Comput Geosci (2010) 14:465–481
DOI 10.1007/s10596-009-9162-x

ORIGINAL PAPER

A global method for coupling transport with chemistry in heterogeneous porous media

Laila Amir · Michel Kern

Received: 27 November 2008 / Accepted: 2 September 2009 / Published online: 23 September 2009

Abstract Modeling reactive transport in porous media, using a local chemical equilibrium assumption, leads to a system of advection–diffusion PDEs coupled with algebraic equations. When solving this coupled system, the algebraic equations have to be solved at each grid point for each chemical species and at each time step. This leads to a coupled non-linear system. In this paper, a global solution approach that enables to keep the software codes for transport and chemistry distinct is proposed. The method applies the Newton–Krylov framework to the formulation for reactive transport used in operator splitting. The method is formulated in terms of total mobile and total fixed concentrations and uses the chemical solver as a black box, as it only requires that one be able to solve chemical equilibrium problems (and compute derivatives) without having to know the solution method. An additional advantage of the Newton–Krylov method is that the Jacobian is only needed as an operator in a Jacobian matrix times vector product. The proposed method is tested on the MoMaS reactive transport benchmark.

Keywords Geochemistry · Transport in porous media · Newton–Krylov methods · Advection-diffusion-reaction equations

L. Amir · M. Kern (✉)
INRIA Paris-Rocquencourt, B.P. 105,
78153 Le Chesnay Cedex, France
e-mail: Michel.kern@inria.fr

L. Amir
ITASCA Consultants, S.A., 64, Chemin des Mouilles,
69134 Ecully, France
e-mail: Laila.Amir@inria.fr

Mathematics Subject Classifications (2000) 76V05 · 65M99

1 Introduction

The simulation of multi-species reacting systems in porous media is of importance in several different fields: for computing the near field in nuclear waste simulations, in the treatment of bio-remediation, in CO_2 sequestration simulations, and in the evaluation of underground water quality.

This work deals with numerical methods for solving coupled transport and chemistry problems. The transport of solutes in porous media is described by partial differential equations of advection–diffusion type, whereas multi-species chemistry involves the solution of ordinary differential equations (if the reactions are kinetic) or nonlinear algebraic equations (if local equilibrium is assumed). After discretization, one is led to a system of nonlinear equations, coupled by the unknowns for all chemical species at all grid points.

After the influential paper by Yeh and Tripathi [46], operator splitting methods, where transport and chemistry are solved for separately at each time step (possibly iterating to convergence), became the methods of choice. Some representative papers where operator splitting methods are used are [4, 21, 29, 38, 39, 41]. Operator splitting methods are easy to implement, and the splitting errors can be controlled by carefully restricting the time step. On the other hand, the time-step restriction can become their main drawback, as it can be difficult to get the fixed point iteration to converge for more difficult problems.

More recently, global methods have become popular, due to the increase in computing power now available. In this approach, the full non-linear system is solved in one step, usually by some form of Newton's method. Most papers use the direct substitution approach (see [14, 17]), where one *substitutes* the chemical equations in the transport equations. On the other hand, the problem can also be put in the form of differential algebraic equations (DAE), enabling the use of powerful software (see [11]). Finally, the chemical equations can be eliminated locally, and a system involving transport equations, with a source term coming from the reactions, has to be solved. This approach is taken in [25, 26], where, additionally, a reduction method leads to a smaller system. Most of the papers quoted above employ a Newton method for solving the nonlinear system at each time step, with the difficulty that the Jacobian matrix has to be computed, stored, and factored. This can become problematic for large problems, and Hammond et al. [17] have used the Jacobian-free Newton–Krylov method, where the Newton correction is solved for by an iterative method. The Jacobian is only needed through the computation of a directional derivative. The method keeps the fast convergence of Newton's method, while only requiring Jacobian matrix–vector products, and these can be approximated by finite differences.

The method presented in this paper is a global method where the chemical equations are eliminated locally, leading to a nonlinear system where the transport and chemistry subsystems remain separated. Thus, the residual can be evaluated by calling separately written transport and chemistry modules. The system is then solved by a Newton–Krylov method, and it will be shown how the Jacobian matrix–vector product can also be computed by the same module. Thus, the main contribution of this paper is to show that a global method can be implemented while still keeping transport and chemistry modules separated. This property will be referred to as using "black-box solvers." As the chemical equilibrium equations are not substituted in the transport equations, the transport and chemistry parts of the nonlinear residual are easily identified, and can each be computed by calling on standard solution modules.

An outline of the paper is as follows. In Section 2, the chosen model is explained, and the methods used for solving the (non-reactive) transport part and the chemical equilibrium system are detailed. Section 2.3 shows how we obtain the coupled model. Couple formulations and coupling algorithms are the subject of Section 3, beginning with a review of existing methods, while our approach is presented in Section 3.2. Numerical results, in particular, experience with the MoMaS benchmark, are shown in Section 4.

2 Reactive transport equations

In this work, the transport of several reacting species in a single phase flow through a porous medium is considered. The species can react both between themselves and with the porous matrix. In this section, the numerical methods used to solve the individual subsystems of the coupled problem will be described.

2.1 Transport model

The transport of a single species through a porous medium (a domain $\Omega \subset \mathbf{R}^d$, with $d = 1, 2$ or 3), with porosity ϕ, in a known Darcy field u, subject to dispersion and molecular diffusion, follows the linear advection–dispersion equation

$$\phi \frac{\partial c}{\partial t} + L(c) = q, \quad \text{in } \Omega, \tag{1}$$

where

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

is the transport operator and q is a source term. The diffusion–dispersion tensor D is given by

$$D = d_e I + |u| \big(\alpha_L E(u) + \alpha_t (I - E(u)) \big), \quad E_{ij}(u) = \frac{u_i u_j}{|u|^2},$$

where d_e is the molecular diffusion coefficient and α_L (resp. α_t) is the longitudinal (resp. transverse) dispersivity coefficient.

In this work, we restrict ourselves to a one-dimensional problem, so that the transport equation over a bounded interval $\Omega =]0, L[$ can be written as

$$\phi \frac{\partial c}{\partial t} + \frac{\partial}{\partial x} \left(-D \frac{\partial c}{\partial x} + uc \right) = q, \quad 0 < x < L, \quad 0 < t < T, \tag{2}$$

where the porosity ϕ and the diffusion–dispersion coefficient D can both depend on space. Because the flow is assumed compressible, the velocity u is taken to be a constant.

The initial condition is $c(x, 0) = c_0(x)$ and, in view of the applications, the boundary conditions are a Dirichlet condition (given concentration) $c(0, t) = c_d(t)$ at the left boundary ($x = 0$) and zero diffusive flux $\dfrac{\partial c}{\partial x} = 0$

at the right boundary ($x = L$). More general boundary conditions could easily be accommodated.

2.1.1 Discretization in space

We treat the space and time discretization separately, as we will use different time discretizations for the different parts of the transport operator.

For space discretization, a cell-centered finite-volume scheme will be used, see, for instance, [13]. The interval $[0, L]$ is divided into N_g intervals $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ of length h_i, where $x_{\frac{1}{2}} = 0, x_{N_g+\frac{1}{2}} = L$. For $i = 1, .., N_g$, denote by x_i the center and $x_{i+1/2}$ the right end of element i. Finally, denote by $c_i, i = 1, .., N_g$ the approximate solution in cell i.

Equation 2 is written in the form

$$\phi \frac{\partial c}{\partial t} + \frac{\partial \varphi}{\partial x} = q, \tag{3}$$

where the flux $\varphi(x, t) = -D \frac{\partial c}{\partial x} + uc$ has been split as the sum of a diffusive flux $\varphi_d = -D \frac{\partial c}{\partial x}$ and an advective flux $\varphi_a = uc$.

Equation 3 is integrated over a cell $]x_{i-1/2}, x_{i+1/2}[$, giving

$$\phi_i h_i \frac{dc_i}{dt} + \varphi_{d,i+\frac{1}{2}} + \varphi_{a,i+\frac{1}{2}} - \varphi_{d,i-\frac{1}{2}} - \varphi_{a,i-\frac{1}{2}} = h_i q_i,$$
$$i = 2, \ldots, N_g. \tag{4}$$

The flux approximations required to close the system are provided by finite differences. The diffusive flux needs a value for the diffusion coefficient, which is taken as the harmonic average (as done in mixed finite element methods):

$$\varphi_{d,i+\frac{1}{2}} = -D_{i+\frac{1}{2}} \left(\frac{c_{i+1} - c_i}{h_{i+\frac{1}{2}}} \right), \tag{5}$$

with

$$D_{i+\frac{1}{2}} = \frac{2D_i D_{i+1}}{D_i + D_{i+1}}, \quad D_{\frac{1}{2}} = D_1, \quad D_{N_g+\frac{1}{2}} = D_{N_g} \quad \text{and}$$
$$h_{i+\frac{1}{2}} = \frac{h_i + h_{i+1}}{2}$$

For the advective flux, an upwind approximation is used, so that (assuming $u > 0$) $\varphi_{a,i+\frac{1}{2}} = uc_i$

These approximations are corrected to take into account the boundary conditions, both at $x = 0$ and at $x = L$. The semi-discrete system can be summarized by the finite dimensional system

$$M \frac{dc}{dt} + Lc = q + g, \tag{6}$$

where $c \in \mathbf{R}^{N_g}$ now represents the vector of cell concentrations, $L \in \mathbf{R}^{N_g,N_g}$ is the matrix form of the transport operator, $M \in \mathbf{R}^{N_g,N_g}$ is a mass matrix accounting for variable porosity and mesh size, $q \in \mathbf{R}^{N_g}$ is a given source term, and $g \in \mathbf{R}^{N_g}$ represents the effects of the boundary conditions.

