Callgrind tool [47]. The profiling was executed with a 50 μ m mesh and with OpenMP disabled. The semismooth reformulation is not as computationally expensive as it may seems. The computational power is mostly used to form and solve the chemistry system (60 % of the estimated CPU cycles). However, only 1.5 % of the computation is spent reformulating the Jacobian and the residuals. The aqueous solution non-ideality model is another important part of the computation (25 %). Simplified benchmarks such as the MoMaS benchmark may fail to represent accurately the relative performance of reactive transport simulators in reallife problems. The non-ideality model is expensive because it is called many times and includes many power and exponential computations. Modern hardware which provides more advanced instruction sets for floating point computations (e.g., AVX or AVX2) does provide a non negligible performance gain.

Indeed, the choice of the logarithm of the molalities as the main variables has a significant impact on the computational efficiency. Twenty-four percent of the estimated CPU cycles is spent calling the power and exponential functions. For comparison, 26 % of the CPU cycles is used to invert the chemistry system and 5 % is spent solving the transport problem. It is good that much of the computational power is spent on actual computation rather than on bookkeeping. Nonetheless, it highlights the cost of the logarithms of the molalities rather than the molalities themselves as main variables for the speciation solver. However, as discussed through this paper, robustness is an important part of efficiency. The logarithms allow the system to be more resilient to the large range of molalities. A basis where some components are not the dominant species can also be used. In the traditional algorithm, the basis is regularly changed and can be adapted to the problem [4, 37, 48]. With the complementarity approach, this is no longer required. However, in the course of a reactive transport simulation, the conditions



Fig. 16 Effect of the number of OpenMP threads on the execution time of the leaching in CO_2 -saturated water simulation. Left figure presents the percent of total execution time spent in each stagger. Right figure displays the total speedup

may change dramatically, and the initial basis may become sub-optimal. The logarithms and the scaling help to reduce the impact of a poor basis and remove the need for basis switching.

As stated earlier, using OpenMP accelerates the computation with a minimal impact on the code. Figure 16 presents the effect of the number of OpenMP threads on the computation for two meshes. As the number of threads increases, the percentage due to the transport stagger also increases, but the chemistry stagger is always more expensive (> 60 %). The speedup is significant but it is less than the theoretical prediction given by Amdhal's law [1]

Speedup(s) =
$$\frac{1}{(1-p) + \frac{p}{s}}$$
(14)

where *s* the number of threads and *p* is the percentage of time spent in the parallelizable section when s = 1. This is due to the bookkeeping introduced by OpenMP, and the fact that the chemistry problems are inhomogeneous since most of the difficulties are concentrated at the leaching front.

6 Summary and Conclusion

ReactMiCP, a reactive transport simulator, is introduced. Its main features are :

- 1. It is based on the operator splitting method, using a sequential iterative algorithm. The convergence is checked using the residuals of the governing equations compared to the initial perturbation.
- 2. Three staggers are defined: transport, chemistry, and upscaling. The operator splitting method separates the staggers into self-contained modules.
- 3. The transport stagger defines and solves the governing equation of the system. Different chemistry staggers and upscaling staggers can be used with the same transport stagger to adapt the simulation to the problem.
- 4. The chemistry stagger solves the chemical reactions driving the system back to local equilibrium. The implemented chemistry staggers use SpecMiCP, a semismooth speciation solver. We demonstrated that its efficiency and robustness properties are important in the context of reactive transport.
- The upscaling stagger computes the macroscopic parameters from a model of the microstructure. It can be customized to adapt the simulation to the user specific problem.

The correctness and efficiency of the algorithms and their implementation were demonstrated using the MoMaS benchmark and two sets of cement paste simulations. Our simulations are able to provide solutions similar to previous simulation [8, 22, 33] and/or to match the experimental data [13, 33]. Quantitative predictions are limited by our knowledge of the microstructure and available models and data. We believe that the flexibility offered by React-MiCP will give modelers new methods to investigate these problems and to develop new quantitative upscaling models. To test these models, efficient numerical tools are a key requirement. The performance of ReactMiCP on the MoMaS benchmark is good, but more importantly, real-life simulations of saturated cement paste leaching were solved in less than an hour for a simulated time of thirty days.

