Journal of Computational Physics 228 (2009) 6395–6410

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A global strategy for solving reactive transport equations ☆

C. de Dieuleveult [a], J. Erhel [b,*], M. Kern [c]

[a] ANDRA and INRIA Rennes – Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France
[b] INRIA Rennes – Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France
[c] INRIA Paris – Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay Cedex, France

ARTICLE INFO

Article history:
Received 17 November 2008
Received in revised form 11 May 2009
Accepted 13 May 2009
Available online 7 June 2009

Keywords:
Reactive transport
DAE solver
Nonlinear system

ABSTRACT

Reactive transport models are complex nonlinear Partial Differential Algebraic Equations (PDAE), coupling the transport engine with the geochemical operator. We propose an efficient and robust global numerical method, based on a method of lines and Differential Algebraic Equations (DAE) solvers, combined with a Newton method using a powerful sparse linear solver. Numerical experiments show the performances of the method. We also propose a unified framework to describe classical methods such as Sequential Non-Iterative Approach, Sequential Iterative Approach, Direct Substitution Approach and compare them with our global method.

1. Introduction

Many environmental studies rely on modelling geochemical reactions combined with hydrodynamic processes such as groundwater flow, transport of solutes by advection and dispersion, heat transfer in porous media. Some potential applications concern for example aquifer contamination, underground waste disposal, remediation. Reactive transport models involve large nonlinear algebraic partial differential equations. The transport model is based on partial differential equations, with advection, diffusion and reaction operators. In case of kinetic reactions, the chemistry model is composed of reaction terms in the transport equations for aqueous species and of ordinary differential equations for immobile species. In case of local equilibrium reactions, the chemistry model includes algebraic equations [1]. Here, we assume that all chemical reactions are fast compared to transport so that local equilibrium holds. The methods described in this paper can be extended to the kinetic case. Here, we want to focus on the numerical difficulties induced by the coupling of differential and algebraic equations.

In order to solve the nonlinear Partial Differential Algebraic Equations (PDAE), three main approaches are generally used. A Sequential Non-Iterative Approach (SNIA) solves first for transport then for chemistry at each time step [2–4]. It is easy to implement and can be efficient for some applications but becomes inaccurate and requires very small time steps for complex systems. Thus a Sequential Iterative Approach (SIA) is preferred, iterating between transport and chemistry at each time step [5,6]. The third approach, called a global approach, does not decouple transport from chemistry but solves a coupled nonlinear system at each time step. First attempts were efficient for small systems [7,8] but interest grows again with the advent of high performance computing [9–11]. There are several variants of the global approach, differing in the way the equations are coupled; in the Direct Substitution Approach, the chemistry variables are substituted in the transport equations [12–14], leading to a nonlinear system of PDEs.

☆ This work was partially supported by the GdR Momas (CNRS Pacen Program) and by Andra Agency.

* Corresponding author. Tel.: +33 299847339.
E-mail addresses: erhel@irisa.fr, jocelyne.erhel@inria.fr (J. Erhel).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.05.044

We present here a unified framework for most of the methods mentioned above and we introduce a new formulation of a global approach using the power of DAE solvers. We consider only chemistry at equilibrium and only aqueous or solid species but no gas. In the transport model, we assume that the porous medium is saturated, that the dispersion tensor and the porosity do not depend on the species. Moreover, the porosity is strictly positive everywhere. With these assumptions, it is easy to write a reactive transport model where the differential variables are the total analytical aqueous components. By adding the chemistry model, we get a coupled set of PDAE. There are various ways to discretize the transport operator but we restrict this study to Eulerian approaches, for example by Finite Differences, Finite Volume or Finite Element methods. Moreover, we consider a method of lines where space is first discretized, and the resulting semi-discrete system is then solved by an ODE (or DAE) solver. This excludes methods combining time and space discretizations. We also assume that the same discrete space is used for advection and dispersion, so that we get a discrete advection–dispersion equation. With these assumptions, the reactive transport model is transformed into a set of DAE (Differential Algebraic Equations) of finite dimension, where unknowns are in the discrete space.

Most of the commonly used approaches can indeed be rewritten in this framework; they differ in the numerical scheme used to discretize in time and in the numerical method used to solve the nonlinear equations. Our approach uses an implicit scheme and a Newton method embedded in a DAE solver; the objective is to rely on the solver for controlling the time step, the order of the scheme, the updates of the Jacobian and the convergence of Newton iterations.

The paper is organized as follows: Section 2 describes the reactive transport model with the set of PDAEs and the unified framework using the method of lines leading to the set of DAEs; Section 3 describes various common approaches rewritten within this framework. In Section 4, we present our new formulation using the classical interface of DAE solvers. Section 5 is devoted to numerical experiments; we have chosen two examples from the literature, with two different chemical systems. In the first example, we vary the transport conditions by varying the Peclet number, the ratio of advection over diffusion. We compare our global approach with two other methods, namely a method of type SNIA implemented in the PhreeqC software and a method of type SIA that we have implemented. Results show that our global approach is quite competitive. Finally, Section 6 gives some concluding remarks and a roadmap for future work.

2. Reactive transport model

2.1. Chemistry

We consider a geochemical system composed of aqueous, sorbed and precipitated chemical species and we assume that all chemical processes are in thermodynamical equilibrium. Several chemistry models can be defined by considering various sets of unknowns [5]. One can keep all species as unknowns in the equations [8] or define additional unknowns or split species into two parts, primary and secondary. Following the method proposed in [15,16], we split the species between components (primary species) and secondary species. Moreover, we focus our attention on formulations based on total analytical concentrations [17,18].

We assume that there are N_r aqueous and adsorption chemical reactions between N_e chemical species:

$$\sum_{j=1}^{N_e} \alpha_{ij} X_j \leftrightarrows 0, \quad i = 1, \ldots, N_r.$$

The matrix $\alpha \in \mathbf{R}^{N_e \times N_r}$ is the stoichiometric matrix, such that the submatrix corresponding to the components is a full rank submatrix of α. Additionally, as this is essential to formulate the reactive transport problem in the next subsection, we also distinguish between aqueous and fixed species. Let N_c be the number of aqueous components, N_s be the number of sorbed components, N_x be the number of secondary aqueous species, N_y be the number of secondary sorbed species. We end up with the following system of aqueous and adsorption reactions:

$$\begin{cases} x_i \leftrightarrows \sum_{j=1}^{N_c} S_{ij} c_j, \quad i = 1, \ldots, N_x, \\ y_i \leftrightarrows \sum_{j=1}^{N_c} A_{ij} c_j + \sum_{j=1}^{N_s} B_{ij} s_j, \quad i = 1, \ldots, N_y. \end{cases}$$

These chemical reactions are described by the mass action laws, where we use the same notation for a chemical species and its concentrations and take the activity coefficients equal to one; the mass action laws are written

$$\begin{cases} x_i = K_{xi} \prod_{j=1}^{N_c} c_j^{S_{ij}}, \quad i = 1, \ldots, N_x, \\ y_i = K_{yi} \prod_{j=1}^{N_c} c_j^{A_{ij}} \prod_{j=1}^{N_s} s_j^{B_{ij}}, \quad i = 1, \ldots, N_y, \end{cases}$$

where $c \in \mathbf{R}^{N_c}$ (resp. $s \in \mathbf{R}^{N_s}$) is the vector of concentrations of the aqueous (resp. sorbed) components, $x \in \mathbf{R}^{N_x}$ (resp. $y \in \mathbf{R}^{N_y}$) is the vector of concentrations of the aqueous (resp. sorbed) secondary species, $K_x \in \mathbf{R}^{N_x}$ and $K_y \in \mathbf{R}^{N_y}$ are the vectors of equilibrium constants, and $S \in \mathbf{R}^{N_x \times N_c}$, $A \in \mathbf{R}^{N_y \times N_c}$, $B \in \mathbf{R}^{N_y \times N_s}$ are the stoichiometric coefficients.

We also consider reactions of precipitation and dissolution. Let N_p be the number of precipitated species and $p \in \mathbf{R}^{N_p}$ the vector of the number of moles of the precipitated species per unit of volume; at equilibrium, reactions of precipitation are governed by a threshold of saturation $\Pi \in \mathbf{R}^{N_p}$ and a mass action law

$$\begin{cases} \Pi_i = K_{pi} \prod_{j=1}^{N_c} c_j^{G_{ij}}, \quad i = 1, \ldots, N_p, \\ p_i(\Pi_i - 1) = 0, \\ p_i \geqslant 0, \\ \Pi_i \leqslant 1. \end{cases}$$

From now on, we assume that each mineral is either always totally dissolved ($p_i = 0$) or never totally dissolved ($p_i > 0$); with this assumption, the number N_p of precipitated species where the threshold is attained is known and fixed, so that the equations above can be rewritten as $\log K_p + G \log c = 0$ with the constraint $p > 0$.

The model is completed by a mass conservation relation for each component. Let T and W be the total concentrations for mobile and fixed components. In the context of a closed chemical system, T and W are known. For an open system, W is assumed given and constant, whereas T is one unknown of the system, coupled with the transport equations. It is useful, both for writing the previous equations in a concise form, and also for the numerical solution, to work with the logarithm of the concentrations. Let us introduce the variables $lc = \log c$ and $ls = \log s$ (log means the natural logarithm here) and rewrite the system as

$$\begin{cases} \exp(lc) + S^T x + A^T y - T = 0, \\ \exp(ls) + B^T y - W = 0, \end{cases} \tag{1}$$

if there are no precipitated species or

$$\begin{cases} \exp(lc) + S^T x + A^T y + G^T p - T = 0, \\ \exp(ls) + B^T y - W = 0, \\ \log K_p + Glc = 0, \\ p > 0, \end{cases} \tag{2}$$

if there are precipitated species, with, in both cases,

$$\begin{cases} x = \exp(\log K_x + Slc), \\ y = \exp(\log K_y + Alc + Bls). \end{cases}$$

This is a system of $(N_c + N_s + N_p)$ nonlinear equations with $(N_c + N_s + N_p)$ unknowns $X = \begin{pmatrix} lc \\ ls \\ p \end{pmatrix}$ along with an inequality constraint.

