TReacLab: An object-oriented implementation of non-intrusive splitting methods to couple independent transport and geochemical software

Daniel Jara, Jean-Raynald de Dreuzy, Benoit Cochepin

To cite this version:


HAL Id: insu-01588325
https://insu.hal.science/insu-01588325
Submitted on 15 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
TReacLab: An object-oriented implementation of non-intrusive splitting methods to couple independent transport and geochemical software

Daniel Jara, Jean-Raynald de Dreuzy, Benoit Cochepin

PII: S0098-3004(17)30251-0
DOI: 10.1016/j.cageo.2017.09.005
Reference: CAGEO 4019

To appear in: Computers and Geosciences

Received Date: 6 March 2017
Revised Date: 1 September 2017
Accepted Date: 7 September 2017


This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
TReacLab: an object-oriented implementation of non-intrusive splitting methods to couple independent transport and geochemical software

Daniel Jara¹, Jean-Raynald de Dreuzy¹, Benoît Cochepin²

¹Géosciences Rennes, UMR CNRS 6118, Campus de Beaulieu, University of Rennes 1, Rennes, France
²ANDRA, 1/7 Rue Jean Monnet, 92298 Châtenay-Malabry, France

Abstract

Reactive transport modeling contributes to understand geophysical and geochemical processes in subsurface environments. Operator splitting methods have been proposed as non-intrusive coupling techniques that optimize the use of existing chemistry and transport codes. In this spirit, we propose a coupler relying on external geochemical and transport codes with appropriate operator segmentation that enables possible developments of additional splitting methods. We provide an object-oriented implementation in TReacLab developed in the MATLAB environment in a free open source frame with an accessible repository. TReacLab contains classical coupling methods, template interfaces and calling functions for two classical transport and reactive software (PHREEQC and COMSOL). It is tested on four classical benchmarks with homogeneous and heterogeneous reactions at equilibrium or kinetically-controlled. We show that full decoupling to the implementation level has a cost in terms of accuracy compared to more integrated and optimized codes. Use of non-intrusive implementations like TReacLab are still justified for coupling independent transport and chemical software at a minimal development effort but should be systematically and carefully assessed.
Keywords: Porous media; Reactive transport; Operator splitting; Object-oriented programming.

Corresponding author: daniel.jara.heredia@gmail.com

1. Introduction

The fate of chemical species in geological media results from the interaction of physical transport and chemical reactivity (Steefel et al., 2005). Understanding how they interact requires field and laboratory studies as well as numerical models. Numerical models are important for building predictive scenarios where experiments are limited spatially and temporally, as in long-term nuclear waste disposal assessment (Marty et al., 2014; Thouvenot et al., 2013; Trotignon et al., 2007). On the physical transport side, extensive work in applied mathematics and computational science has provided widely-used software for single and multi-phase flows as well as transport of chemical species such as MODFLOW (McDonald and Harbaugh, 1988), MT3DMS (Zheng and Wang, 1999), HYDRUS (Kool and Van Genuchten, 1991), COMSOL (COMSOL, 2010), FEFLOW (Diersch, 1996), MRST (Lie, 2014), and TOUGH2 (Pruess et al., 1999). On the chemistry side, geochemical software have implemented a wide range of chemical functions and reactions, including equilibrium aqueous speciation, equilibrium mineral dissolution/precipitation, gas phase exchange, ion exchange, redox reactions, and kinetic reactions. Some of these software are PHREEQC (Parkhurst and Appelo, 1999), GEMS (Kulik et al., 2013), CHEPROO (Bea et al., 2009), MINTEQ (Peterson et al., 1987), CHESS (Van der Lee, 2002), and Geochemist's Workbench (Bethke, 2007).

To combine physical and chemical reactivity, couplers have been developed between transport and geochemical codes such as PHAST for coupling HST3D and PHREEQC (Parkhurst et al., 2004), HP1 for HYDRUS and PHREEQC (Šimůnek et al., 2006), PHT3D
for MT3DMS and PHREEQC (Prommer et al., 1999), HYTEC for RT1D/R2D2/METIS and CHESS (van der Lee et al., 2003), OpenGeoSys-GEMS (Kulik et al., 2013) and iCP for COMSOL and PHREEQC (Nardi et al., 2014), UTCHEM-IPhreeq and UTCHEM-EQBATCH (Kazemi Nia Korrani et al., 2015, 2016), multicomponent transport software-IPhreeqc (Muniruzzaman and Rolle, 2016), FEFLOW-IPhreeqc (MIKE(DHI), 2016), Lattice Boltzmann transport software-IPhreeqc (Patel et al., 2013). Most of the previously cited codes have embedded the coupling method with the geochemical and transport methods to enhance global performance and reliability. Here, in order to gain flexibility, we propose in our code TReacLab a complementary development in the form of an ensemble of Operator Splitting methods (OS) with a generic set of interfaces to transport and reaction operators. In this context, OS decouples chemistry from transport as opposed to global implicit solvers, which have been proven to be more accurate but less flexible (Hammond et al., 2012; Hammond et al., 2014; Mayer, 2000; Steefel, 2009; Zhang, 2012).

TReacLab is designed as an open toolbox where additional OS techniques can be implemented and benchmarked. Other transport and geochemical codes may also be used at the minimal cost of developing the necessary interfaces. TReacLab is written in MATLAB based on a series of abstract classes using object-oriented programming (Commend and Zimmermann, 2001; Register, 2007; Rouson et al., 2011).