More results and discussions concerning cement paste leaching will be presented in a future paper [18]. In particular, the mechanisms will investigated, and more complex models are introduced making full use of the upscaling stagger.

Acknowledgments This work was supported by a grant from LafargeHolcim and CSTB, Scientific and Technical Center for Building, to Princeton university. This support is most gratefully acknowledged. The authors would like to thank Robert. J. Vanderbei and Hande Y. Benson for their helpful insights into the complementarity problem and the semismooth Newton method, and Jérôme Carrayrou for providing us the original results of the MoMaS benchmark.

A SpecMiCP enhancements

The first tests using SpecMiCP[17] in our reactive transport simulator highlighted some weaknesses due to the mathematical formulation of the problem. This section describes the modification made to SpecMiCP in order to resolve these problems and to improve the code. The main changes are the introduction of intensive main variables and the scaling of the main equations. In the process, gaseous species and sorbed species were added to the system. Also, modeling constraints such as fixed-fugacity gas, or saturated samples were implemented to allow more flexibility for the modeler. However, the main features of the solver remain mostly unchanged, and we refer the reader to the original publication for further information.

The first tests of the reactive transport solver revealed that the mesh should be very fine in order to obtain a good solution (e.g., see Section 5.1 and Fig. 11). The previous system used extensive variables: the mass of water and the mole number of solid phases. As a consequence, it scaled badly and failed for fine meshes. To remove this dependency to the mesh discretization, all primary variables were switched to intensive variables. In the new system, the main variables are the volume fraction of water (φ_w), the volume fractions of each solid phases (φ_m), and the logarithm of the molalities of each component $x_i = \log_{10}(b_i)$. The volume fractions are good variables since they are naturally contained between 0 and 1. The downside is that they

introduce physical parameters into the mass conservation equations, such as the density of liquid water (ρ_w) or the solid phase molar volumes (\bar{V}_m). It is a drawback since they may be difficult to characterize, especially for amorphous or poorly crystallized phases like C-S-H [45]. Nevertheless these parameters are already needed in a reactive transport code. They are used to compute the porosity and the effective transport properties of the porous media.

In addition, the equations are now scaled when an obvious scaling factor exists. For example, the local mass conservation equations are scaled by the total concentration of the corresponding component. This new choice of variables and the scaling allow to temporarily switch off some of the safeties described in [17] (non-monotone linesearch, perturbation of the Jacobian, descent condition, ...). Depending on the problem disabling some of these options may allow faster and easier convergence. The scaling introduced in this version is only for the residuals, the scaling of the Jacobian described in the previous paper can still be applied. It may be important since it avoids failures when the components in the basis are not dominant in the system.

The local mass conservation equation of water is now :

$$\mathscr{C}_{w} = \rho_{w}\varphi_{w}\left(\frac{1}{M_{w}} + \sum_{j=1}^{N_{\text{aqueous}}} v_{jw}b_{j}\right) + \rho_{w}\varphi_{w} \cdot \sum_{s=1}^{N_{\text{sorbed}}} v_{sw}s_{s} + \sum_{m=1}^{N_{\text{mineral}}} v_{mw}\frac{\varphi_{m}}{\bar{V}_{m}} + \sum_{h=1}^{N_{\text{gas}}} v_{hw}\varphi_{g}\frac{f_{h}P_{g}}{RT}$$

$$(15)$$

where \mathscr{C}_w is the total concentration of water (in number of moles per volume), b_i are the molalities of the components, and b_j are the molalities of the secondary species. v_{kj} denotes the stoichiometric coefficient of component A_i in the reaction of species A_k . From left to right, the term of the right hand side corresponds to the total aqueous concentration, the total sorbed concentration (see below), the total solid concentration, and the total gaseous concentration (see below). Similarly, the total concentration for an aqueous component *i* is replaced by:

$$\mathscr{C}_{i} = \rho_{w}\varphi_{w}\left(b_{i} + \sum_{j=1}^{N_{\text{aqueous}}} v_{ji}b_{j}\right) + \rho_{w}\varphi_{w} \cdot \sum_{s=1}^{N_{\text{sorbed}}} v_{si}s_{s}$$
$$+ \sum_{m=1}^{N_{\text{mineral}}} v_{mi}\frac{\varphi_{m}}{\bar{V}_{m}} + \sum_{h=1}^{N_{\text{gas}}} v_{hi}\phi_{g}\frac{f_{h}P_{g}}{RT}$$
(16)