We assume that the system has a unique solution, that the functions are differentiable and that the Jacobian of the system is nonsingular. In what follows, we will write a local chemistry system in each cell of a spatial grid. We use a subscript c in order to make the distinction with the global chemistry system written in the whole domain. Under these assumptions, we write the system (2) in the condensed form

$$\Phi_c(X, T) = 0. \tag{3}$$

The Jacobian of this system is given by

$$J_c(X) = \begin{pmatrix} \operatorname{diag} \exp(lc) + S^T \operatorname{diag} xS + A^T \operatorname{diag} yA & A^T \operatorname{diag} yB & G^T \\ B^T \operatorname{diag} yA & \operatorname{diag} \exp(ls) + B^T \operatorname{diag} yB & 0 \\ G & 0 & 0 \end{pmatrix},$$

where diag$x$ is a diagonal matrix with the vector x along the diagonal.

In reactive transport models, it is useful to split the total concentrations between mobile and immobile concentrations, defining

$$\begin{cases} C_c(X) = c + S^T x, \\ F_c(X) = A^T y + G^T p. \end{cases} \tag{4}$$

We also introduce the notations

$$\begin{cases} W_c(X) = s + B^T y, \\ P_c(X) = \log K_p + Glc. \end{cases}$$

Most chemistry solvers get T and W as input and compute X, then $C_c(X)$ and $F_c(X)$ as output. The solver is in general based on Newton's method, modified to ensure global convergence. The solver can be formally written as $X = \Psi_c(T)$. The software implementing a chemistry solver is in general a black box, in the sense that there is no function returning the nonlinear equations or their derivatives. Thus it is not possible to evaluate the function $\Phi_c(X, T)$ or the Jacobian $J_c(X)$. Therefore, the Jacobian of $\Psi_c(T)$ or the derivatives of $C_c(X)$, $F_c(X)$ can only be approximated numerically by finite differences.

2.2. Reactive transport

We consider a porous medium in the subsurface and make some classical assumptions: the flow is stationary and the porous medium is saturated; the porosity is constant; the transport affects only aqueous species (by definition); the dispersion coefficients do not depend on the species. Under these assumptions, the transport of a species of concentration u is governed by advection and dispersion and can be written as

$$\omega \frac{\partial u}{\partial t} + L(u) = q, \tag{5}$$

where ω is the porosity, q is the source term and the spatial part of the transport operator is given by

$$L(u) = \nabla \cdot (vu - D\nabla u),$$

where v is the fluid velocity (presumably computed by a flow simulation) and D is the diffusion–dispersion tensor, given by

$$D = d_m I + \alpha_T \|v\| I + (\alpha_L - \alpha_T)E(v),$$

with $E_{ij}(v) = \frac{v_i v_j}{\|v\|}$, where d_m is the molecular diffusion coefficient, α_L (resp. α_T) is the longitudinal (resp. transverse) dispersion coefficient. Both v and D may depend on space but do not depend on u so that L is linear. This partial differential equation is completed by boundary conditions and an initial condition. Usually, boundary conditions prescribe u for inflow and prescribe the flux for outflow; either the total flux, or only the diffusive flux (the most common choice) can be given.

Transport equations such as (5) are defined for each species, with q including both a given source term and a chemistry reaction term. Of course, in the equilibrium case, these chemistry reaction terms are not known but it is possible to eliminate them by using the components and secondary species introduced above [5,19]. Thus the reactive transport model is a set of independent transport equations written for each component

$$\omega \frac{\partial T_i}{\partial t} + L(C_i) = q_i, \quad i = 1, \ldots, N_c, \tag{6}$$

where $T_i,\ C_i$ are the total analytical and total mobile concentrations of the i chemical component, with q_i now including only the known source term. Transport equations (6) are coupled with the chemistry equations (3) and (4).

The boundary and initial conditions are applied to the variable T. Initial conditions for other variables are obtained by solving the chemical equations at each point, using the initial condition for T. In this way, we get a consistent set of initial conditions. We follow the same lines for Dirichlet boundary conditions. Fluxes of fixed species are zero thus fluxes of T and C are equal, so that Neumann boundary conditions can be applied also to C.

A different approach starts from the individual transport equations, but does not introduce components and secondary species [14,10,12]. The unknown reaction rates are still eliminated, but in a way that leads to a smaller number of transport equations, and to a decoupling of these equations, so that the resulting nonlinear system is smaller than in the other approaches.

2.3. A unified framework for numerical methods

Quite often, the equations are discretized by the method of lines [20]. In a first step, the method applies a spatial discretization to the transport operator L, in order to transform the Partial Differential Equations (6) into a system of Differential Equations of finite dimension. The transport operator is spatially discretized for example by a Finite Volume Method, using a mesh of the computational domain. We do not discuss here the choice of spatial discretization, but it should be guided by numerical considerations such as artificial oscillations or numerical diffusion [20]. Let N_m be the number of unknowns in the mesh, called points in the following. Since L is linear, we still denote the discrete operator by L, thus $L(u) = Lu + g$, with $L \in \mathbf{R}^{N_m \times N_m}$, where $g \in \mathbf{R}^{N_m}$ represents the contributions of the boundary conditions. We also assume that $\omega = 1$ and omit the porosity from now on. The semi-discrete transport equations are written

$$\frac{du}{dt} + Lu = q + g.$$

In the coupled reactive transport model, the chemical equations are also discretized, in the sense that they are written for each degree of freedom of the spatial scheme, so that they are transformed into a system of algebraic equations of finite dimension. The discrete reactive transport numerical model is thus a system of Differential Algebraic Equations of index one (DAE) [21]. Indeed, it is sufficient to differentiate once the algebraic chemistry equations in order to get a system of Ordinary Differential Equations (ODE). Thus the differentiation index is equal to one here.

In order to get a unified framework for all numerical models and methods, we introduce a matrix storage of the data. All variables and data are matrices $T = (T_1, \ldots, T_{N_m}),\ C,\ F,\ q,\ g \in \mathbf{R}^{N_c \times N_m},\ X = (X_1, \ldots, X_{N_m}) \in \mathbf{R}^{(N_c+N_s+N_p)\times N_m}$. Each column of T, denoted by $T_j,\ j = 1, \ldots, N_m$, represents all the components at one point of the mesh and each row of T, noted $T_i,\ i = 1, \ldots, N_c$, represents one component at all points of the mesh. Thanks to this notation, we can write all the transport equations in two different equivalent ways, either

$$\frac{d}{dt}T^T + LC^T = q^T + g^T$$

using matrices or

$$\frac{d}{dt}(\mathbf{vec}\,T) + (L \otimes I)\mathbf{vec}\,C = \mathbf{vec}\,q + \mathbf{vec}\,g$$

using vectors with **vec** operators, which rearrange matrices into vectors and with Kronecker products.

Similarly, the chemistry operators become in matrix form

$$\Phi(X,T) = (\Phi_c(X_1,T_1),\dots,\Phi_c(X_{N_m},T_{N_m})),$$

$$\Psi(T) = (\Psi_c(T_1),\dots,\Psi_c(T_{N_m})),$$

$$C(X) = (C_c(X_1),C_c(X_2),\dots,C_c(X_{N_m})),$$

$$F(X) = (F_c(X_1),F_c(X_2),\dots,F_c(X_{N_m})),$$

$$W(X) = (W_c(X_1),W_c(X_2),\dots,W_c(X_{N_m})),$$

$$P(X) = (P_c(X_1),P_c(X_2),\dots,P_c(X_{N_m}))$$

and can also be written in vector form using the **vec** operator.

Depending on which of the three variables $T,\ C,\ F$ are kept, we can define several formulations [5,17], which we write using a unifying DAE framework. We study two of them, the so-called *TC* and *CC* formulations.

We get for the *TC* formulation

$$\begin{cases} \frac{d}{dt}T^T + LC^T = q^T + g^T, \\ \Phi(X,T) = 0, \\ C - C(X) = 0, \end{cases} \tag{7}$$

whereas the formulation *CC* is written

$$\begin{cases} \frac{d}{dt}C^T + \frac{d}{dt}F^T + LC^T = q^T + g^T, \\ \Phi(X,C+F) = 0, \\ F - F(X) = 0. \end{cases} \tag{8}$$

Both systems are composed of N_mN_c differential equations and $N_m(2N_c + N_s + N_p)$ algebraic equations, completed with an initial condition. We can now apply a time discretization in order to solve these DAE systems of index 1. We first recall classical schemes such as explicit Euler, which leads to the class of SNIA methods and implicit Euler, which leads to the classes of SIA methods and global methods. Then we describe our global approach, which is based on general DAE solvers.