2.1.2 Time discretization

Let us denote by Δt the time step (taken constant for simplicity) used to discretize the time interval $[0, T]$ and denote by c_i^n the (approximate) value of $c_i(n\Delta t)$. The first and most straightforward alternative is to discretize Eq. 6 by the backward Euler method, see for instance [3]. This is the method that is used in Section 3 to keep the description simple, but it is not the recommended method, as it leads to an overly diffusive scheme.

Better alternatives are obtained by exploiting the structure of the transport operator and by using different time discretizations for the advective and for the diffusive parts. Specifically, the diffusive terms should be treated implicitly, and the advective terms are better handled explicitly.

If this idea is applied directly to Eq. 6, the resulting fully discrete scheme is only stable under a Courant–Friedrichs–Lewy (CFL) condition $u\Delta t \leq \max_i h_i$. As this may be too severe a restriction (some of our applications require integration over a very large time interval), an alternative is to use an operator splitting scheme, as proposed by Siegel et al. [43] (see also [20, 31]). In this work, splitting is used only within the (linear) transport step, but recent papers by Kačur et al. [16, 22] apply splitting directly to a transport with sorption model by solving (analytically) a nonlinear advection step, followed by a nonlinear diffusion step. This is different from operator splitting as used in geochemical models, as the chemistry terms are solved for together with the transport terms.

The splitting scheme works by taking several small time steps of advection, controlled by a CFL condition, within a large time step of diffusion. The scheme has been shown to be unconditionally stable, and has a good behavior in advection-dominated situations.

More precisely, the time step Δt will be used as the diffusion time step; it is divided into M time steps of advection Δt_c such that $\Delta t = M\Delta t_c$, where $M \geq 1$. The advection time step will be controlled by the CFL condition. Equation 3 will be solved over the time step

$[t^n, t^{n+1}]$ by first solving the advection equation $\phi \frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(uc) = 0$ over M steps of size Δt_c each, and then solving the diffusion equation $\phi \frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(-D \frac{\partial c}{\partial x}) = q$ starting from the value at the end of the advection step.

Advection step The interval $[t^n, t^{n+1}]$ is divided into M intervals $[t^{n,m}, t^{n,m+1}]$, $m = 0, \dots M-1$, where $t^{n,0} = t^n$, $t^{n,M} = t^{n+1}$. Denote $c_i^{n,m}$ as the approximate concentration c at time $t^{n,m}$ and $c^{n,0} = c^n$. The advection equation is discretized in time using the explicit Euler method to obtain

$$\begin{cases} \phi_i(\dfrac{c_i^{n,m+1} - c_i^{n,m}}{\Delta t_c}) + u\left(\dfrac{c_i^{n,m} - c_{i-1}^{n,m}}{h_{i-1/2}}\right) = 0, \quad i = 2, \dots, N_g, \\ c_1^{n,m+1} = c_g(t^{n,m+1}) \end{cases} \qquad m = 0, \dots, M-1. \tag{7}$$

Diffusion step The diffusion part is discretized by an implicit Euler scheme, starting from $c_i^{n,M}$:

$$\begin{aligned} &-\frac{D_{i-\frac{1}{2}}}{h_{i-\frac{1}{2}}}\Delta t c_{i-1}^{n+1} + \left(\phi_i h_i + \frac{D_{i+\frac{1}{2}}}{h_{i+\frac{1}{2}}}\Delta t + \frac{D_{i-\frac{1}{2}}}{h_{i-\frac{1}{2}}}\Delta t\right) c_i^{n+1} \\ &-\frac{D_{i+\frac{1}{2}}}{h_{i+\frac{1}{2}}}\Delta t c_{i+1}^{n+1} = \phi_i h_i c_i^{n,M} + q_i h_i \Delta t, \quad i = 2, \dots, N_g - 1 \end{aligned} \tag{8}$$

As above, two equations accounting for the boundary conditions must be added.

2.2 Chemical equations

The chemical model is described in this section. In this study, we assume a local chemical equilibrium at every point, which means that the chemical phenomena occur on a much faster scale than transport phenomena. This is a common modeling assumption for reactive transport in porous media, at least when the only reactions considered are aqueous phase and sorption reactions (these are "sufficiently fast" reactions according to Rubin [34]). This would not be the case if mineral dissolution was taken into account, as these reactions typically need kinetic models.

Consider a set of N_e chemical species $(X_j)_{j=1,\dots,N_e}$ linked by N_r reactions

$$\sum_{j=1}^{N_e} \nu_{ij} X_j \leftrightarrows 0, \quad i = 1, \dots, N_r,$$

where ν is the stoichiometric matrix. Following Morel [32], we distinguish between *component* and *secondary* species by extracting a full rank matrix from ν. Component species are a minimal subset of the species such that the other secondary species can be written in terms of them (in a unique way). Each secondary species gives rise to a reaction that expresses how it is formed in terms of the components, and to a mass action law that gives the value of its *activity* in terms of the component activities. Similarly, each component gives rise to a conservation equation, expressing how the given total concentration of such a component is distributed among the component itself and the secondary species.

Additionally, in the context of reactive transport, it is required to know how the species are split between those that are in solution and those that have been adsorbed on the solid matrix (in this paper, we do not take precipitation into account). We thus introduce (with, obviously, $N_e = N_c + N_s + N_x + N_y$)

- Mobile components c_j, $j = 1, \dots, N_c$
- Fixed components s_j, $j = 1, \dots, N_s$
- Mobile secondary species x_j, $i = 1, \dots, N_x$
- Fixed secondary species y_j, $i = 1, \dots, N_y$

We have identified the name of the species with their concentrations, and we assume an ideal solution (activities and concentrations are identified). Mobile secondary species x can be expressed as linear combinations of mobile components while secondary fixed species depend on both mobile and fixed components. Therefore, the mass action laws are written as

$$\begin{aligned} x_i &= K_{xi} \prod_{j=1}^{N_c} c_j^{S_{ij}}, \quad i = 1, \dots, N_x, \\ y_i &= K_{y_i} \prod_{j=1}^{N_c} c_j^{A_{ij}} \prod_{j=1}^{N_s} s_j^{B_{ij}}, \quad i = 1, \dots, N_y, \end{aligned} \tag{9}$$

where K_{xi} and K_{y_i} are the equilibrium constants, and S_{ij}, A_{ij}, and B_{ij} are the entries of the stoichiometric matrices $S \in \mathbf{R}^{N_c \times N_x}$, $A \in \mathbf{R}^{N_c \times N_y}$, and $B \in \mathbf{R}^{N_s \times N_y}$.

Mass conservation for each component is expressed in the form

$$c + S^T x + A^T y = T, \quad s + B^T y = W, \tag{10}$$

where T_j is the total concentration of the mobile component j and W_j is the total concentration of the fixed component j (T and W are vectors of size N_c and N_s, respectively). In the case of ion exchange, the second mass conservation equation is simply $B^T y = W$, and W is the cationic exchange capacity of the porous matrix (see Appelo and Postma [2]). As will be seen later, in the context of coupled transport and chemistry, T_j is given by the transport model and W is constant. In a closed chemical system, T_j would be part of the data (total concentration of the components).

Due to the wildly different orders of magnitude of the concentrations that are commonly encountered, the chemical problem is reformulated by using as main unknowns the logarithms of the concentrations. This has the added advantage that concentrations are automatically positive, and has become the standard way to solve the problem [28]. An additional advantage has been pointed out by Samper et al. [41]: by taking the logarithms of the concentrations as unknowns, the Jacobian of the nonlinear system is symmetric, and with a proper choice of the component species, it can be shown to be diagonally dominant, and thus, nonsingular. The symmetry can also be seen in Eq. 16 below. Let $\log u$ be the vector with entries $\log u_i$, where u_i are the entries of vector u. Equation 9 can then be rewritten as a linear system

$$\begin{aligned} \log x &= \log K_x + S \log c \\ \log y &= \log K_y + A \log c + B \log s \end{aligned} \tag{11}$$

The nonlinear system of Eqs. 10 and 11 forms what will be called the *chemical problem*. In the sequel, it will be assumed that this problem always has a (positive) solution (c, s), for all feasible values of the data T and W. This is true in our simplified setting because the chemical equilibrium problem is a consequence of the minimization of the Gibbs free energy, which can be shown to be convex in the absence of minerals (see [42]).

To solve the chemical problem, a variant of Newton's method is used. As is well known, Newton's method is not always convergent, unless the initial point is sufficiently close to the solution. However, and this is especially true in the context of a coupled code where the chemical problem will be solved repeatedly, it is essential to ensure that the solver "never" fails. We have found that using a globalized version of Newton's method (using a line search, cf. [23]) was effective in making the algorithm converge from an arbitrary initial guess. In order to get a smaller system, the secondary concentrations are eliminated, and the system to be solved involves only $lc = \log c \in \mathbf{R}^{N_c}$ and $ls = \log s \in \mathbf{R}^{N_s}$. Define the function $H : \mathbf{R}^{N_c+N_s} \to \mathbf{R}^{N_c+N_s}$ by

$$H \begin{pmatrix} lc \\ ls \end{pmatrix} = \begin{pmatrix} \exp(lc) + S^T \exp(\log K_x + Slc) + A^T \exp(\log K_y + Alc + Bls) \\ \exp(ls) + B^T \exp(\log K_y + Alc + Bls), \end{pmatrix} \tag{12}$$

where the notation $\exp(v)$ for a vector v means the vector with elements $\exp(v_j)$, then Eqs. 10 and 11 are equivalent to:

$$H \begin{pmatrix} lc \\ ls \end{pmatrix} = \begin{pmatrix} T \\ W \end{pmatrix}. \tag{13}$$

This is the nonlinear system that is to be solved for lc and ls, given T and W. The secondary concentrations can then be computed from Eq. 11.