The complementarity condition for the solid phase A_m is now

$$\varphi_m \ge 0, -\mathrm{SI}_m \ge 0, \text{ and } -\varphi_m \mathrm{SI}_m = 0$$
 (17)

where SI_m is the saturation index of the mineral :

$$SI_m = \log \Omega_m, \qquad \Omega_m = \frac{\prod_{i=1}^{N_c} a_i^{\nu_{mi}}}{K_m}$$
(18)

Two new terms appear in the local mass conservation equations: the total gaseous concentration and the total sorbed concentration. Gaseous species are included in the program to model unsaturated samples. Since there is very few evidence of change in the gas pressure inside cement paste or concrete samples under usual conditions, a constant gas pressure (P_g) is assumed throughout the computation. In addition, the gases are assumed ideal. These assumptions greatly simplify the computation since only two cases are possible: (1) the fugacity of the gaseous species is not fixed; in this case, it is treated as a secondary species and it's fugacity is given by Eq. 19, or (2) the fugacity of the gaseous species is fixed, and in this case, the mass conservation equation for the corresponding component is replaced by the equation giving the fugacity :

$$f_h = \frac{\prod_{i=1}^{N_c} a_i^{\nu_{ji}}}{K_h}$$
(19)

If this corresponding component is charged, then the electroneutrality needs to be enforced. The mass conservation equation of another charged component is replaced by the charge balance equation :

$$0 = \sum_{i=1}^{N_c} z_i b_i + \sum_{j=1}^{N_{\text{aqueous}}} z_j b_j$$
(20)

where z_k is the charge of aqueous species A_k . The charge balance can be use in any computation to replace the equation for a component. It was observed that it facilitates the convergence, especially when poor starting guess are used.

The sorption model is also simple, and its implementation was mainly motivated by the MoMaS reactive transport benchmark [9]. A fixed total concentration of sorption sites may be occupied by sorbed species. Each sorbed species is defined by a desorption reaction :

$$\mathbf{A}_{s} \Longleftrightarrow \sum_{i=1}^{N_{c}} \nu_{si} \mathbf{A}_{i} + \nu_{ss} \mathbf{S}_{f}$$
(21)

where S_f denotes the free sorption sites and v_{ss} is the number of sites occupied by the adsorbed species. The law of mass action for the sorbed species (A_s) is defined as :

$$s_{s} = \frac{s_{f}^{\nu_{ss}}}{K_{s}} \prod_{i=2}^{N_{c}} (\gamma_{i} b_{i})^{\nu_{si}}$$
(22)

where s_f is the concentration (per mass of solution) of free sorption site. Hence, these species can be treated as ordinary secondary species, and only the equation describing the conservation of sorption sites is added to the system :

$$\mathscr{C}_{s} = \rho_{w}\varphi_{w}\left(s_{f} + \sum_{s=1}^{N_{\text{sorbed}}} \nu_{ss}s_{s}\right)$$
(23)

 $x_f = \log_{10} s_f$ is the main variable used to solve this equation by analogy with the components.

For greater flexibility, any local mass conservation equations can be replaced to simulate different equilibrium constraints (fixed fugacity, saturated sample, fixed activity, and charge balance on selected component). The equations are chosen automatically according to the constraints specified by the user.

The ionic strength model is solved using the strategy presented in the previous paper [17]. Furthermore, only minor modifications were made to the semismooth newton solver, such as more return codes to distinguish between the different problems that can occur or adding some shortcuts to bypass some safeties (e.g., the Jacobian condition number checking) if some options are turned off.