3. Classical numerical methods

3.1. Explicit time discretization and Sequential Non-Iterative Approach (SNIA)

Explicit schemes are easy to use and enable decoupling the equations. With the *CC* formulation, transport and chemistry equations remain coupled, so that the system is a global set of nonlinear equations. On the other hand, with the *TC* formulation, transport and chemistry equations are fully decoupled. Therefore, the implementation can use a transport module and a chemistry module as black boxes. The transport module takes the source terms and the values at the previous time step as inputs and returns T_{n+1} as output. The chemistry module takes T_{n+1} as input and returns C_{n+1} as output. For example, with an explicit Euler scheme and a time step Δt, the discrete equations are written

$$\begin{cases} T_{n+1}^T + \Delta t L C_n^T - T_n^T = \Delta t(q_n^T + g_n^T), \\ \Phi(X,T_{n+1}) = 0, \\ C_{n+1} - C(X) = 0. \end{cases}$$

This scheme is called a Sequential Non-Iterative Approach (SNIA). Transport equations can be solved independently and explicitly for each species, then chemical equations can be solved independently for each point. Moreover, if there is no modification at a point, there is no need to solve chemical equations at that point. However, stability conditions require a very small time step, leading to heavy CPU requirements. This method is mostly suited for advection-dominated problems where the stability condition is not too high, whereas the stability condition for dispersion-dominated problems is prohibitive.

It should be noted that sometimes the spatio-temporal scheme must still satisfy some restriction (for example a cell Peclet condition) to avoid numerical oscillations. However, it is possible to choose a positive scheme without these oscillations.

3.2. Operator splitting

A classical way to improve the numerical scheme is to apply operator splitting [20]. Here, the transport operator is split into advection and diffusion operators by $L = L_{adv} + L_{dis}$ and a different scheme, explicit or implicit, can be applied to each operator. However, boundary conditions can introduce a significant error if they are not correctly implemented. For example, with an explicit Euler scheme in both advection and diffusion and with a time step Δt, the operator splitting leads to the equations

$$\begin{cases} T_*^T + \Delta t L_{adv} C_n^T - T_n^T = \Delta t\left(q_n^T + g_n^T\right), \\ \Phi(X, T_*) = 0, \\ C_* - C(X) = 0, \\ T_{n+1}^T + \Delta t L_{dis} C_*^T - T_*^T = 0, \\ \Phi(X, T_{n+1}) = 0, \\ C_{n+1} - C(X) = 0. \end{cases}$$

This scheme is implemented in the PhreeqC library for 1D problems [22–24]. Again, stability conditions lead to severe restrictions on the time step. However, an implicit scheme could be used for dispersion in order to remove stability conditions for this operator [25]. Thus, SNIA can be used for advection and one of the methods described below can be applied to dispersion.

3.3. Implicit time discretization and Sequential Iterative Approach (SIA)

Now, we consider implicit schemes applied globally to the transport operator. The methodology is still relevant if operator splitting is applied with an implicit scheme for dispersion. With an implicit scheme, each time step leads to a global nonlinear system of equations, where transport and chemistry equations are coupled. Therefore, it is necessary to solve nonlinear coupled equations.

For example, an implicit Euler scheme with a time step Δt is written

$$\begin{cases} T_{n+1}^T + \Delta t L C_{n+1}^T - T_n^T = \Delta t\left(q_{n+1}^T + g_{n+1}^T\right), \\ \Phi(X, T_{n+1}) = 0, \\ C_{n+1} - C(X) = 0 \end{cases} \tag{9}$$

in the *TC* formulation and

$$\begin{cases} C_{n+1}^T + F_{n+1}^T + \Delta t L C_{n+1}^T - C_n^T - F_n^T = \Delta t\left(q_{n+1}^T + g_{n+1}^T\right), \\ T_{n+1} = C_{n+1} + F_{n+1}, \\ \Phi(X, T_{n+1}) = 0, \\ F_{n+1} - F(X) = 0 \end{cases} \tag{10}$$

in the *CC* formulation, where we have introduced *T* for sake of consistency with the chemistry operator.

Various methods based on a fixed point method have appeared in the geochemical literature to solve these nonlinear systems [5,26,6,27–29]. When the *CC* formulation is used, this method is known as Picard's method or Sequential Iterative Approach (SIA). It can also be viewed as a nonlinear block-SOR–Newton method [30]. The idea is very simple and generalizes the classical linear SOR or block-SOR method. Here, the first equation (with the transport operator) is solved for *T* with a given *C* in the *TC* formulation, or for *C* with a given *F* for the *CC* formulation. Then the nonlinear chemistry equations are solved for *X* with the updated *T*, using Newton iterations. Finally *C* or *F* is computed from *X*. This scheme is iterated until convergence. Thus outer iterations implement a fixed point approach (block-SOR) and inner iterations implement a nonlinear solver (Newton).

Using the *TC* formulation, the method solves for *T* then for *X* then for *C*, so that the transport equations do not require any linear solver, are independent for each component and decoupled from chemistry. Then chemistry can be solved simultaneously for each point, using any chemistry solver, decoupled from transport. This scheme has the same advantages as the explicit scheme and is more accurate because it iterates. However, experiments show that it may still require a small time step to converge.

In the *CC* formulation, blocks of unknowns are $C,\ X,\ F$ and the SIA method yields

$$\begin{cases} C_i^{k+1} + \Delta t L C_i^{k+1} + F_i^k - T_i = \Delta t(q_{n+1,i} + g_{n+1,i}), \quad i = 1, \ldots, N_c, \\ T_j^{k+1} = C_j^{k+1} + F_j^k, \quad j = 1, \ldots, N_m, \\ \Phi_c\left(X_j, T_j^{k+1}\right) = 0, \quad j = 1, \ldots, N_m, \\ F_j^{k+1} - F_c(X_j) = 0, \quad j = 1, \ldots, N_m \end{cases} \tag{11}$$

with some initial guess F_i^0. The transport equations can be viewed as one step of an implicit scheme applied to the transport equation with C_i^{k+1} as unknown and F_i^k as source term. The formulation iterates between a transport step and a chemistry step. Therefore, it is easy again to reuse transport and chemistry numerical models as black boxes. However, in some cases, convergence is obtained only for small time steps. Also, one drawback of this formulation is that the equation $T = C + F$ is not satisfied during iterations. This is not important if convergence is attained but can be a problem if iterations are stopped before convergence.

Some implementations of this method do not iterate until convergence but perform only one iteration of the fixed point scheme; this variant is sometimes called an SNIA approach [31,32]. With the *TC* formulation and provided that the initial guess is the value at the previous time step, it is effectively an SNIA approach with an explicit scheme, as described in Section 3.1. However, with the *CC* formulation, performing only one iteration introduces an additional error.

Usually, implementations of this method solve the chemistry until convergence. But the convergence analysis of nonlinear block-SOR–Newton methods [30] shows that, under certain assumptions, it is sufficient to run only one Newton iteration in order to get the same asymptotic convergence.

Thus, it is in theory possible to drastically decrease the CPU requirements, if these assumptions are satisfied and allow to stop the inner Newton iterations early.

3.4. Implicit time discretization and global-ODE approach

Because convergence can be slow in fixed point methods, several authors have switched to a Newton-like method, developing the so-called global approach [33,11,5]. There are several variants, differing by the variables considered in the Newton method. In the global-ODE approach, an implicit Euler scheme is used and the chemistry variables are replaced formally by $X = \Psi_c(T)$, which means that a chemistry solver is called [13,19,34]. In the *TC* formulation, at each time step it is then required to solve the nonlinear equations

$$T_{n+1}^T + \Delta t LC(\Psi(T_{n+1}))^T - T_n^T = \Delta t(q_{n+1}^T + g_{n+1}^T).$$

The main advantage is to use the chemistry solver as a black box, using T as input and returning $X,\ C,\ F$ as output. In general, the chemistry solver does not provide a way to compute the Jacobian. Thus, it must be approximated either by numerical differentiation or by a BFGS-type method. Quite often, the linear system is solved by a Krylov method which involves matrix–vector products. The main drawback is the cost of the method since each function evaluation involves a chemistry solver and each matrix–vector product involves also a chemistry solver. There are two embedded Newton loops, outer iterations for the global system and inner iterations for each chemistry speciation. Moreover, if a Newton–Krylov method is used, then convergence must be accelerated by preconditioning the system but it is difficult to develop a matrix-free preconditioner [13].

3.5. Implicit time discretization and global-DSA approach

In methods called Direct Substitution Approach (DSA), the variables $T,\ C,\ F$ are expressed in terms of the main chemistry variables X, so that X remains the sole unknown. There are two ways of doing this, either before time discretization or after. When it is done before, then X is a solution of a DAE with a mass matrix which depends nonlinearly on X. This is a rather difficult problem, where numerical methods can become inaccurate or are very expensive, thus we exclude this approach. When the substitution is done after the discretization, then X is a solution of a nonlinear system of equations. For example, with the implicit Euler scheme and the *TC* formulation, each time step is written

$$\begin{cases} C(X)^T + F(X)^T + \Delta t LC(X)^T - T_n^T = \Delta t(q_{n+1}^T + g_{n+1}^T), \\ W(X) = W, \\ P(X) = 0, \end{cases} \tag{12}$$

which is a nonlinear system of size $N_m(N_c + N_s + N_p)$. This approach couples transport and chemistry operators, through the compound operator $LC(X)$. The nonlinear system is then solved by a Newton-type method, see for example [11,35].

This approach requires computing $C(X),\ F(X),\ W(X),\ P(X)$ and their derivatives so that it is not possible to use a chemistry solver as a black box; here, the chemistry modules must provide functions and their derivatives, with analytical differentiation. Also, the chemistry modules and the transport modules interact for computing $LC(X)$ and the derivative. Since the system is global, it is much larger than systems in SIA methods, which are written either for all points of one species or for all species at one point.

On the other hand, Newton iterations have a local quadratic convergence so that the approach requires fewer iterations than the fixed point SIA approach. Moreover, there is no need to solve chemistry equations since Newton iterations in DSA require only function evaluations and a global large linear system to solve. Nowadays, many sparse linear solvers are very efficient, even with very large linear systems. The linearized system can be solved either by a direct method (Newton–LU) or by an iterative method such as Newton–Krylov, with efficient preconditioners, although these preconditioners are not easy to design.