When solving the coupled problem, the distribution of the species between their mobile form and their fixed form will be needed. The individual concentrations must still be solved for, but they are intermediate quantities. Once the component concentrations have been computed as described in the previous paragraph, one can compute for each species its mobile part C_j and its fixed part F_j by

$$C = c + S^T x, \qquad F = A^T y. \tag{14}$$

Note that, by definition, the relationship $T = C + F$ holds.

In the formulation to be presented below, it will be convenient to represent the mapping from the vector of total concentrations to the vector of fixed concentrations. This mapping, denoted by ψ, is defined by first solving the chemical problem Eq. 13, then computing F by Eq. 14. More precisely,

$$\begin{aligned} \psi : \quad & \mathbf{R}^{Nc} \to \mathbf{R}^{Nc} \\ & T \mapsto \psi(T) = A^T y, \end{aligned} \tag{15}$$

where Eq. 13 is first solved for lc and ls, then y is computed by Eq. 11.

It is important to keep in mind that computing $\psi(T)$ means solving the chemical system (plus some simple computations), as this will be the most expensive part when evaluating the residual of the coupled system (see Eq. 26 in Section 3.2).

As this will be useful later on, the computation of the Jacobian of ψ is outlined here. Assume $\psi(T)$ itself has been computed, so that the nonlinear system Eq. 13 has been solved. First, the Jacobian matrix of H should also be computed as part of the solution process. This is almost certainly needed for solving the chemical problem if Newton's method is used. Differentiating Eq. 12 leads to:

$$H'\begin{pmatrix} lc \\ ls \end{pmatrix} = \begin{pmatrix} \operatorname{diag}(\exp(lc)) & 0 \\ 0 & \operatorname{diag}(\exp(ls)) \end{pmatrix} + \begin{pmatrix} S^T & A^T \\ 0 & B^T \end{pmatrix} \begin{pmatrix} \operatorname{diag}(x) & 0 \\ 0 & \operatorname{diag}(y) \end{pmatrix} \begin{pmatrix} S & 0 \\ A & B \end{pmatrix}, \tag{16}$$

where $\operatorname{diag}(v)$ is the diagonal matrix with vector v along the diagonal. Then, by an application of the implicit function theorem (see for instance [35]), and by differentiating Eq. 11, there comes

$$\psi'(T) = A^T \operatorname{diag}(y) \begin{pmatrix} A & B \end{pmatrix} \left(H'\begin{pmatrix} lc \\ ls \end{pmatrix} \right)^{-1} \begin{pmatrix} I \\ 0 \end{pmatrix}. \tag{17}$$

It should be stressed that the Jacobian of H is needed to computed the Jacobian of ψ (inverting it is straightforward, as this will usually be a small matrix). This may prove problematic in practice for several reasons. First, the chemical solver may not give access to the Jacobian, even if it is used internally. This is a limitation to the "black-box" approach. Second, for more realistic chemical models, including non-ideal chemistry, and taking minerals into account, computing the Jacobian may be much more difficult than the fairly simple computation outlined above. As a last resort, one could compute the Jacobian by finite differences, but it will be argued in Section 3.2 that, for this particular problem, the analytical computation is more efficient.

2.3 Coupled transport and chemistry

The starting point for the coupled model is the following set of equations for the total, mobile, and fixed concentrations of each component

$$\begin{cases} \phi \dfrac{\partial C_j}{\partial t} + \phi \dfrac{\partial F_j}{\partial t} + L(C_j) = 0 & j = 1, \dots, N_c \\ \dfrac{\partial W_j}{\partial t} = 0, & j = 1, \dots, N_s \end{cases} \tag{18}$$

These equations can be derived from the individual conservation equations by standard algebraic manipulations, see, for instance, Yeh and Tripathi [46]. It is the formulation given in the benchmark definition [6], see also [11, 37]. The second equation is obvious, as W_j was taken as a constant (at each point in space).

Taking into account the relation $T_j = C_j + F_j$, $j = 1, \dots, N_c$ noted above, the first equation of the system is equivalent to

$$\phi \frac{\partial T_j}{\partial t} + L(C_j) = 0 \quad j = 1, \dots, N_c, \tag{19}$$

where T_j is the total concentration, C_j is the total mobile concentration, and F_j is the total fixed concentration for component j.

From now on, L will denote the discretized transport operator, as defined in Eq. 6. Each unknown concentration depends on both the grid point index and the chemical species index. We will use a notation inspired from Matlab. For a concentration u_{ij}, where $i \in [1, N_g]$ represents the spatial index and $j \in [1, N_c]$ represents the chemical index, we shall denote by

– $u_{:,j}$ the column vector of concentrations of species j at all grid points
– $u_{i,:}$ the row vector of concentrations of all chemical species in grid cell x_i

The unknowns will be numbered first by chemical species, then by grid points. Thus, all the unknowns for a single grid point are numbered contiguously.

The coupled problem is obtained by putting together Eq. 19 above with the definition of the chemical solution operator ψ, defined in Eq. 15 (the subscript T denotes transposition):

$$\begin{cases} M \dfrac{\partial C_{:,j}}{\partial t} + M \dfrac{\partial F_{:,j}}{\partial t} + L(C_{:,j}) = g_{:,j} & j = 1, \dots, N_c \\ T_{ij} = C_{ij} + F_{ij}, & i = 1, \dots, N_g,\ j = 1, \dots, N_c \\ F_{i,:} = \psi(T_{i,:}^T)^T, & i = 1, \dots, N_g \end{cases} \tag{20}$$

This system is then discretized in time to obtain the fully discrete coupled nonlinear system. In this work, we restrict to a simple backward Euler scheme with constant step size, noting that other, more sophisticated strategies are obviously possible (in particular, an adaptive step-size is essential for efficiency). Denoting time indexes by a superscript, the following system is obtained

$$\begin{cases} M\dfrac{C^{n+1}_{:,j} - C^{n}_{:,j}}{\Delta t} + M\dfrac{F^{n+1}_{:,j} - F^{n}_{:,j}}{\Delta t} + L(C^{n+1}_{:,j}) = g_{:,j} & j = 1, \dots, N_c \\ T^{n+1}_{ij} = C^{n+1}_{ij} + F^{n+1}_{ij} & i = 1, \dots, N_g,\ j = 1, \dots, N_c \\ F^{n+1}_{i,:} = \psi((T^{n+1}_{i,:})^T)^T & i = 1, \dots, N_g \end{cases} \tag{21}$$

This is the system to be solved at each time step.

3 Formulation and coupling algorithms

The formulation of reactive transport seen above gives rise to a large system of nonlinear equations. For complex problems, its solution will require a large amount of computer time, which makes it important to choose an appropriate method. In this section, several formulations and approaches that have appeared in the literature will be reviewed.

Thanks to the relationship $T = C + F$, it is easy to eliminate one of the three variables, and this leads to different formulations for the coupled problem, depending on which variables are kept in the transport equation. We keep the system continuous in time, as it makes the notation somewhat lighter, but the same manipulations can obviously be done at the discrete level too.

According to Saaltink et al. [37], see also Salignac [40], one can derive three main formulations from the system given in Eq. 20:

- Formulation (TC) where T is the principal variable and C is the transported variable

$$M\frac{\partial T_{:j}}{\partial t} + L(C_{:j}) = g_{:,j} \tag{22}$$

  This is the formulation used by Erhel et al. in [9, 11], as it lends itself best to a DAE-type algorithm. It is not convenient for our purpose, as the transport equation then involves both T and C, and is thus not easily used with an existing transport solver.
- Formulation (TT) where T is the principal variable and T is the transported variable

$$M\frac{\partial T_{:j}}{\partial t} + L(T_{:j}) + L(F_{:j}) = g_{:,j} \tag{23}$$

  This seems to be the least satisfactory formulation, as the transport operates on the fixed species, and for this reason, it will not be considered further.
- Formulation (CC) where C is the principal variable and C is the transported variable

$$M\frac{\partial C_{:j}}{\partial t} + M\frac{\partial F_{:j}}{\partial t} + L(C_{:j}) = g_{:,j} \tag{24}$$

  This is formulation 4 in Saaltink et al. [37], and is the formulation chosen below. It has been reported that this formulation is the least suitable for use in an operator split algorithm because C and F are used at different time levels (to compute the data for the chemical problem). When this formulation is used in a global method, this should not matter as much, as the iterations are run to convergence, and both values should eventually get close to their limits.

Formulation (CC) will be used in the rest of the paper because it takes the form of a standard transport operator, with a source term coming from the chemical part. Its structure is closely related to the system describing single-species transport with sorption, as seen, for instance, in [16] or [21], with the main differences that the unknown is a vector of concentration and mostly that what plays the role of the sorption isotherm is the implicitly defined function Ψ introduced in Eq. 15.

3.1 Review of former approaches

At each time step, the system given by Eq. 21 (one transport equation for each component and one chemical system for each grid point) forms a large nonlinear system, whose size is the number of components times the number of grid points. This system has traditionally been solved by a sequential two-step approach, as reviewed below (cf. [46]). However, this method suffers from several defects: it may severely restrict the step size to ensure convergence, and if used non-iteratively, it is only first order in time, which may introduce additional errors (cf. [4]). Due to its quadratic convergence rate, Newton's method would be an ideal candidate for solving the system. On the other hand, a practical difficulty has to be reckoned with: Newton's method requires the solution of a linear system with the Jacobian matrix at each iteration step. In realistic situations, it will not be possible to store, much less factor, the Jacobian matrix. As will be seen in Section 3.2, this difficulty can be overcome by resorting to an iterative method for solving the linear system.