B Code

SpecMiCP and ReactMiCP are available under a 3-clause BSD license. They can be obtained from a git repository at https://bitbucket.org/specmicp/specmicp. Compilation and installation instructions, as well as examples from the articles and additional tests are available in the same repository.

References

- Amdahl, G.M.: Validity of the single processor approach to achieving large scale computing capabilities. In: Proceedings of the April 18-20, 1967, Spring Joint Computer Conference, ACM, New York, NY, USA, AFIPS '67 (Spring), pp. 483–485 (1967). doi:10.1145/1465482.1465560
- Bary, B., Sellier, A.: Coupled moisture–carbon dioxide–calcium transfer model for carbonation of concrete. Cement Concret. Res. 34(10), 1859–1872 (2004). doi:10.1016/j.cemconres.2004.01.025
- Bernabé, Y., Mok, U., Evans, B.: Permeability-porosity relationships in rocks subjected to various evolution processes. In: Kümpel, H.J. (ed.) Thermo-Hydro-Mechanical Coupling in Fractured Rock, Pageoph Topical Volumes, pp. 937–960, Birkhäuser Basel (2003). doi:10.1007/978-3-0348-8083-1_9
- Bethke, C.: Geochemical and biogeochemical reaction modeling. Cambridge University Press, Cambridge (2008)
- Brunet, J.P.L., Li, L., Karpyn, Z.T., Kutchko, B.G., Strazisar, B., Bromhal, G.: Dynamic evolution of cement composition and transport properties under conditions relevant to geological carbon sequestration. Energy Fuels 27(8), 4208–4220 (2013). doi:10.1021/ef302023v
- Carrayrou, J.: Looking for some reference solutions for the reactive transport benchmark of MoMaS with SPECY. Comput. Geosci. 14(3), 393–403 (2010). doi:10.1007/s10596-009-9161-y

- Carrayrou, J., Mosé, R., Behra, P.: Operator-splitting procedures for reactive transport and comparison of mass balance errors. J. Contam. Hydrol. 68(3–4), 239–268 (2004). doi:10.1016/S0169-7722(03)00141-4
- Carrayrou, J., Hoffmann J, Knabner, P., Kräutle, S., De Dieuleveult, C., Erhel, J., der Lee, J.V., Lagneau, V., Mayer, K.U., MacQuarrie, K.T.: Comparison of numerical methods for simulating strongly nonlinear and heterogeneous reactive transport problems—the MoMaS benchmark case. Comput. Geosci. 14(3), 483–502 (2010a). doi:10.1007/s10596-010-9178-2
- Carrayrou, J., Kern, M., Knabner, P.: Reactive transport benchmark of MoMaS. Comput. Geosci. 14(3), 385–392 (2010b). doi:10.1007/s10596-009-9157-7
- Dagum, L., Menon, R.: OpenMP: an industry standard API for shared-memory programming. IEEE Comput. Sci. Eng. 5(1), 46– 55 (1998)
- Dilnesa, B., Wieland, E., Lothenbach, B., Dähn, R., Scrivener, K.: Fe-containing phases in hydrated cements. Cement Concret. Res. 58(0), 45–55 (2014). doi:10.1016/j.cemconres.2013.12.012
- Dove, P.M., Han, N., Wallace, A.F., De Yoreo, J.J.: Kinetics of amorphous silica dissolution and the paradox of the silica polymorphs. Proc. Nat. Acad. Sci. **105**(29), 9903–9908 (2008). doi:10.1073/pnas.0803798105
- Duguid, A., Scherer, G.W.: Degradation of oilwell cement due to exposure to carbonated brine. Int. J. Greenh. Gas Control 4(3), 546–560 (2010). doi:10.1016/j.ijggc.2009.11.001
- Facchinei, F., Pang, J.S.: Finite-dimensional variational inequalities and complementarity problems. Springer, New York (2003)
- Gallé, C., Peycelon, H., Bescop, P.L.: Effect of an accelerated chemical degradation on water permeability and pore structure of cementbased materials. Adv. Cement Res. **105–114**(9), 16 (2004)
- Georget, F., Prévost, J.H., Huet, B.: A robust numerical framework for reactive transport in cement paste. In: 5th International Workshop for Young Doctors in Geomechanics (2015a)
- Georget, F., Prévost, J.H., Vanderbei, R.J.: A speciation solver for cement paste modeling and the semismooth Newton method. Cement Concret. Res. 68(0), 139–147 (2015b). doi:10.1016/j. cemconres.2014.11.001
- 18. Georget, F., Huet, B., Prevost, J.H.: (In preparation) Reactive transport modelling of cement paste leaching in brines
- Guennebaud, G., Jacob, B., et al. Eigen v3. http://eigen.tuxfamily. org (2010)
- Hills, R.G., Porro, I., Hudson, D.B., Wierenga, P.J.: Modeling onedimensional infiltration into very dry soils: 1. Model development and evaluation. Water Resour. Res. 25(6), 1259–1269 (1989). doi:10.1029/WR025i006p01259
- Hoffmann, J., Kräutle, 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. 14(3), 421–433 (2010). doi:10.1007/s10596-009-9173-7
- Huet, B.M., Prevost, J.H., Scherer, G.W.: Quantitative reactive transport modeling of Portland cement in CO2-saturated water. Int. J. Greenh. Gas Control 4(3), 561–574 (2010). doi:10.1016/j.ijggc.2009.11.003
- 23. Hummel, W.: NAGRA/PSI chemical thermodynamic data base 01/01 Universal Pub, Parkland, Fla (2002)
- Iqbal, P.O., Ishida, T.: Modeling of chloride transport coupled with enhanced moisture conductivity in concrete exposed to marine environment. Cement Concret. Res. **39**(4), 329–339 (2009). doi:10.1016/j.cemconres.2009.01.001
- Kräutle, S.: The semismooth Newton method for multicomponent reactive transport with minerals. Adv. Water Resour. 34(1), 137– 151 (2011). doi:10.1016/j.advwatres.2010.10.004