4. Our global approach using DAE solvers

We propose to consider the semi-discrete reactive transport model as a system of differential–algebraic equations of index 1 and to use the framework of DAE solvers. Thus we do not assume that an implicit Euler scheme is used.

Let us first consider the TC formulation. Let $y = \begin{pmatrix} \mathbf{vec}\,T \\ \mathbf{vec}\,X \\ \mathbf{vec}\,C \end{pmatrix}$, then system (7) is written

$$M_1 \frac{dy}{dt} + f_1(y) = 0, \tag{13}$$

where $M_1 = \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $f_1(y) = \begin{pmatrix} (L \otimes I)\mathbf{vec}\,C - \mathbf{vec}\,q - \mathbf{vec}\,g \\ \mathbf{vec}\,\Phi(X,T) \\ \mathbf{vec}\,C - \mathbf{vec}\,C(X) \end{pmatrix}$.

The Jacobian J_1 of f_1 is

$$J_1 = \begin{pmatrix} 0 & 0 & L \otimes I \\ -N & \mathbf{diag}(J_c(X_j)) & 0 \\ 0 & -\mathbf{diag}(\frac{dC_c}{dX}(X_j)) & I \end{pmatrix},$$

where $N = \begin{pmatrix} I \\ 0 \end{pmatrix}$. Here $\mathbf{diag}(J_c(X_j))$ is the block-diagonal matrix composed of the blocks $J_c(X_j)$ which are the chemistry Jacobian matrices defined at each point j of the mesh.

Let us now consider the CC formulation. Let $z = \begin{pmatrix} \mathbf{vec}\,C \\ \mathbf{vec}\,X \\ \mathbf{vec}\,F \end{pmatrix}$, then system (8) is written

$$M_2 \frac{dz}{dt} + f_2(z) = 0,$$

where $M_2 = \begin{pmatrix} I & 0 & I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $f_2(z) = \begin{pmatrix} (L \otimes I)\mathbf{vec}\,C - \mathbf{vec}\,q - \mathbf{vec}\,g \\ \mathbf{vec}\,\Phi(X, C+F) \\ \mathbf{vec}\,F - \mathbf{vec}\,F(X) \end{pmatrix}$.

The Jacobian J_2 of f_2 is

$$J_2 = \begin{pmatrix} L \otimes I & 0 & 0 \\ -N & \mathbf{diag}(J_c(X_i)) & -N \\ 0 & -\mathbf{diag}(\frac{dF_c}{dX}(X_i)) & I \end{pmatrix}.$$

In classical global approaches, an implicit Euler scheme is used along with a TC formulation. This leads, at each time step, to the nonlinear system $M_1 y + \Delta t f_1(y) - M_1 y_n = 0$ and the Jacobian of this system is the matrix $M_1 + \Delta t J_1$. This system can be solved by a nonlinear block-SOR–Newton method as in the Sequential Iterative Approach or by a Newton method as in the DSA approach. Newton iterations are written for the TC formulation

$$(M_1 + \Delta t J_1)(y^{k+1} - y^k) = -(M_1 y^k + \Delta t f_1(y^k) - M_1 y^n).$$

We claim that an efficient way to solve the DAE equations is to use a DAE solver with an implicit time discretization. Our approach is a generalization of the global approach, which is a particular case with an implicit Euler scheme and a fixed time step. Thus we keep the advantages recalled above: the method is robust, there is no chemistry solver inside Newton iterations. Again, chemistry modules must provide chemistry functions and their derivatives. Since transport and chemistry equations remain well separated (there is no compound operator), it becomes easier than in a DSA approach to clearly identify transport and chemistry operators along with their interfaces. Compared to a fixed implicit scheme such as Euler, a big advantage of DAE solvers is their control of accuracy with a variable order scheme. Moreover, these solvers provide a control of time step and the associated control of Jacobian updates (ensuring convergence of Newton iterations for example). It is in principle possible to implement these controls but they are rather sophisticated in DAE solvers and not so easy to reproduce.

On the other hand, a drawback of our approach is the size of the linear system which is $N_m(3N_c + N_s + N_p)$ instead of $N_m(N_c + N_s + N_p)$ in the DSA approach. However, the matrix is quite sparse and has some structure, so it is not clear wether solving this large system requires much more time than solving the system in DSA. Actually, it is possible to use substitution in the linearized equations. Indeed, the unknowns $T,\ C$ can be eliminated in order to get a linear system with only X as unknown; the resulting matrix is roughly the same as the matrix used in DSA. Therefore, we advocate this global approach where the nonlinear system is considered with all variables and where an appropriate sparse solver is used.

In order to separate transport from chemistry, a linear block-SOR method (based on a fixed point technique) can be applied to the linear system inside each Newton iteration, leading to a Newton–block-SOR method. For example, with a CC formulation and an implicit Euler scheme, the method first computes C by solving the transport equations, then X by solving the linearized chemistry equations at each point, finally F with a linearized evaluation at each point. Compared to the SIA method (block-SOR–Newton), now outer iterations are Newton and inner iterations are fixed point, so that the fixed point strategy is applied to linearized operators, thus reducing drastically the CPU requirements. At least in theory [30], Newton–block-SOR converges faster than block-SOR–Newton.

Similarly to DSA methods, the nonlinear equations can be solved by Newton–LU or by Newton–Krylov methods. Here, it is easy to identify transport functions and chemistry functions.

4.1. Our DAE-based reactive transport solver

In order to separate as much as possible transport and chemistry, we adopt the following strategy. We use software libraries and write modules clearly identified as transport or chemistry. We apply a DAE solver (in our experiments, the IDA module, from the SUNDIALS package [36]), to which we provide the mass matrix M, the function f, the Jacobian J and consistent initial conditions. Then the DAE solver applies a time discretization scheme and solves at each time step a system of nonlinear equations. In order to distinguish between transport and chemistry as much as possible, we keep all the unknowns, so that we can clearly identify transport modules and chemistry modules. Thus we do not reduce the size of the linear systems but provide an efficient sparse linear solver. In our experiments, we implement Newton–LU by interfacing SUNDIALS with the library UMFPACK so that we can solve efficiently the large linear systems.

In the first line of both Jacobians, corresponding to the transport equation, the discrete linear transport operator has no chemistry interference. The second line is the linearized chemistry operator, with $C + F$ or T as input. The third line corresponds to the coupling between transport and chemistry. Therefore, it is easy at this stage to identify transport and chemistry operations. Thus, we define modules to compute blocks of the function f and blocks of the Jacobian J:

computation of L: transport;
computation of $\Phi_c(X)$: chemistry;
computation of $F_c(X)$ or $C_c(X)$: chemistry;
computation of $J_c(X)$: chemistry;
computation of $\frac{dF_c}{dX}(X)$ or $\frac{dC_c}{dX}(X)$: chemistry.

Thanks to these modules, we can provide modules which compute the function f and the Jacobian J. Recall that vectors are numbered point by point, so that chemistry is easily written for each point. Therefore, the chemistry modules can be written at each point of the mesh with results assembled in a large vector or large block-diagonal matrix. On the other hand, we have to compute the Kronecker product $L \otimes I$.

4.2. Transport modules

In this section, we give an example of transport modules for a 1D problem with a Finite Volume spatial discretization.

Let us consider a 1D domain $[0, L]$. We assume a constant flow velocity $v > 0$, a constant diffusion parameter $D > 0$. A Dirichlet boundary condition is imposed on inflow $(x = 0)$ and a Neumann boundary condition is set on outflow $(x = L)$, where fluxes of T and C are equal. The initial condition and Dirichlet boundary condition for C are computed by solving the chemistry equations (knowing T and W).

For each chemical component, the 1D transport equations are written as follows:

$$\begin{cases} \frac{\partial T}{\partial t} + v\frac{\partial C}{\partial x} - D\frac{\partial^2 C}{\partial x^2} = q, \\ T(x,0) = T_0(x), \\ T(0,t) = T_D(t), \\ D\frac{\partial C}{\partial x}(L,t) - vC(L,t) + \lambda C(L,t) = g_L(t). \end{cases} \tag{14}$$

The boundary condition can be either for the total flux (if $\lambda = 0$), or for the diffusive flux (if $\lambda = v$).

We use a cell-centered finite volume scheme [37] and define a uniform space grid on $[0, L]$. The advection term is discretized by a first-order upwind scheme, whereas we use a second-order centered scheme for the diffusion term. We adopt a method of lines, so that we get a set of differential equations. We integrate Eq. (14) above on each control cell, assuming the function C is constant on each cell and the same for the source term q.

The semi-discrete scheme in space can thus be written $\frac{dT}{dt} + LC = f + g$, where L is a tridiagonal matrix of order N_m. Details can be found in [38].

5. Numerical experiments

In order to assess the efficiency of our method, we compare it to two nonglobal methods, namely the SNIA method with operator splitting implemented in PhreeqC and a SIA method. We have implemented a SIA method (with the *CC* formulation) and our global DAE method (with the *TC* formulation), leading to Eq. (13). We consider 1D domains and discretize spatially with the transport module described above, yielding the discrete transport operator L. This operator L is used in both our SIA and global DAE implementations. We use the same chemistry modules in SIA and DAE.

For SIA, we implement an implicit Euler scheme and a basic control of time step and convergence. We solve the nonlinear chemistry equations using KINSOL, included in SUNDIALS. If convergence fails, then we reduce the time step. Also, if the outer fixed point iterations fail to converge, then we reduce the time step. Conversely, we increase the time step if convergence is fast.