3.1.1 Sequential approach

The sequential approach consists of separately solving the chemical equations and the transport equations. The method has been used in numerous papers: see, for instance, [46], and also [4, 21, 29, 37], or [30]. At each iteration, a transport equation for each component is solved first, with a source term given by the (change in) fixed concentration at the previous iteration. This total mobile concentration will be added to a total fixed concentration computed in the previous iteration, to obtain the total used as data for solving a chemical problem at each grid point. These steps are then iterated until convergence.

In the geochemical literature, this is known as an operator-splitting approach (usually called standard iterative approach, or SIA), but it is more properly a block Gauss–Seidel method on the coupled system, as each subsystem is solved alternatively. The method is quite appealing, as it is easy to implement starting from separate transport and chemistry codes, and can provide good accuracy if implemented carefully, as shown in the references above. As will be seen below, these advantages can be retained in the Newton–Krylov framework.

The standard non-iterative approach (SNIA) is the case where only one iteration of the method is carried out at each time step. In that case, splitting errors can become important, and the method is not really suitable for difficult problems.

The SIA approach does not suffer from splitting errors if the tolerance is small enough, but it may require a small time step to obtain convergence in the case of stiff problems. The main drawback of the method is thus that the size of the time step is used to control convergence, and not based on the physical character of the solution.

3.1.2 Direct substitution approach

As computing power increased, it was recognized that the operator splitting methods of the previous sections could not satisfactorily handle difficult problems, and more tightly coupled methods came to more widespread use.

The direct substitution approach method consists in solving for the individual concentrations of the components, that is *substituting* Eqs. 10–11 in Eq. 1 (this can be done explicitly, as in Hammond et al. [17], or implicitly, as in Kräutle et al. [25, 26] or Saaltink et al. [37]). It is also possible to reformulate the problem as a differential algebraic system (DAE), and to take advantage of the high-quality software available for such problems, as in Erhel et al. [10, 11], or [8]. A high-performance parallel implementation is described by Hammond et al. [17], using a Jacobian-free Newton–Krylov method (see Section 3.2).

The main advantages of this approach are to avoid the errors caused by the separation of operators, and to allow fast convergence independently of the time step, but its principal drawback is the need to form and to store the Jacobian matrix especially for a large problem. Moreover, sometimes, it may be difficult to calculate the exact derivatives for geochemical processes especially when precipitation phenomena or kinetic reactions are taken into account.

The size of the system can be made smaller by means of a reduction method, cf. Kräutle et al. [25, 26], and [18]. The reduction method makes a change of variables in the chemical system, so that a set of decoupled transport equations is first solved, leaving a smaller nonlinear system, that is still solved with Newton's method.

3.2 A Newton–Krylov-based fully coupled method

As was already mentioned in the previous section, Hammond et al. [17] have used a Newton–Krylov method for solving the system obtained from the DSA approach. Substituting the chemical equations in the transport operator is the most straightforward way of formulating the coupled problem but leads to a system where chemistry and transport terms are mixed, and makes it virtually impossible to separate the transport and chemistry modules. However, this separation is seen as one of the important advantages of the operator splitting approaches.

By coupling the formulation given in Section 2.3 with the Newton–Krylov framework, a strongly coupled method that can be implemented by keeping transport and chemistry separate is obtained. Thus, the chemical equations are not directly substituted in the transport equation, but the function ψ introduced previously in Eq. 15 is used to represent the effect of chemistry. Different formulations could be adopted depending on the choice of unknowns (refer back to Section 3). In this work, both the total mobile and fixed concentrations, and also the total concentrations (though they could easily be eliminated), are chosen as main unknowns.

Even though this method may be more expensive than the methods based on DSA, its main advantage is to make it possible to treat chemistry as a black-box, even in the Newton–Krylov context. This may be important, as chemical simulators are becoming increasingly sophisticated.

Recall (Eq. 21) that the nonlinear system to be solved at each time step is

$$\begin{cases} (M+\Delta t L)C^{n+1}_{:,j} + MF^{n+1}_{:,j} - b^{n}_{:,j} = 0, & j=1,\dots,N_c \\ T^{n+1} - C^{n+1} - F^{n+1} = 0, & \\ F^{n+1}_{i,:} - \psi\left((T^{n+1}_{i,:})^T\right)^T = 0, & i=1,\dots,N_g \end{cases}, \tag{25}$$

where $b^n_{:,j} = MC^n_{:,j} + \Delta t\, g^{n+1}_{:,j} + M\Delta t F^n_{:,j}$ is known.

Denoting by $G : \mathbf{R}^{3N_cN_g} \to \mathbf{R}^{3N_cN_g}$ the function

$$G\begin{pmatrix} C \\ T \\ F \end{pmatrix} = \begin{pmatrix} \left((M+\Delta t L)C_{:,j} + MF_{:,j} - b^n_{:,j}\right)_{j\in[1,N_c]} \\ T - C - F \\ \left(F_{i,:} - \psi\left((T_{i,:})^T\right)^T\right)_{i\in[1,N_g]} \end{pmatrix}, \tag{26}$$

the nonlinear problem to be solved at each time step is $G(Z) = 0$, where Z denotes the vector $\begin{pmatrix} C^{n+1} \\ T^{n+1} \\ F^{n+1} \end{pmatrix}$.

Recall that, at each step of the "pure" form of Newton's method for solving $G(Z) = 0$, one should compute the Jacobian matrix $J = G'(Z^k)$, solve the linear system (usually by Gaussian elimination)

$$J\,\delta Z = -G(Z^k), \tag{27}$$

and then set $Z^{k+1} = Z^k + \delta Z$. In practice, one should use some form of globalization procedure in order to ensure convergence from an arbitrary starting point. If a line search is used, the last step should be replaced by $Z^{k+1} = Z^k + \lambda\,\delta Z$, where λ is determined by the line search procedure.

The main drawback of the method for large-scale problems is again the need to form, and then factor, the Jacobian matrix. For coupled problems such as the one studied in this paper, there is the additional difficulty of simply computing the Jacobian: the numerical methods for transport and chemistry are quite different, and it is even possible that the simulation codes have been written by different groups.

The Newton–Krylov method (see [23, 24], and [17], to which our work is closely related) is a variant of Newton's method where the linear system that arises at each step of Newton's method is solved by an *iterative* method (of Krylov type). The main advantage of this type of method is that the full Jacobian is not needed; one just needs to be able to compute the product of the Jacobian with a vector. As this is a directional derivative, this leads to the Jacobian free methods, where this product is approximated by finite differences. However, for some problems, it may be possible to compute the needed directional derivative exactly. As will be seen below, this is the case for our coupled problem, provided the Jacobian of the chemical problem can be computed. This is both cheaper and more accurate.

The main contribution of this paper is to show that the formulation given above lends itself to an implementation of Newton's method that allows to keep the two codes separate. This is in keeping with thte philosophy set forth in the review paper by Knoll and Keyes [24] that a Newton–Krylov solver can often be made by wrapping a classical split-step solver. This is what is being done here, as the formulation to which the Newton–Krylov method is applied is the one used for operator splitting. Additionally, it will be shown below that the Jacobian may even be formed in block form, provided the individual codes provide their Jacobians (this is obviously easier for transport than for chemistry), and this obviously carries over to the Jacobian-vector product.

At this point, it is appropriate to add a few comments on the size of the problems envisioned. The examples used in this work are small-scale, one-dimensional problems. They can hardly be called large. On the other hand, we believe they are representative of the problems that will be encountered in more realistic applications. For such problems, in two or three space dimensions, involving tens or hundreds of thousands of grid points and several tens of chemical species, the nonlinear system will indeed be very large, and a method like that of Hammond et al. [17], or like the method presented in this section, will be necessary.

A Krylov subspace method (see, for instance, [23]) is used to approximately solve the linear system in Eq. 27. The linear iterates are drawn from the Krylov subspace, $K_j = \text{span}\{r_0, Jr_0, J^2r_0, \dots, J^{j-1}r_0\}$. In the

GMRES method (see [36]), the iterates are defined to minimize the residual $\|J\delta Z_j + G(Z)\|_2$ over K_j. Other methods, such as Bi-CGSTAB [44] or QMR [15] could be used as well.

As the linear system is not solved exactly, the convergence theory for Newton's method does not apply directly. However, the theory has been extended by Dembo et al. [7] to the class of inexact Newton methods, of which the Newton–Krylov methods are representatives. The main consequences of this analysis are summarized below.

An important issue in such methods is the stopping criterion for the inner linear iteration. A stopping criterion of the form

$$\|J\delta Z + G(Z^k)\| \leq \eta_k \|G(Z^k)\| \tag{28}$$

in this context, as the initial iterate is usually 0. The choice of the *forcing term* η_k should strike a balance between two conflicting goals:

- Keep the (local) convergence of Newton's methods.
- Avoid over-solving, which is taking too many linear iterations when still far away from the nonlinear solution.