- Lagneau, V., Van der Lee, J.: HYTEC results of the MoMas reactive transport benchmark. Comput Geosci. 14(3), 435–449 (2010a). doi:10.1007/s10596-009-9159-5
- Lagneau, V., Van der Lee, J.: Operator-splitting-based reactive transport models in strong feedback of porosity change: The contribution of analytical solutions for accuracy validation and estimator improvement. J. Contam. Hydrol. **112**(1–4), 118–129 (2010b). doi:10.1016/j.jconhyd.2009.11.005, frontiers in Reactive Transport: Microbial Dynamics nad Redox Zonation in the Subsurface
- Leonard, B.P., Drummond, J.E.: Why you should not use 'Hybrid', 'Power-Law' or related exponential schemes for convective modelling? There are much better alternatives. Int. J. Numer. Methods Fluids 20(6), 421–442 (1995). doi:10.1002/fld.1650200602
- LeVeque, R.: Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics, Cambridge University Press (2002)
- Lichtner, P., Steefel, C., Oelkers, E. (eds) (1996) Reactive transport in porous media. No. 34 in Reviews in mineralogy, Mineralogical Society of America
- Lothenbach, B., Matschei, T., Möschner, G., Glasser, F.P.: Thermodynamic modelling of the effect of temperature on the hydration and porosity of Portland cement. Cement Concret Res. 38(1), 1–18 (2008). doi:10.1016/j.cemconres.2007.08.017
- Lothenbach, B., Bary, B., Bescop, P.L., Schmidt, T., Leterrier, N.: (2010) Sulfate ingress in Portland cement. Cement Concret. Res. 40(8), 1211–1225. doi:10.1016/j.cemconres.2010.04.004
- Mainguy, M., Tognazzi, C., Torrenti, J.M., Adenot, F.: Modelling of leaching in pure cement paste and mortar. Cement Concret. Res. 30(1), 83–90 (2000). doi:10.1016/S0008-8846(99)00208-2
- 34. Marty, N., Munier, I., Gaucher, E., Tournassat, C., Gaboreau, S., Vong, C., Giffaut, E., Cochepin, B., Claret, F.: Simulation of cement/clay interactions: feedback on the increasing complexity of modelling strategies. Transp. Porous Med. **104**(2), 385–405 (2014). doi:10.1007/s11242-014-0340-5
- Matschei, T., Lothenbach, B., Glasser, F.P.: Thermodynamic properties of Portland cement hydrates in the system CaO–Al2O3– SiO2–CaSO4–CaCO3–H2O. Cement Concret. Res 37(10), 1379– 1410 (2007). doi:10.1016/j.cemconres.2007.06.002
- Mayer, K., MacQuarrie, K.: Solution of the MoMaS reactive transport benchmark with MIN3P-model formulation and simulation results. Comput. Geosci. 14(3), 405–419 (2010). doi:10.1007/s10596-009-9158-6
- Morel, F., Morgan, J.: Numerical method for computing equilibriums in aqueous chemical systems. Environ. Sci. Technol. 6(1), 58–67 (1972). doi:10.1021/es60060a006