For our global method, we use the software IDA, included in SUNDIALS, which we interfaced with UMFPACK to solve the DAE, thus we write functions to compute M_1, f_1 and J_1 and to compute the initial state.

We focus our numerical experiments on two particular examples. The first example, an exchange of ions inside a column, comes from the test cases of the PhreeqC software [22], whereas the second example, an injection of alcaline water into a column of quartz, comes from the qualification plan of the French Alliances project [32]. These two examples are used in the geochemistry community and are a good basis for validating our software and for showing the efficiency of our method. For the first example, we consider several transport models, ranging from pure advective to pure diffusive operators. To validate our method, the numerical results obtained with PhreeqC are considered as the reference. To show the efficiency of our method, SIA is also considered as a reference since it is heavily used in geochemistry. We check convergence of the numerical scheme by refining the mesh and we study the efficiency by measuring the difference between the current mesh and the finest mesh. We do not study the impact of the time step because the time step is controlled internally by the numerical scheme. Computations are done on an Intel Pentium 4 with a frequency of 3 GHz, 1 GB of RAM and 1 MB of cache, under the operating system Linux Fedora Core 6.

5.1. Ion-exchange inside a column

5.1.1. Description of the test case

This test case is Example 11 from the base of examples proposed by PhreeqC [22]. In this test case, the chemical process takes place in a 1D domain. The computational domain is a column of length $L = 0.08$ m containing initially potassium, sodium and nitrate ions in equilibrium with a cation exchanger. On top of this column, a calcium chloride solution is injected continuously. Calcium reacts with the cation exchanger modifying the chemical equilibrium, whereas chloride is only transported along the column. In this test case, the four aqueous components are Na^+, Ca^{2+}, K^+ and Cl^-, whereas the only sorbed component is Na—X. Since Cl^- does not react with other species, it is a useful inert tracer. Like Cl^-, nitrate is only transported, thus we do not take it into account. Finally, $Ca—X_2$ and K—X are the two secondary fixed species. The reactions involved are

$$\mathrm{Ca{-}X_2} + 2\mathrm{Na}^+ \leftrightarrows 2\mathrm{Ca}^{2+} + 2\mathrm{Na{-}X} \quad \log(K_{y_1}) = 8.654,$$
$$\mathrm{K{-}X} + \mathrm{Na}^+ \leftrightarrows \mathrm{K}^+ + \mathrm{Na{-}X} \quad \log(K_{y_2}) = 1.6118,$$

where log still means the natural logarithm. Additionally, W is equal to $W_{\mathrm{Na-X}} = 1.1 \times 10^{-3}$ mol/l.

We run experiments with four distinct transport conditions: a pure advective transport, an advective diffusive transport (the original test case proposed by PhreeqC), another advective diffusive transport with more diffusion and a pure diffusive transport; values are reported in Table 1. Moreover, the porosity is equal to 1 everywhere.

The Dirichlet condition on top of column (the injection condition) and the initial conditions are given in Table 2, the Neuman function g_L and the sink source term q are equal to zero.

The transport of Na^+ is retarded with respect to the tracer Cl^-, due to the initial presence of Na^+ fixed on the cation exchanger. Then ions Na^+ are replaced by K^+ and Ca^{2+}. Finally, the remaining fixed K^+ is itself replaced by the injected Ca^{2+} ions.We simulate the geochemical system till $t = 24$ h for the advective diffusive and pure advective transport conditions and till $t = 1000$ h for the pure diffusive transport conditions. The value of the output time step is equal to 720 s and internal time steps are used by SUNDIALS. We consider several meshes, with a number of cells ranging from 20 to 1280 for the finest mesh.

5.1.2. Validation and convergence analysis

For the various configurations, we plot the evolution in time of the concentrations of the four aqueous components at point $x = 0.08$ m and the spatial distribution of the same concentrations at a given time. We present here the results for PhreeqC and for our method, since the method SIA has the same behaviour as our method (both use the same spatial discretization and an implicit scheme). For the four test cases considered (pure advection, advection–diffusion dominated by

Table 1
Transport conditions used for Example 11 from PhreeqC.

	Darcy velocity (in m/s)	Diffusion coefficient (in m^2 s)
Pure advective transport	2.77×10^{-6}	0
Advection-dominated transport	2.77×10^{-6}	5.55×10^{-9}
Diffusion-dominated transport	2.77×10^{-6}	5.55×10^{-8}
Pure diffusive transport	0	5.55×10^{-9}

Table 2
Chemical composition of initial and injecting solutions (unit: mol/l) used for Example 11 from PhreeqC.

Condition	Na^+	Ca^{2+}	K^+	Cl^-
Initial T_0 (mol/l)	4.49×10^{-4}	0	7.5×10^{-4}	0
Injecting C_D (mol/l)	0	0.6×10^{-3}	0	1.2×10^{-3}

advection, advection–diffusion dominated by diffusion, pure diffusion), we get similar results with the three approaches and all methods converge when the number of cells is increased. However, we observe some differences between PhreeqC and our method.

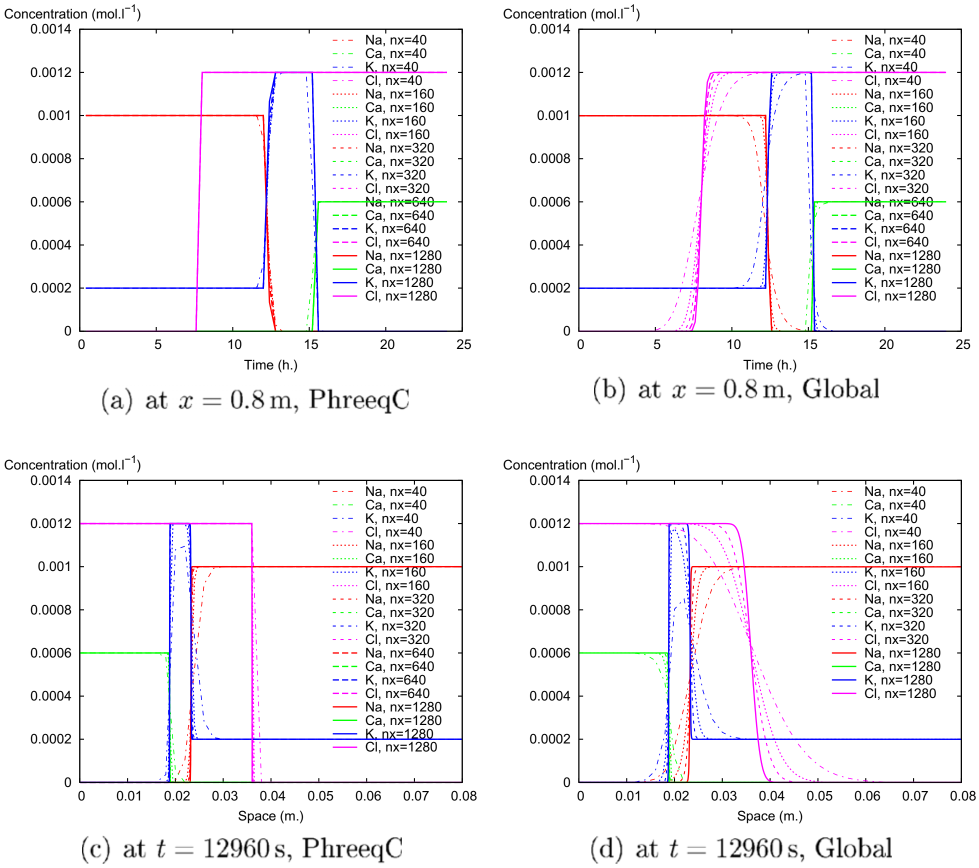
In Fig. 1, we consider the pure advective test case and show the results at point $x = 0.08$ m on graphics (a) and (b) (top) and at time $t = 12{,}960$ s on graphics (c) and (d) (bottom) for different number of cells, with PhreeqC on graphics (a) and (c) (left) and our global method on graphics (b) and (d) (right). We observe a better convergence by PhreeqC, because our method induces some numerical diffusion, arising from our spatial discretization combined with an implicit scheme. In Fig. 2, we consider the advective–diffusive, advection-dominated, test case. Again, results with our method (right) show some numerical diffusion, more than results with PhreeqC (left). In Fig. 3, results for the pure diffusive case show a rapid convergence, without numerical diffusion, for both PhreeqC (left) and our method (right). In order to reduce numerical diffusion, we could replace the basic 1D transport module by a less diffusive one and still use our global method. We could also use an operator splitting approach, with an explicit scheme for advection and an implicit scheme for diffusion and apply our global method for the diffusion equations. However, this would prevent us from using any existing DAE solver and would require implementing control of order, Newton convergence and time step.