The first goal will tend to require a small value for η_k, while the second one obviously tends to make η_k larger. It has been shown (see theorem 6.1.4 in [23]) that, provided η_k is bounded away from 1, the inexact Newton's method will converge, and that superlinear convergence obtains if η_k goes to zero faster than $\|G(Z^k)\|$. Based on this result, the strategy proposed by Kelley in [23] (after the choice in [12]) computes η_k as

$$\eta_k = \gamma \, \|G(Z_k)\|^2 / \|G(Z_{k-1})\|^2, \tag{29}$$

where $\gamma \in [0, 1]$ is a parameter (the value suggested in [23] is $\gamma = 0.9$). Safeguards are added to this choice in order to prevent η_k from becoming too close to 1, or too small. It is also necessary to globalize the algorithm, and this can be done using a line search, just as in the classical Newton's method.

The other main practical advantage of the Newton–Krylov methods is that they do not require forming the Jacobian matrix. All that is needed is the ability to compute the product of the Jacobian matrix by an arbitrary vector, in order to enlarge the Krylov subspace. This matrix–vector product can be interpreted as a directional derivative. This means that, for complex functions, G, it may not be necessary to compute the Jacobian, at the cost of one extra evaluation of the function itself. It turns out, however, that in our case, this trade-off is not advantageous. Indeed, it is well known that the most expensive part of the evaluation of G is the solution of the chemical problem at each grid point. On the other hand, it was shown above that computing the Jacobian of ψ is actually cheaper than computing ψ itself (once ψ has already been computed), as it only involves the solution of a linear system (see Eq. 17), whereas computing ψ itself requires the solution of a nonlinear system.

It will now be shown how the method can be implemented, given modules for transport and chemistry. The first ingredient needed is the computation of the residual, which is evaluating the function G defined in Eq. 26. Given a vector $Z = \begin{pmatrix} C \\ T \\ F \end{pmatrix}$, Z is first split into its three components, and each sub-vector is regarded as a $N_g \times N_c$ matrix, as in Section 2.3. Then, $G(Z)$ is computed by block:

- For the transport block, the transport operator is applied to each species $C_{:,j}$, with a source term given by $-M\dfrac{F_{:,j} - F^n_{:,j}}{\Delta t}$, for $j = 1, \ldots, N_c$ (F^n denotes F at the previous time step).
- The second block is the trivial computation $T - C - F$.
- The third block is the solution of the chemical problem at each grid point: $F_{:,i} - \psi(T_{i,:})$, for $i = 1, \ldots, Ng$.

This shows that the first block will only need transport related-quantities, whereas the third block will only call chemistry-related ones. Actually, these are the same computations that would be needed for implementing an operator-splitting method.

As far as the Jacobian matrix–vector product is concerned, and using the computation in Section 2.2, the action of the Jacobian on a vector $v = \begin{pmatrix} v_C \\ v_T \\ v_F \end{pmatrix}$ (that is the directional derivative of G in the direction of the vector v) can be computed as

$$J \begin{pmatrix} v_C \\ v_T \\ v_F \end{pmatrix} = \begin{pmatrix} \left((M + \Delta t L) v_{C:,j} + M v_{F:,j}\right)_{j \in [1,N_c]} \\ -v_C + v_T - v_F \\ \left(v_{F i,:} - v_{T i,:} (\psi'(T^T_{i,:}))^T\right)_{i \in [1,n_x]} \end{pmatrix}. \tag{30}$$

Even though it is not used as such in this work, it is valuable to examine the structure of the Jacobian. As the previous computation shows, the Jacobian also has a natural block structure. Recall that the unknowns are numbered by species at each point in space. Then, the block corresponding to the action of L can be written

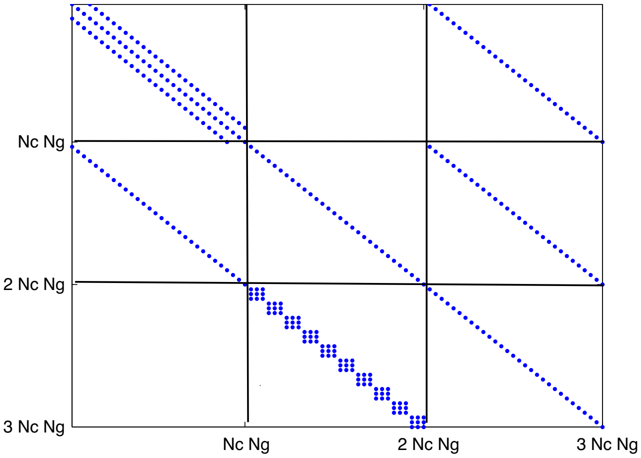


Fig. 1 The block structure of the Jacobian matrix

using the Kronecker product (see for instance [19]) as $A = (M + \Delta t L) \otimes I$. Then, the Jacobian matrix is

$$J = \begin{pmatrix} A & 0 & M \\ -I & I & -I \\ 0 & -\Psi'(T^T) & I \end{pmatrix}, \tag{31}$$

where $\Psi'(T) = \operatorname{diag}(\psi'(T^T_{1,:}), \dots, \psi'(T^T_{N_g,:})$ is the Jacobian of ψ, and for each $i = 1, \dots, N_g$, $\psi'(T^T_{i,:})$ is a small N_c-by-N_c block. The structure of the Jacobian is illustrated in Fig. 1, for the case $N_g = 10$, $N_c = 3$. It is a 3×3 block matrix, each bock being of size $N_g \times N_c$. We can clearly see the different parts of the Jacobian: the transport part in the upper left corner has three diagonals corresponding to the Kronecker product structure (remember that L is tridiagonal), and the chemistry part at the bottom has ten 3×3 blocks.

It would, in principle, be possible to compute and store the Jacobian matrix according to Eq. 31 as a sparse matrix, and to compute the matrix–vector product using a general-purpose routine. The advantage of the method given in Eq. 30 is that the structure of the Jacobian is fully exploited, which leads to a much more economical computation.

4 Numerical results

4.1 Ion exchange

The following example of advective transport in the presence of cation exchangers is adopted as a first test case comparison of both approaches. The example is used in the documentation of PHREEQC-2 [33] as Example 11. The one-dimensional simulation problem describes a column experiment where the chemical composition of the effluent from a column containing a cation exchanger is simulated. Initially, the column contains a sodium–potassium–nitrate solution in equilibrium with the cation exchanger. The column is then flushed with three pore volumes of calcium chloride solution, so that an equilibrium state with calcium and chloride is reached. Calcium, potassium, and sodium react to equilibrium with the exchanger at all times. The flow and transport parameters used for this example are presented in Table 1, and the initial and injected concentrations are listed in Table 2. The cationic exchange capacity for the exchanger is 1.1 mmol/l.

Table 1 Physical parameters

Darcy velocity	$2.78\,10^{-6}$ m/s
Diffusion coefficient	$5.56\,10^{-9}$ m^2/s
Length of column	0.08 m
Mesh size	0.0002 m
Duration of experiment	1 day
Time step	720 s

The chemical reactions for this example are:

$$\mathrm{Na^+ + X^- \rightleftharpoons NaX}$$
$$\mathrm{K^+ + X^- \rightleftharpoons KX}$$
$$\frac{1}{2}\mathrm{Ca^{2+}} + \mathrm{X^-} \rightleftharpoons \frac{1}{2}\mathrm{CaX_2},$$

with NaX, KX, and CaX_2 as (sorbed) complexes, and X indicates an exchange site with charge −1.

4.1.1 Comparison with PhreeqC

Figure 2 shows elution curves, which is the evolution of the concentration of the various species at the end of the column, as a function of time. The sorbed potassium and sodium ions are successively replaced by calcium. Because potassium exchanges more strongly than sodium (as indicated by a larger value of log K in the exchange reaction), sodium is released first, followed by potassium. Finally, when the entire concentration has been released, the concentration of calcium increases to its steady-state value, the potassium is displaced from the exchanger, and the concentration in solution increases to balance the Cl^- concentration.

Table 2 Initial and injected concentrations

Component	C_{init}	C_{inflow}
Ca	0	$0.6\,10^{-3}$
Cl	0	$1.2\,10^{-3}$
K	$2.0\,10^{-4}$	0
Na	$1.0\,10^{-3}$	0

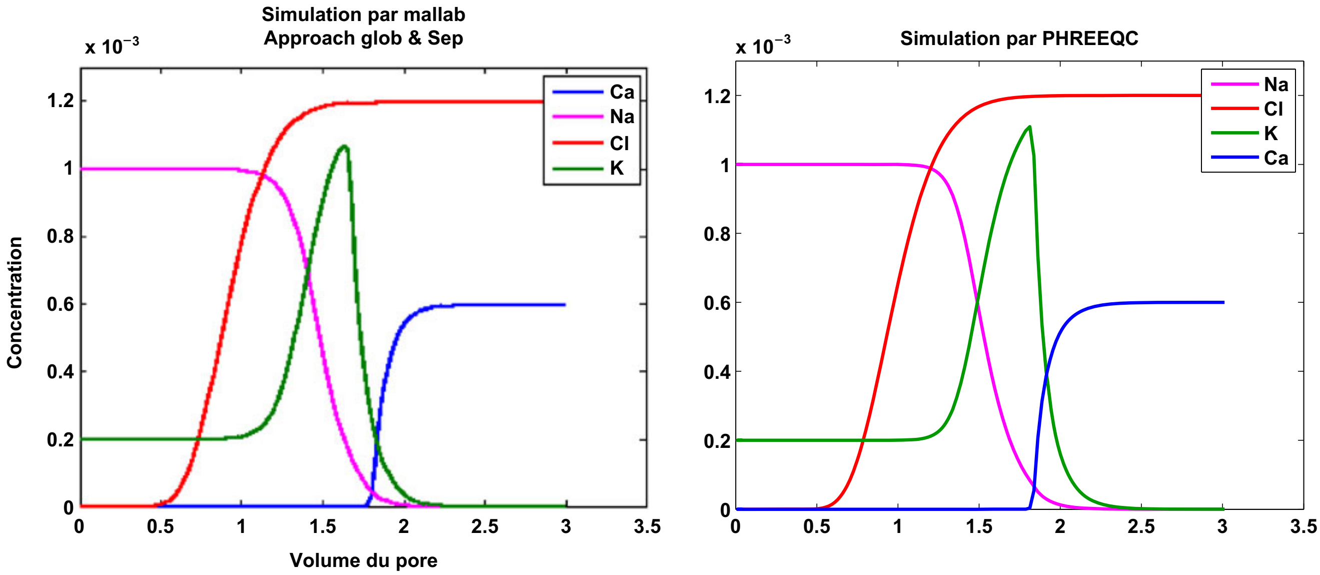


Fig. 2 Elution curves (concentrations at the end of the column) vs time, for the problem of Section 4.1. *Left*: global method, *right*: PhreeqC reference

Both the sequential method and the global method described in Section 3.2 have been applied to the test case described in Section 4.1. Both the computational demands and the accuracy of the solutions will be compared.