- Prévost, J.H.: Dynaflow: a nonlinear transient finite element analysis program User Manual. NJ, Princeton (2010)
- Schmidt, T., Lothenbach, B., Romer, M., Scrivener, K., Rentsch, D., Figi, R.: A thermodynamic and experimental study of the conditions of thaumasite formation. Cement Concret. Res. 38(3), 337–349 (2008). doi:10.1016/j.cemconres.2007.11.003
- Schwotzer, M., Scherer, T., Gerdes, A.: Protective or damage promoting effect of calcium carbonate layers on the surface of cement based materials in aqueous environments. Cement Concret. Res. 40(9), 1410–1418 (2010). doi:10.1016/j.cemconres.2010.05.001
- Shen, J., Dangla, P., Thiery, M.: Reactive transport modeling of {CO2} through cementitious materials under {CO2} geological storage conditions. Int. J. Greenh. Gas Control 18(0), 75–87 (2013). doi:10.1016/j.ijggc.2013.07.003
- 42. Steefel, C., Appelo, C., Arora, B., Jacques, D., Kalbacher, T., Kolditz, O., Lagneau, V., Lichtner, P., Mayer, K., Meeussen, J., Molins, S., Moulton, D., Shao, H., Šimunek, J., Spycher, N., Yabusaki, S., Yeh, G.: Reactive transport codes for subsurface environmental simulation. Computational Geosciences, 1–34 (2014). doi:10.1007/s10596-014-9443-x
- Steefel, C.I., MacQuarrie, K.T.Lichtner, P.C., Steefel, C.I., Oelkers, E.H. (eds.): Approaches to modeling of reactive transport in porous media, vol. 34. Mineralogical Society of America, Washington D. C (1996)
- 44. Stora, E., Bary, B., He, Q.C., Deville, E., Montarnal, P.: Modelling and simulations of the chemo-mechanical behaviour of leached cement-based materials: leaching process and induced loss of stiffness. Cement Concret. Res. **39**(9), 763–772 (2009). doi:10.1016/j.cemconres.2009.05.010
- 45. Taylor, H.: Cement chemistry. Thomas Telford (1997)
- 46. Torquato, S.: Random heterogeneous materials : microstructure and macroscopic properties. Springer, New York (2002)
- 47. Weidendorfer, J., Kowarschik, M., Trinitis, C.Bubak, M., Van Albada, G., Sloot, P., Dongarra, J. (eds.): a tool suite for simulation based analysis of memory access behavior, vol. 3038. Springer, Berlin Heidelberg (2004). doi:10.1007/978-3-540-24688-6_58
- Wolery, T.J.: Calculation of chemical equilibrium between aqueous solution and minerals the EQ3/6 software package. Tech. Rep. UCRL-52658, LLNL (1979)
- Wolery, T.J., Larek, R.L.: Software User's Manual. EQ3/6, version 8.0 Technical Report SNL-2002-099. Technical report Sandia National Laboratories (2003)
- Yeh, G.T., Tripathi, V.S.: A critical evaluation of recent developments in hydrogeochemical transport models of reactive multichemical components. Water Resour. Res. 25(1), 93–108 (1989). doi:10.1029/WR025i001p00093