5.1.3. Error analysis and efficiency

In order to quantify the accuracy of the results, we introduce the quantity E_N defined by

$$E_N = \max_{1 \leqslant i \leqslant N_c} \left(\sqrt{\frac{\sum_{j=1}^{N_j} \left| c_i^{N_m}(t_j) - c_i^{N_{max}}(t_j) \right|^2}{N_j}} \right),$$

where N_c is the number of components, N_j is the number of measured time steps, $c_i(t_j)$ is the concentration of component i at time t_j and point $x = 0.08$ m, N_m is the number of cells in the current mesh and N_{max} is the number of cells in the reference mesh. Since we do not have experimental or analytical evaluation of the solution, we use a different reference for each method by taking the concentration relative to the finest mesh, so that errors are relative to a method but cannot be directly compared. We run experiments for the four test cases and measure the global CPU time for the whole simulation. In our tests, we



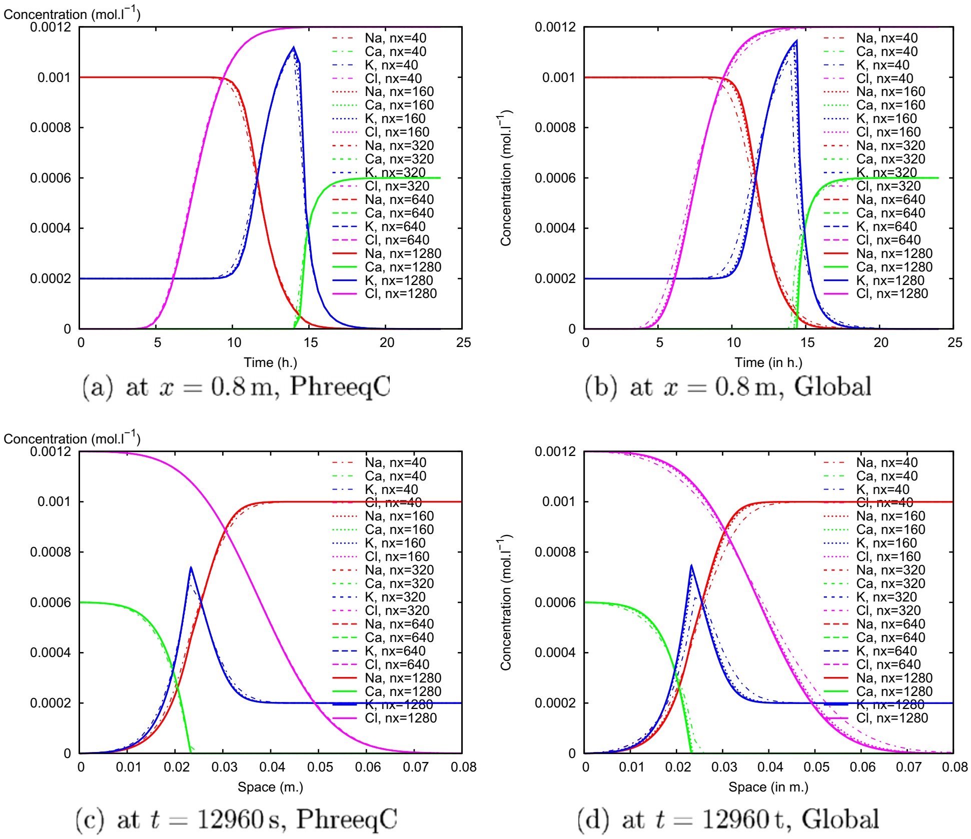
(a) at $x = 0.8$ m, PhreeqC

(b) at $x = 0.8$ m, Global

(c) at $t = 12960$ s, PhreeqC

(d) at $t = 12960$ s, Global

Fig. 1. Results for Example 11 from PhreeqC in a pure advective case, using either PhreeqC or our global method.



(a) at $x = 0.8\,\text{m}$, PhreeqC

(b) at $x = 0.8\,\text{m}$, Global

(c) at $t = 12960\,\text{s}$, PhreeqC

(d) at $t = 12960\,\text{t}$, Global

Fig. 2. Results for Example 11 from PhreeqC in an advection–diffusion case (advection-dominated), using either PhreeqC or our global method.

take $N_{max} = 640$ cells and $N = 20, 40, 80, 160, 320$. In Fig. 4, we plot the error in function of the CPU time for the three methods used (PhreeqC, SIA, our method) and for the four transport conditions.

In view of the results obtained with the other transport conditions, we do not run simulations with the SIA method for the diffusion-dominated case. In all cases tested, SIA is the slowest method and the least accurate one. Therefore, our global method with a Newton method is much more efficient than a SIA approach with a weaker coupling.

In the pure advective case, PhreeqC is clearly the winner. Indeed, an explicit scheme for advection is here very well-suited and much more efficient than an implicit scheme. In the advection-dominated case, PhreeqC is faster for low accuracy but becomes slower for high accuracy with a large number of cells. This trend is even increased in the diffusion-dominated case, where our method is more accurate and faster than PhreeqC. In the pure diffusive case, PhreeqC is really very slow. Here, an explicit scheme for diffusion requires very small time steps for ensuring stability whereas an implicit scheme for diffusion has no stability restriction; moreover, as noted earlier, the implicit advection scheme does not induce too much numerical diffusion.

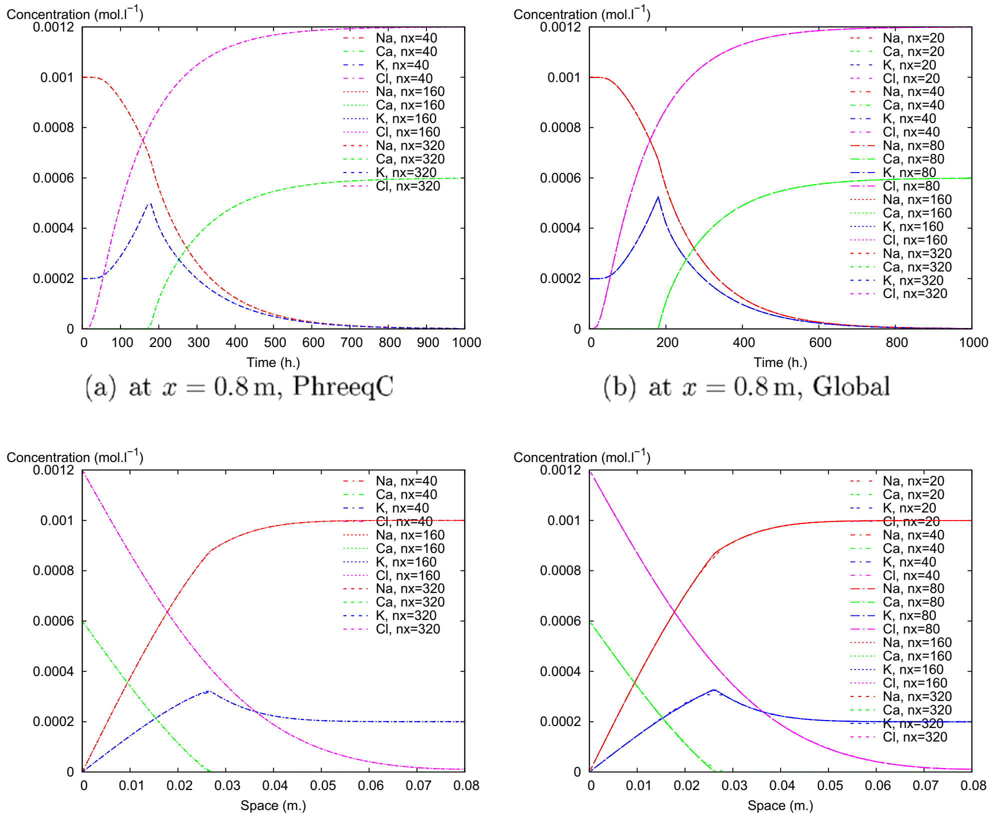
Therefore, these experiments show that an implicit scheme with our global method based on a DAE solver is very efficient in the case of advection–diffusion problems, whereas an explicit scheme is more appropriate in the case of pure advection problems.

5.2. Injection of alcaline water into a column of quartz

5.2.1. Description of test case

This chemical system is the second test case of the qualification plan of Alliances project (Alliances 6.2) [32]. The 1D domain is a column of length $L = 0.4$ m composed of quartz grain (SiO_2). Alcaline water (NaOH) is injected at the top of column involving homogeneous reaction and precipitation of the quartz. The reactions involved here are:

$$\begin{aligned}
&H_2O \leftrightarrows H^+OH^- \quad K_1 = 10^{-14},\\
&H_4SiO_4 \leftrightarrows H_3SiO_4^- + H^+ \quad K_2 = 10^{-9.8},\\
&H_4SiO_4 \leftrightarrows SiO_2(s) + 2H_2O \quad K_3 = 10^{3.6}
\end{aligned} \tag{15}$$



(a) at $x = 0.8\,\mathrm{m}$, PhreeqC

(b) at $x = 0.8\,\mathrm{m}$, Global

(c) at $t = 12960\,\mathrm{s}$, PhreeqC

(d) at $t = 12860\,\mathrm{s}$, Global

Fig. 3. Results for Example 11 from PhreeqC in a pure diffusive case, using either PhreeqC or our global method.

We split the species into three aqueous components Na^+, OH^- and H_4SiO_4, a precipitated species $SiO_2(s)$ and two aqueous secondary species H^+ and $H_3SiO_4^-$. As the medium is composed of quartz grains, the number of moles of $SiO_2(s)$ is considered as strictly positive during the whole simulation. Therefore, in the chemical model (2), we have $N_p = 1$ and one equation of saturation, which means here that the concentration of H_4SiO_4 remains constant.

We consider a pure diffusive transport with $D = 3 \times 10^{-10}\ \mathrm{m^2/s}$. Porosity is taken equal to 1. We assume that initially each cell contains 10 mol of quartz by volume of 1 l. The initial and boundary conditions are given in Table 3.