As can be seen in Fig. 2, the results obtained are close to those computed by PhreeqC. One can still see differences both in the location and amplitude of the peak in potassium concentration, and in the region where the three curves cross. These results are also comparable to those obtained by Xu et al. [45].

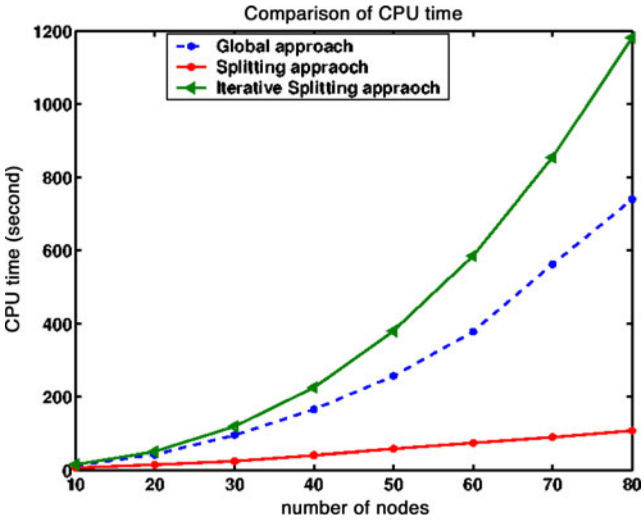


Fig. 3 Computing time for three methods applied to the ion exchange of Section 4.1

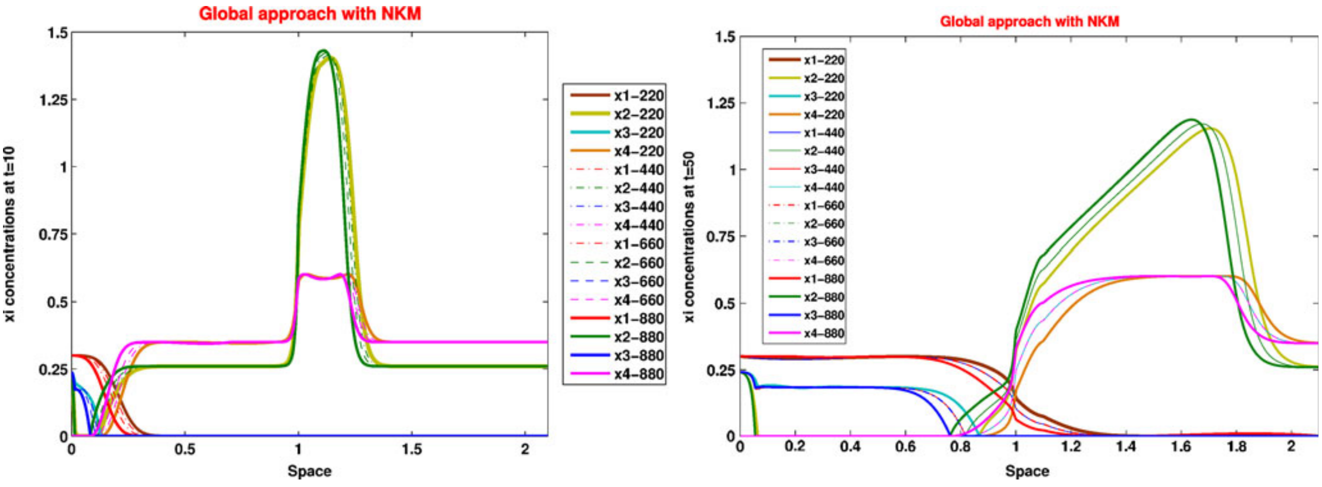
4.1.2 Performance of the method

The CPU times for the iterative splitting, non-iterative splitting, and global approaches are compared in Fig. 3. The CPU time required for each method is plotted vs the number of the nodes of the grid. As expected, it can be seen that the non-iterative method requires much less CPU time than the iterative methods. On the other hand, the global approach described in the paper requires less time than the iterative splitting, at least for the simple chemical system considered here.

For a single time step, the iterative splitting approach requires between 20 and 27 iterations on average. The number of fixed-point iterations increases with the number of nodes in the grid. On the other hand, the number of Newton iterations for the Newton–Krylov method is less than six, independent of the number of nodes. The number of Krylov iterations for each Newton step, however, does increase with the number of nodes. We go back to this issue in Subsection 4.2.1.

4.2 The 1D "easy" MoMaS benchmark

The global and the splitting approaches will now be applied to the 1D easy GDR Momas Benchmark, as described in the introductory paper to this special issue [6], see also the original description in [5]. Let us just recall that the model is a one-dimensional column, made of two different media: the part in the middle is less conductive but more reactive than the surrounding



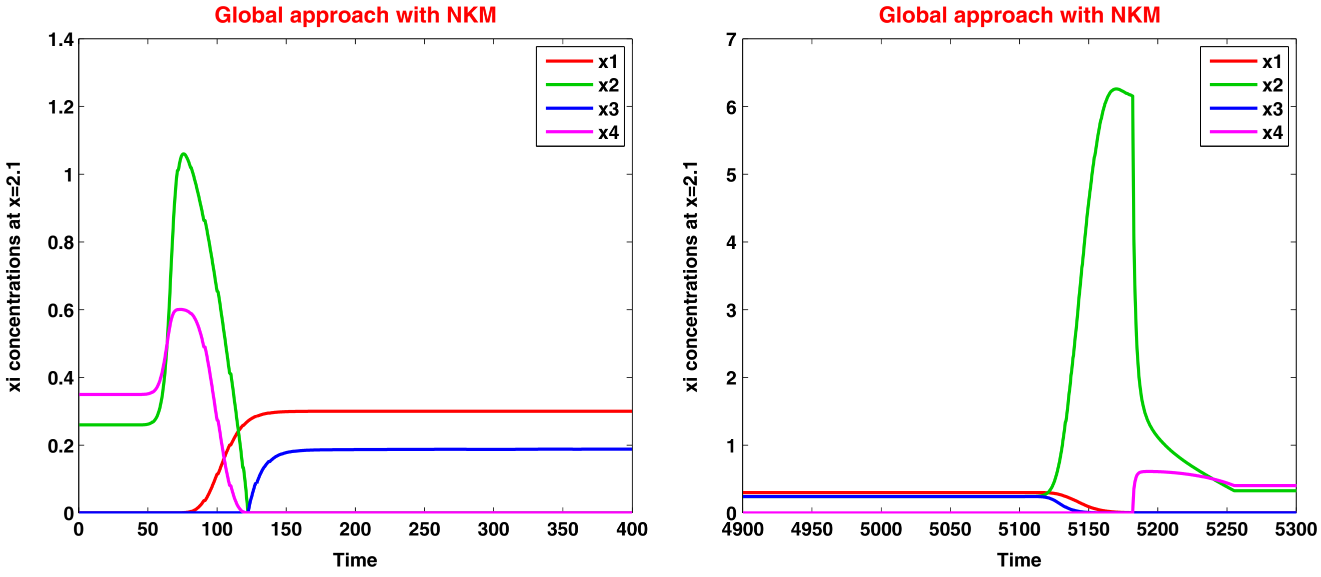
(a)

(b)

Fig. 4 Concentration of all components at times $t = 10$ (**a**) and $t = 50$ (**b**), for various mesh resolutions

medium. The chemical system has five components (four mobile components and a fixed component), and seven secondary species. The equilibrium constants vary over 50 orders of magnitude, and the stoichiometric coefficients can be as large as four, making the problem highly non-linear.