For this test case, we can write in detail all the chemical equations and we get

$$\begin{aligned}
&c_{H_4SiO_4} = \frac{1}{K_3}, \quad x_{H^+} = \frac{K_1}{c_{OH^-}}, \quad x_{H_3SiO_4^-} = \frac{K_2}{K_1} c_{H_4SiO_4} c_{OH^-},\\
&C_{Na^+} = c_{Na^+}, \quad C_{OH^-} = c_{OH^-} - x_{H^+} + x_{H_3SiO_4^-}, \quad C_{H_4SiO_4} = c_{H_4SiO_4} + x_{H_3SiO_4^-},\\
&T_{Na^+} = C_{Na^+}, \quad T_{OH^-} = C_{OH^-}, \quad T_{H_4SiO_4} = C_{H_4SiO_4} + p_{SiO_2}.
\end{aligned}$$

Thus, we get the same transport equation for both C_{Na^+} and C_{OH^-}, which is

$$\omega \frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial^2 x}. \tag{16}$$

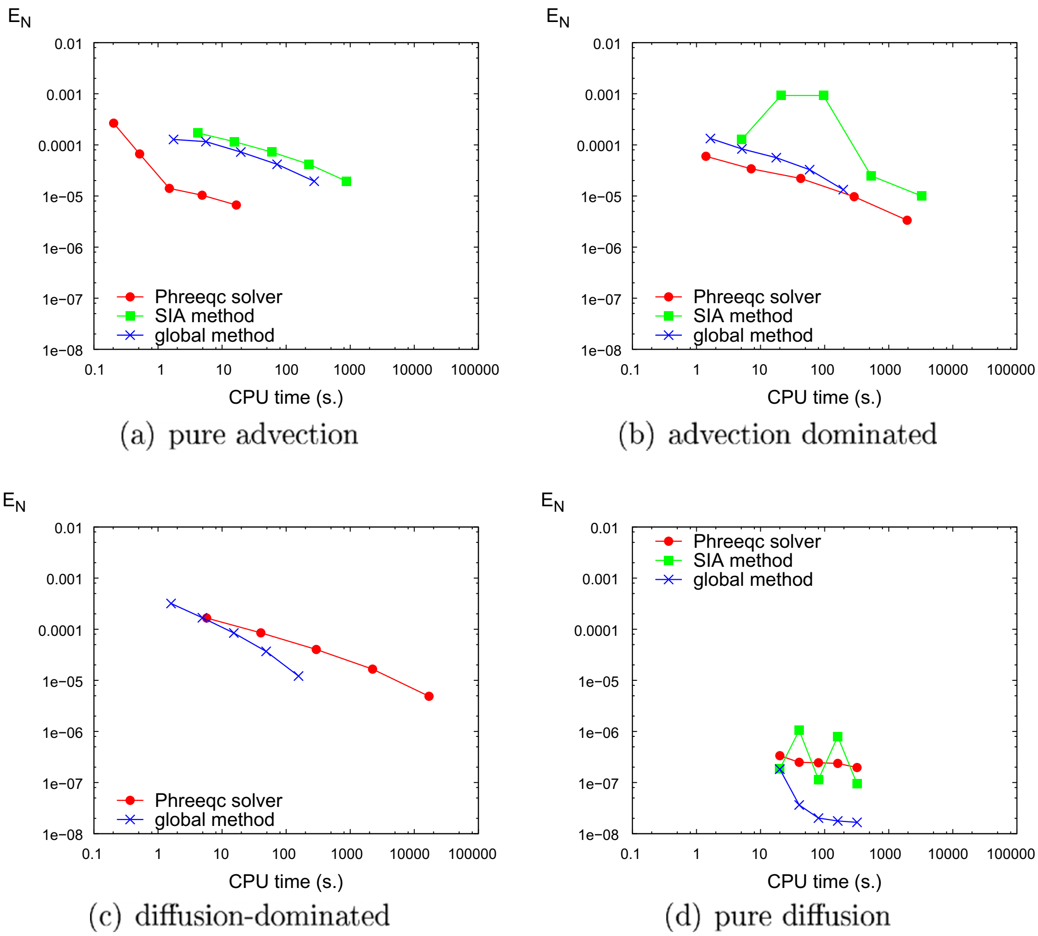
If we take the same initial and boundary conditions, we get, at each point and each time, $C_{Na^+} = C_{OH^-}$, which can be rewritten

$$c_{Na^+} - c_{OH^-} + x_{H^+} - x_{H_3SiO_4^-} = 0.$$

This is nothing but the law of electric neutrality. Here, we do not use this invariant for reducing the size of the system but for validating the method. For the same reason, we do not remove the unknown $c_{H_4SiO_4}$, which is constant.

Table 3
Chemical composition of initial and infiltrating solutions (unit: mol/l) for example Alliances 6.2.

Condition	Na^+	OH^-	H_4SiO_4
Initial T_0	10^{-20}	10^{-20}	10
Injecting C_D	2×10^{-2}	2×10^{-2}	1.62×10^{-2}



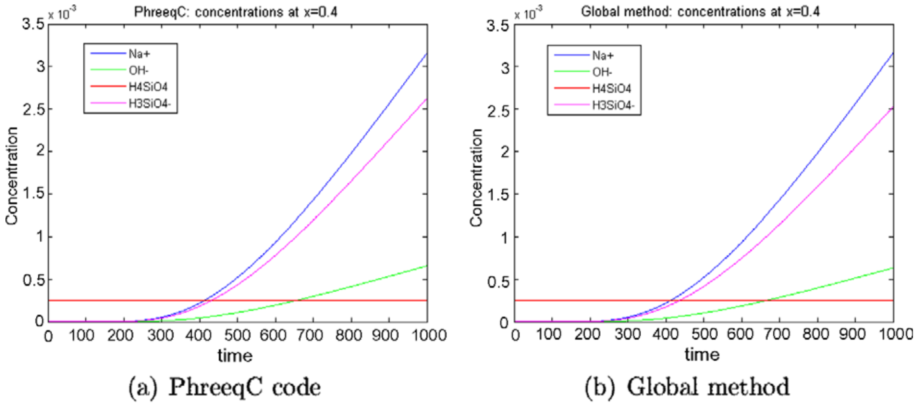
(a) pure advection

(b) advection dominated

(c) diffusion-dominated

(d) pure diffusion

Fig. 4. Results for Example 11 from PhreeqC in four transport conditions, using either PhreeqC or SIA or our global method (SIA was not tested for conditions (c)). Accuracy measured by E_N versus CPU time (logarithmic scale).



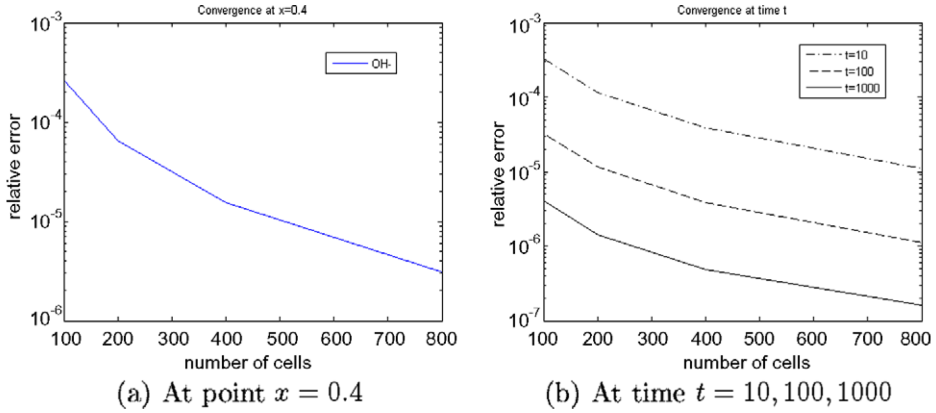
(a) PhreeqC code

(b) Global method

Fig. 5. Results for example Alliances 6.2, using either PhreeqC or our global method. Concentrations of the different components at the end of the column during 1000 days.

5.2.2. Comparison between two methods

We simulate the geochemical system during 1000 days and generate output every day. Results are given for a mesh with 100 cells. We first compare our results with those of PhreeqC. In Fig. 5, we plot the concentrations of the three aqueous components and the secondary species $H_3SiO_4^-$ during the simulated time period at point $x = 0.4$, the end of the column. Our results are in very good agreement with those of PhreeqC.



(a) At point $x = 0.4$ (b) At time $t = 10, 100, 1000$

Fig. 6. Results for example Alliances 6.2, using our global method. Error estimation with the finest mesh $N_{max} = 1600$ as reference. Concentration of OH^- at the end of the column during 1000 days and in the column at times $t = 10, 100, 1000$ days.

This example shows that our chemical model can be used for systems with precipitated species, as long as the quantity remains strictly positive. Up to now, the model cannot deal with precipitation–dissolution, when some quantity becomes zero at some point, because the nonlinear system is no longer differentiable. We plan to apply a semi-smooth Newton method to overcome this difficulty.

5.2.3. Error analysis

Now, we analyze convergence of our global method; again, we take the finest mesh with $N_{max} = 1600$ cells as the reference; we choose to analyze the component OH^- because Na^+ is only an inert tracer, H_4SiO_4 remains constant, whereas OH^- gives directly the concentrations of the secondary species. We compute a relative error at a given point x by

$$E(x) = \left(\frac{\sum_{j=1}^{J} |c_N(x, t_j) - c_{ref}(x, t_j)|^2}{\sum_{j=1}^{J} |c_{ref}(x, t_j)|^2} \right)^{1/2},$$

where J is the number of external time steps, c_N and c_{ref} are the concentrations of OH^- computed, respectively, with N and N_{max} cells; we also compute a relative error at a given time t by

$$E(t) = \left(\frac{\sum_{k=1}^{N} |c_N(x_k, t) - c_{ref}(x_k, t)|^2}{\sum_{k=1}^{N_{max}} |c_{ref}(x_k, t)|^2} \right)^{1/2}.$$

We run simulations for $N = 100, 200, 400, 800, 1600$. In Fig. 6, we plot the error $E(x)$ at point $x = 0.4$, the end of the column (left) and the error $E(t)$ at times $t = 10, 100, 1000$ days (right). We observe a good convergence, at any point and at any time.