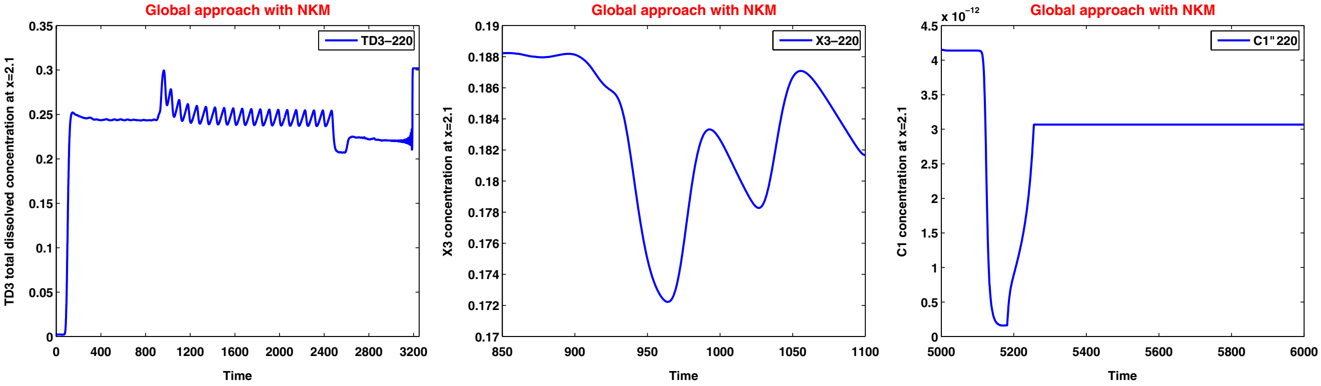
First, results showing the evolution of the component species at various times, and using several spatial and temporal resolutions, are shown in Fig. 4a. The left figure is at time $t = 10$, the right one at $t = 50$. As expected, the concentrations remain almost constant in the middle (reactive) region. Meshes with 220, 440, 660, and 880 points have been used, and in each case, the time step is chosen as 0.9 times the limit fixed by the CFL condition. For these early times, the dependence on the mesh is not very strong. Elution curves (concentrations at the end of the column as functions of time) are shown in Fig. 5, first for t going from 0 to 400 (Fig. 5a), then for t going from 4,900 to 5,300 (Fig. 5b). The elution curves show that the correct limiting behavior is reached before the leaching phase begins.



(a)

(b)

Fig. 5 Concentration of the components X1 and X4 at the end of the column ($x = 2.1$) as a function of time. **a** $t = 0$ to $t = 400$. **b** $t = 4{,}900$ to $t = 5{,}300$



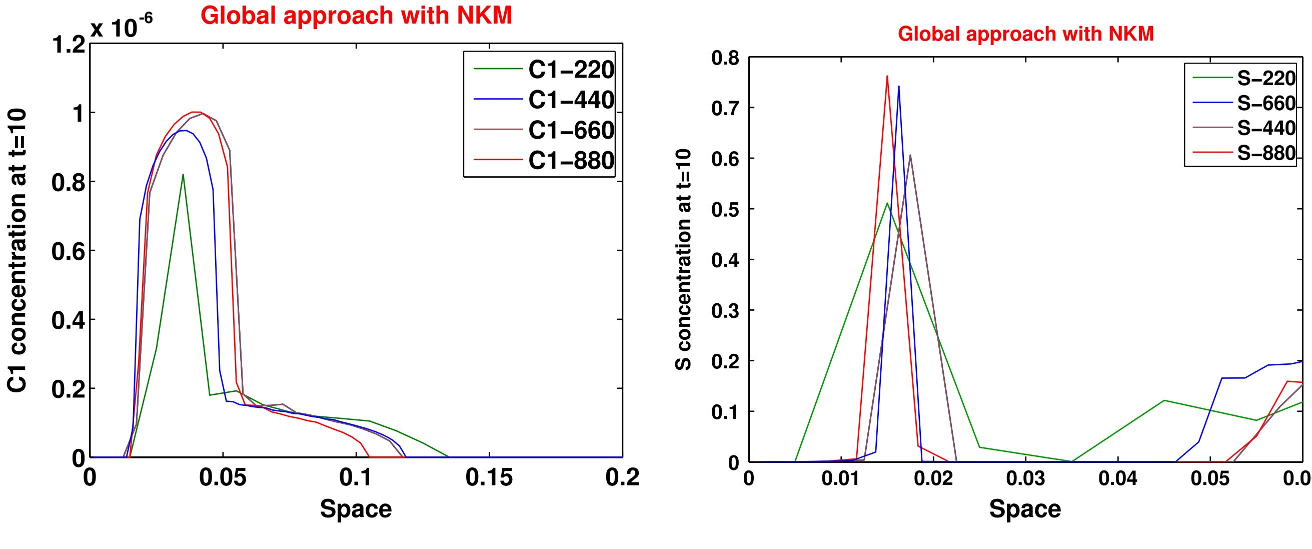
(a) (b) (c)

Fig. 6 Elution curve (concentrations at $x = 2.1$ as function of time). **a** Total dissolved concentration C3. **b** Component X3. **c** Species C1

The output results required in the benchmark definition are included. Most were obtained with a 220-point mesh, which may not be sufficient, as will be seen below. It has not yet been possible to obtain results with a finer mesh resolution for significantly longer times.

Figure 6a–c (elution curve for the total dissolved concentration of component X3 and species C1) show an oscillation pattern that has been observed by other groups working on the benchmark. These oscillations have been convincingly explained by V. Lagneau [27] as being due to the interaction of the very rapid chemistry and the discrete nature of the grid. They are a discretization artifact but appear independently of the method. They can be reduced by using a more refined grid.

Figure 7a, b shows the influence on the mesh, by showing the concentration over a small spatial region, for time $t = 10$. The concentrations are computed with four meshes of increasing resolution. The peaks in the solution are not resolved satisfactorily for the coarser mesh, with 220 points, but 660 (and better 880) points give the correct location and amplitude. Even if the method as it is currently implemented cannot yet be considered as robust, its ability to locate these solution features with reasonably coarse meshes was seen as one of its strong points. Unfortunately, this may still not be enough to eliminate the oscillations shown in Fig. 6. This issue is currently being worked on, part of the difficulty being that increasing the mesh resolution may not be sufficient. As the nonlinear problem becomes more



(a) (b)

Fig. 7 Concentration profiles. **a** Species X1, $t = 10$. **b** Component S, $t = 10$

difficult, it may be necessary to increase the maximum number of iterations allowed to make sure the Newton–Krylov method has converged.

4.2.1 Performance of the method

The benchmark was intended to be a difficult test for numerical methods, and this is indeed the case. On the average, more than 20 Newton iterations are required at each time step, and between 15 and 40 conjugate gradient steps are needed at each nonlinear iteration. Figure 8 shows a typical time step: the solid curve shows the cumulative number of conjugate gradients (alternatively, the number of matrix vector products), and the dots represent the nonlinear iterations.

Statistics for a single time step are gathered in Table 3, for three different mesh resolutions (220, 440, and 660 points). They give the number of non-linear iterations (NNI) for a (typical) time step, and the total number of linear iterations (NLI) accummulated over the whole Newton iteration. The number of nonlinear iterations depends only weakly on the mesh resolution, whereas the number of linear iterations increases with the mesh resolution.

Table 3 shows that the solver spends a large proportion of its time in the linear solver, despite the adaptive choice of the forcing parameter (Eq. 29). Moreover, the number of linear iterations for each nonlinear iteration also increases with the mesh resolution. Actually, this is expected, as the solution of the linearized problem includes the solution of the transport operator, which has an elliptic-like structure, so that its condition number grows like the square of the number of grid points. This problem could be alleviated by using a suitable preconditioner that would make the number of iterations independent of the mesh resolution (a domain decomposition preconditioner could be used as in [1]). As noticed by Hammond et al. [17], designed a matrix-free preconditioner (so as to be compatible with the Newton–Krylov framework) is a challenge. Natural choices would exploit the block structure of the Jacobian, the simpler ones being based on block-Jacobi or block Gauss–Seidel. Operator-splitting as a preconditioner has also been proposed in [17]. These possibilities are currently being explored, exploiting the block structure of the Jacobian, and the results will be reported in a forthcoming paper (Taakili and Kern, unpublished manuscript).

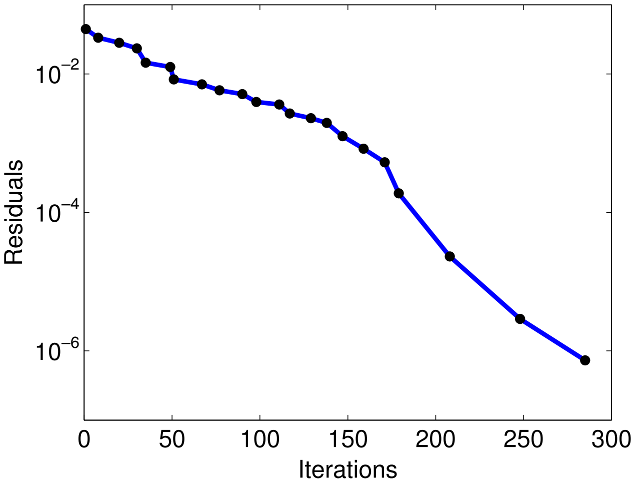


Fig. 8 Iterations

Table 3 Statistics on Newton and GNRES iterations, for one time step

Mesh 220		Mesh 440		Mesh 660	
NNI	NLI	NNI	NLI	NNI	NLI
25	494	18	551	25	636

NNI number of nonlinear iterations, *NLI* number of linear iterations

5 Conclusions—perspectives

In this paper, it was shown that a global method for coupling transport with chemistry based on the Newton–Krylov technology can be implemented while keeping the transport and chemical solvers separated. The results shown are promising: it is possible to efficiently solve geochemical problems using the method, although there remain several issues that need to be addressed.

- The first is to run test cases on more demanding configurations, where the method can be expected to show its full potential. This includes the other MoMaS test cases, with a more complex chemistry model, and also an implementation of the method in two and three dimensions.
- It will then certainly be necessary to explore the question of how to precondition the Jacobian, in order to reduce the number of Krylov iterations. An natural avenue is to reuse the operator splitting methods, as proposed by [17]. A similar study is being carried out for a related, but simpler, model.
- The results reported above used a fixed time step, which was clearly insufficient for the large interval

of integration. To successfully solve difficult problems like the benchmark above, it will clearly be necessary to use adaptive time stepping.

- A more difficult problem will be to take into account precipitation–dissolution phenomena in the chemical model. As the models are non-differentiable, this makes it more difficult to employ Newton's method.