Again, these experiments show that our implementation of a global method simulates efficiently reactive transport systems.

6. Conclusion

In this paper, we have presented a unified framework for most of the classical approaches, which allows a fair comparison between them. It appears that SNIA, which is an explicit scheme in time, is efficient for advection-dominated problems for which stability conditions are not too restrictive. In this case, there is no coupling between transport and chemistry, both operators can be black boxes, and chemistry can be solved only at points requiring speciation. It also appears that SIA, which is an implicit scheme in time combined with a nonlinear block-SOR–Newton method, is not efficient because it converges slowly or requires small time steps to converge. A global method based on an ODE solver is also likely to be inefficient because it requires solving too many speciations. Therefore, a global method based on DAE should be preferred, even though it requires handling chemistry equations and cannot rely on a chemistry solver as a black box. The class of DSA is quite efficient but couples strongly transport and chemistry and requires control of the time step and the convergence of Newton iterations. We advocate the use of DAE solvers, which are very efficient in this control, at the price of larger nonlinear systems but with the benefit of a better separation between transport and chemistry operators. Therefore we implement a numerical reactive transport model in a modular and generic way, using external efficient libraries. Currently, we use a Newton–LU method with a direct sparse linear solver but it is easy to interface iterative solvers as well. Also, we currently use a basic spatial discretization for transport, but it is easy to interface another numerical scheme introducing less numerical diffusion.

Our numerical experiments show that the solver PhreeqC, which implements SNIA with operator splitting, is the most efficient method for advection-dominated problems and that SIA is the least efficient method in any case. Our global method is quite robust and can be used for complex geochemistry systems. Our current experiments concern 1D problems and we plan to extend them to 2D and 3D problems [38]. In this case, it might become necessary to switch from direct to iterative solvers because of memory requirements induced by fill-in during factorization of sparse matrices. It will also be useful to compare the method with DSA for 2D and 3D problems. On the one hand, our method induces CPU overhead with the large linear system; on the other hand, accuracy and convergence are controlled efficiently by the DAE solver. Finally, we plan to investigate semi-smooth Newton methods in order to deal with precipitation–dissolution reactions which involve non differentiable equations.

References

[1] C.M. Bethke, Geochemical Reaction Modeling: Concepts and Applications, Oxford University Press, 1996.

[2] A. Walter, E. Frind, D. Blowes, C. Ptacek, J. Molson, Modeling of multicomponent reactive transport in groundwater. 1. Model development and evaluation, Water Resour. Res. 30 (11) (1994) 3137–3148.

[3] A. Zysset, F. Stauffer, T. Dracos, Modeling of chemically reactive groundwater transport, Water Resour. Res. 30 (7) (1994) 2217–2228.

[4] C.I. Steefel, K.T. MacQuarrie, Approaches to modeling of reactive transport in porous media, in: P. Lichtner, C. Steefel, E. Oelkers (Eds.), Reactive Transport in Porous Media, vol. 34 of Rev. Mineral., Mineral. Soc. Am., 1996, pp. 83–125.

[5] G.T. Yeh, V.S. Tripathi, A critical evaluation of recent developments in hydrogeochemical transport models of reactive multichemical components, Water Resour. Res. 25 (1989) 93–108.

[6] J. Carrayrou, R. MosT, P. Behra, Operator-splitting procedures for reactive transport and comparison of mass balance errors, J. Contam. Hydrol. 68 (3–4) (2004) 239–268.

[7] A. Valocchi, R. Street, P. Roberts, Transport of ion-exchanging solutes in groundwater: chromatographic theory and field simulation, Water Resour. Res. 17 (5) (1981) 1517–1527.

[8] C. Miller, L. Benson, Simulation of solute transport in a chemically reactive heterogeneous system: model development and application, Water Resour. Res. 19 (1983) 381–391.

[9] O. Cirpka, R. Helmig, Comparison of approaches for the coupling of chemistry to transport in groundwater systems, Notes Numer. Fluid Dynam. 59 (1997) 102–120.

[10] M. Saaltink, C. Ayora, J. Carrera, A mathematical formulation for reactive transport that eliminates mineral concentrations, Water Resour. Res. 34 (7) (1998) 1649–1656.

[11] M. Saaltink, J. Carrera, C. Ayora, On the behavior of approaches to simulate reactive transport, J. Contam. Hydrol. 48 (2001) 213–235.

[12] S. Molins, J. Carrera, C. Ayora, M.W. Saaltink, A formulation for decoupling components in reactive transport problems, Water Resour. Res. 40 (2004) W10301.

[13] G. Hammond, A. Valocchi, P. Lichtner, Application of Jacobian-free Newton–Krylov with physics-based preconditioning to biogeochemical transport, Adv. Water Resour. 28 (2005) 359–376.

[14] S. KrSutle, P. Knabner, A reduction scheme for coupled multicomponent transport-reaction problems in porous media: generalization to problems with heterogeneous equilibrium reactions, Water Resour. Res. 43 (2007) W03429.

[15] F. Morel, J. Morgan, A numerical method for computing equilibria in aqueous chemical systems, Environ. Sci. Tech. 6 (1) (1972) 58–67.

[16] F. Morel, J. Hering, Principles and Applications of Aquatic Chemistry, second ed., Wiley, New York, 1993.

[17] P.L. Lucille, A. Burnol, P. Ollar, Chemtrap: a hydrogeochemical model for reactive transport in porous media, Hydrol. Process. 14 (2000) 2261–2277.

[18] D. Kirkner, H. Reeves, Multicomponent mass transport with homogeneous and heterogeneous chemical reactions: effect of the chemistry on the choice of numerical algorithm. 1. Theory, Water Resour. Res. 24 (1988) 1719–1729.

[19] N. Bouillard, P. Montarnal, R. Herbin, Development of numerical methods for the reactive transport of chemical species in a porous media: a nonlinear conjugate gradient method, in: Int. Conf. Comput. Method Coupled Probl. Sci. Eng. Coupled Probl. 2005, ECCOMAS Conference, Santorini Island, Greece, 2005.

[20] W. Hundsdorfer, J. Verwer, Numerical solution of time-dependent advection–diffusion–reaction equations, Springer Series in Computational Mathematics, vol. 33, Springer-Verlag, Berlin, 2003.

[21] E. Hairer, G. Wanner, Solving Ordinary Differential Equations. II. Stiff and Differential–Algebraic Problems, No. 14 in SCM, Springer-Verlag, 1991.

[22] D. Parkhurst, C. Appelo, User's guide to PHREEQC (version 2)—a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations, Tech. Rep., U.S. Geological Survey Water-Resources Investigations Report 99-4259, 1999.

[23] C. Appelo, Some calculations on multicomponent transport with cation exchange in aquifers, Groundwater 32 (1994) 968–975.

[24] C. Appelo, D. Postma, Geochemistry Groundwater and Pollution, second ed., A.A. Balkema, 2005.

[25] H. Hoteit, P. Ackerer, R. MosT, J. Erhel, B. Philippe, New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes, Int. J. Numer. Method Eng. 61 (2004) 2566–2593.

[26] D.A. Barry, C.T. Miller, P.J. Culligan, K. Bajracharya, Analysis of split operator methods for nonlinear and multispecies groundwater chemical transport models, Math. Comput. Simul. 43 (1997) 331–341.

[27] J. vander Lee, L. deWindt, CHESS tutorial and cookbook, updated for version 3.0, user's manual, Tech. Rep. LHM/RD/02/13, École des Mines de Paris, France, 2002.

[28] J.F. Kanney, C.T. Miller, C.T. Kelley, Convergence of iterative split operator approaches for approximating nonlinear reactive transport problems, Adv. Water Resour. 26 (2003) 247–261.

[29] M. Saaltink, J. Carrera, C. Ayora, A comparison of two approaches for reactive transport modelling, J. Geochem. Explor. 69–70 (2000) 97–101.

[30] J. Ortega, W. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Computer Science and Applied Mathematics, Academic Press, 1970.

[31] J. Carrayrou, ModTlisation du transport de solutTs rTactifs en milieu poreux saturT, ThFse de doctorat, UniversitT Louis Pasteur, Strasbourg, 2001.

[32] P. Montarnal, A. Dimier, E. Deville, E. Adam, J. Gaombalet, A. Bengaouer, L. Loth, C. Chavant, Coupling methodology within the software platform Alliances, in: E.O.M. Papadrakakis, B. Schrefler (Eds.), Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering COUPLED PROBLEMS 2005, CIMNE, Barcelona, 2005.

[33] A. Holstad, A mathematical and numerical model for reactive fluid flow systems, Comput. Geosci. 4 (2000) 103–139.

[34] L. Amir, M. Kern, Newton–Krylov methods for coupling transport with chemistry in porous media, in: P. Binning, P. Engesgaard, H. Dahle, G. Pinder, W.G. Gray (Eds.), XVI International Conference on Computational Methods in Water Resources (CMWR XVI), 2006. <http://proceedings.cmwr-xvi.org/>.

[35] H. Shen, N. Nicolaidis, A direct substitution method for multicomponent solute transport in groundwater, Groundwater 35 (1) (1997) 67–78.

[36] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R.Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Trans. Math. Softw. 31 (2005) 363–396 (also available as LLNL Technical Report UCRL-JP-200037).

[37] R. Eymard, T. Galloudt, R. Herbin, Finite volume methods, Handbook of Numerical Analysis, vol. VII, North-Holland, Amsterdam, 2000, pp. 713–1020.

[38] C. de Dieuleveult, An efficient and robust global method for reactive transport, Ph.D. Thesis, University of Rennes 1, December 2008.