As was apparent from the numerical experiments, the method also shows some limitations. The most serious is its high cost, as each evaluation of the residual involves the solution of a chemical problem at each grid point. The fact that the method has two levels of nonlinear iterations means that it may not be as robust as other global methods based on a single level of iterations. Finding a good preconditioner may not be a limitation, but most strategies will involve solving more transport problems, which will also incur a high cost.

Acknowledgements The first author would like to express her sincere thanks to ITASCA Consultants, France, for providing the necessary Ph.D. fellowship to carry out this research in INRIA-Rocquencourt, France. The second author's work was supported by Groupement MoMaS CNRS-2439. We gratefully acknowledge sponsorship of GDR MoMAS by ANDRA, BRGM, CEA, EDF, and IRSN. Both authors thank the referees for their detailed comments, which led to significant improvements in the contents of the paper.

References

1. Achdou, Y., Le Tallec, P., Nataf, F., Vidrascu, M.: A domain decomposition preconditioner for an advection–diffusion problem. Comput. Methods Appl. Mech. Eng. **184**(2–4), 145–170 (2000). doi:10.1016/S0045-7825(99)00227-3
2. Appelo, C.A.J., Postma, D.: Geochemistry, Groundwater and Pollution, 2nd edn. CRC, Boca Raton (2005)
3. Ascher, U.M.: Numerical Methods for Evolutionary Differential Equations. Society for Industrial & Applied Mathematics, Philadelphia (2008)
4. 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)
5. Carrayrou, J., Dimier, A., Kern, M., Knabner, P., Leterrier, N.: GDR MoMaS benchmark—reactive transport. http://www.gdrmomas.org/ex_qualifications.html (2006)
6. Carrayrou, J., Kern, M., Knabner, P.: Reactive transport benchmark of MoMaS. Comput. Geosci. (2009). doi:10.1007/s10596-009-9157-7
7. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact newton methods. SIAM J. Numer. Anal. **19**(2), 400–408 (1982). doi:10.1137/0719025. http://link.aip.org/link/?SNA/19/400/1
8. de Dieuleveult, C.: Un modèle numérique global et performant pour le couplage géochimie-transport. Thèse de doctorat, Universitè de Rennes 1 (2008)
9. de Dieuleveult, C., Erhel, J.: A numerical model for coupling chemistry and transport. In: International Conference on SCIentific Computation And Differential Equations. SciCADE (2007)
10. de Dieuleveult, C., Erhel, J.: A global approach to reactive transport: application to the MoMaS benchmark. Comput. Geosci. (2009). doi:10.1007/s10596-009-9163-9
11. de Dieuleveult, C., Erhel, J., Kern, M.: A global strategy for solving reactive transport equations. J. Comput. Phys. **228**(17), 6395–6410 (2009). doi:10.1016/j.jcp.2009.05.044
12. Eisenstat, S.C., Walker, H.F.: Choosing the forcing terms in an inexact Newton method. SIAM J. Sci. Comput. **17**(1), 16–32 (1996). citeseer.ist.psu.edu/article/eisenstat94choosing.html
13. Eymard, R., Gallouët, T., Herbin, R.: Finite volume methods. In: Ciarlet, P.G., Lions, J.L. (eds.) Handbook of Numerical Analysis, vol. VII, pp. 713–1020. North-Holland, Amsterdam (2000)
14. Fahs, M., Carrayrou, J., Younes, A., Ackerer, P.: On the efficiency of the direct substitution approach for reactive transport problems in porous media. Water Air Soil Pollut. **193**, 299–208 (2008). doi:10.1007/s11270-008-9691-2
15. Freund, R.W., Nachtigal, N.M.: QMR: a quasi-minimal residual method for non-Hermitian linear systems. Numer. Math. **60**, 315–339 (1991)
16. Frolkovič, P., Kačur, J.: Semi-analytica solution of a contaminant transport equation with nonlinear sorption in 1D. Comput. Geosci. **10**(3), 279–290 (2006). doi:10.1007/s10596-006-9023-9
17. Hammond, G.E., Valocchi, A., Lichtner, P.: Application of Jacobian-free Newton–Krylov with physics-based preconditioning to biogeochemical transport. Adv. Water Resour. **28**, 359–376 (2005)
18. 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. (2009, this issue)
19. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1990)
20. Hoteit, H., Ackerer, P., Mosé, R.: Nuclear waste disposal simulations: couplex test cases: simulation of transport around a nuclear waste disposal site: the COUPLEX test cases (editors: Alain Bourgeat and Michel Kern). Comput. Geosci. **8**, 99–124 (2004). doi:10.1023/B:COMG.0000035074.37722.71. http://www.ingentaconnect.com/content/klu/comg/2004/00000008/00000002/05379190
21. Kanney, J.F., Miller, C.T., Kelley, C.T.: Convergence of iterative split operator approaches for approximating nonlinear reactive transport problems. Adv. Water Resour. **26**(247–261) (2003)
22. Kačur, J., Malengier, B., Remešikova, M.: Solution of contaminant transport with equilibrium and non-equilibrium adsorption. Comput. Methods Appl. Mech. Eng. **194**, 479–489 (2005). doi:10.1016/j.cma.2004.05.017
23. Kelley, C.T.: Iterative methods for linear and nonlinear equations. In: Frontiers in Applied Mathematics, vol. 16. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, with separately available software (1995)
24. Knoll, D.A., Keyes, D.E.: Jacobian-free Newton-Krylov methods: a survey of approaches and applications. J. Comput. Phys. **193**(2), 357–397 (2004). doi:10.1016/j.jcp.2003.08.010
25. Kräutle, S., Knabner, P.: A new numerical reduction scheme for fully coupled multicomponent transport-reaction problems

in porous media. Water Resour. Res. **41**(W09414) (2005). doi:10.1029/2004WR003624

26. Kräutle, S., Knabner, P.: A reduction scheme for coupled multicomponent transport-reaction problems in porous media: Generalization to problems with heterogeneous equilibrium reactions. Water Resour. Res. 43(W03429) (2007). doi:10.1029/2005WR004465
27. Lagneau, V., van der Lee, J.: HYTEC results of the MoMas reactive transport benchmark. Comput. Geosci. (2009). doi:10.1007/s10596-009-9159-5
28. van der Lee, J.: CHESS, another speciation and surface complexation computer code. Tech. Rep. LHM/RD/93/39, CIG École des Mines de Paris, Fontainebleau (1993)
29. Lucille, P.L., Burnol, A., Ollar, P.: Chemtrap: a hydrogeochemical model for reactive transport in porous media. Hydrol. Process. **14**, 2261–2277 (2000)
30. Mayer, K., MacQuarrie, K.: Solution of the MoMaS reactive transport benchmark with MIN3P—model formulation and simulation results. Comput. Geosci. (2009). doi:10.1007/s10596-009-9158-6
31. Mazzia, A., Bergamaschi, L., Putti, M.: A time-splitting technique for the advection-dispersion equation in groundwater. J. Comput. Phys. **157**(1), 181–198 (2000). doi:10.1006/jcph.1999.6370
32. Morel, F.M.M., Hering, J.G.: Principles and Applications of Aquatic Chemistry. Wiley, New York (1993)
33. Parkhurst, D.L., Appelo, C.: User's guide to PHREEQC (version 2)- a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. Tech. Rep. 99-4259, USGS (1999)
34. Rubin, J.: Transport of reacting solutes in porous media: relation between mathematical nature of problem formulation and chemical nature of reactions. Water Resour. Res. **19**, 1231–1252 (1983)
35. Rudin, W.: Principles of Mathematical Analysis, 3rd edn. McGraw-Hill, New York (1976)
36. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **7**(3), 856–869 (1986)
37. Saaltink, M., Ayora, C., Carrera, J.: A mathematical formulation for reactive transport that eliminates mineral concentrations. Water Resour. Res. **34**(7), 1649–1656 (1998)
38. Saaltink, M., Carrera, J., Ayora, C.: A comparison of two approaches for reactive transport modelling. J. Geochem. Explor. **69–70**, 97–101 (2000)
39. Saaltink, M., Carrera, J., Ayora, C.: On the behavior of approaches to simulate reactive transport. J. Contam. Hydrol. **48**, 213–235 (2001)
40. Salignac, A.L.: Transport multi-espèces et réactions géochimiques en aquifère : développement et validation du modèle couplé HYTEC 2D. Ph.D. thesis, École des Mines de Paris (1998)
41. Samper, J., Xu, T., Yang, C.: A sequential partly iterative approach for multicomponent reactive transport with CORE2D. Comput. Geosci. **13**, 301–316 (2009). doi:10.1007/s10596-008-9119-5
42. Shapiro, N.Z., Shapley, L.S.: Mass action laws and the Gibbs free energy function. J. Soc. Ind. Appl. Math. **13**(2), 353–375 (1965)
43. Siegel, P., Mosé, R., Ackerer, P., Jaffré, J.: Solution of the advection-dispersion equation using a combination of discontinuous and mixed finite elements. J. Numer. Methods Fluids **24**, 595–613 (1997)
44. van der Vorst, H.A.: Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **13**(2), 631–644 (1992)
45. Xu, T., Samper, J., Ayora, C., Manzano, M., Custodio, E.: Modeling of non-isothermal multi-component reactive transport in field scale porous media flow systems. J. Hydrol. **214**, 144–164 (1999)
46. Yeh, G.T., Tripathi, V.S.: A critical evaluation of recent developments in hydrogeochemical transport models of reactive multichemical components. Water Resour. Res. **25**, 93–108 (1989)