After recalling in section 2 the reactive transport and OS formalism used, we present in section 3 our OS implementation. We especially show how to implement alternative OS methods and how to connect other transport and geochemical codes. Methods are assessed and discussed on the basis of 3 benchmarks in section 4.
2. Numerical model

2.1. Reactive transport equation

The reactive transport equation can be written in a general way as (Saaltink et al., 1998): 

\[ \frac{\partial \mathbf{c}}{\partial t} = M \mathbf{c} + \Theta S_r \mathbf{v}_r + \Theta S_e \mathbf{v}_e + Q, \]  

(1)

where \( \mathbf{c} \) is the vector of concentrations for \( N_c \) chemical species in the system. \( \Theta \) is a diagonal matrix containing the porosity or volumetric content of the phase. \( M \) is a diagonal matrix that specifies whether a species is mobile or immobile. Its diagonal elements are 1 or 0 accordingly. \( S_k^i \) and \( S_e^i \) are the transposed stoichiometric matrix for kinetic and equilibrium reactions, respectively. \( r_c \) and \( r_k \) (ML^{-3}T^{-1}) are the reaction rates of the \( N_e \) equilibrium and \( N_k \) kinetic reactions, respectively. \( Q \) is the external sink/source term (ML^{-3}T^{-1}). \( L \) is the transport operator (ML^{-3}T^{-1}), which includes advection and diffusion. In the following, we consider only single-phase flow:

\[ L(\mathbf{c}) = \nabla \cdot [D \nabla \mathbf{c} - \Theta \mathbf{v} \mathbf{c}], \]  

(2)

\( D \) (L^{2}T^{-1}) is the effective dispersion-diffusion tensor (Bear, 1972). The velocity \( \mathbf{v} \) (LT^{-1}) is computed in a pre-processing phase, which can be decoupled from the reactive transport problem as long as hydraulic properties are not modified by the chemical reactivity. The chemical system can be generically written as the combination of the \( N_e \) equilibrium reactions:

\[ \mathcal{Q}_e(\mathbf{c}) = 0. \]  

(3)
and of the $N_k$ kinetically-controlled reactions:

$$r_k = \varphi_k(c).$$  \hspace{1cm} (4)

The reactive transport problem is thus made up of the $N_c$ mass balance equation (1) and of the $N_c + N_k$ equilibrium and kinetic equations (3) and (4). Its unknowns are the concentrations $c$ and the reaction rates $r_e$ and $r_k$. The chemical equilibrium system (3) is composed of the conservation equation and of the mass action law, relating reactants and products (Apoung-Kamga et al., 2009; Molins et al., 2004):

$$S_e \log(c) = \log(K).$$  \hspace{1cm} (5)

where $K$ is the vector of equilibrium constants.

Components $u$ are generally introduced when considering equilibrium reactions (Saaltink et al., 2011):

$$u = Uc.$$  \hspace{1cm} (6)

where $U$ is the component matrix (Fang et al., 2003; Friedly and Rubin, 1992; Hoffmann et al., 2012; Kräutle and Knabner, 2005; Steefel et al., 2005). They are $N_e - N_c$ linear combinations of chemical species that are not modified by equilibrium reactions (Molins et al., 2004; Morel and Hering, 1993):

$$US^T \hat{r}_e = 0.$$  \hspace{1cm} (7)
The component matrix is not unique. However, its application to equation (1) always leads to a reduced system without the equilibrium rates but with the components $u$ (Molins et al., 2004; Saaltink et al., 1998):

$$\frac{\partial u}{\partial z} = UML(c) + U0S \xi_{k} r_k + UQ.$$  \hspace{1cm} (8)

The reactive transport problem is then made up of the $2N_s - N_s + N_k$ equations (3-6) and (8) for the same number of unknowns $u$, $c$ and $r_k$.

Under the assumption that solid species are not transported and all species have the same diffusion coefficient (i.e. $UML(c) = UL(u)$). Equation (8) classically gives the two following formulations TC and CC (Amir and Kern, 2010):

TC:

$$\frac{\partial u_{a}}{\partial z} = L(u_{a}) + UBS \xi_{k} r_k + UQ.$$  \hspace{1cm} (9)

CC:

$$\frac{\partial u_{a}}{\partial z} + \frac{\partial u_{f}}{\partial z} = L(u_{a}) + UBS \xi_{k} r_k + UQ.$$  \hspace{1cm} (10)

where $u_{a} = UMe$ and $u_{f} = U(I - M)e$ are the aqueous and fixed components. In the TC formulation, the fixed species concentration are deducted from the solution in the total component concentration (T) and the solute concentration (C). In the CC formulation, the total component concentration is divided in aqueous and fixed components.
2.2. Usual first-order sequential non-iterative and iterative approaches

In this section, we show how the reactive transport problem can be solved using independent transport and chemical solvers. We distinguish the sequential non-iterative and iterative approaches respectively based on TC and CC formulations. For the sequential non-iterative approach, we extract from the TC formulation, the transport operator in which we keep the sink/source term:

\[ \frac{\partial u}{\partial t} = L(u) + UQ. \]  

(11)

The chemical operator derives from equations (3-6), and (8). Note that it does not contain any source/sink term, as it has been included in the transport equation:

\[ \frac{\partial u}{\partial t} = U S r_k \]

\[ r_k = \varphi_k(c) \]

(12)

This is still a system of \(2N_s - N_c + N_k\) equations for the same number of unknowns. This decoupled system can be solved with the classical sequential non-iterative approach using an explicit integration of temporal derivatives (herein, we assume forward Euler). The solution at time step \(n+1\) can be obtained from the solution at time step \(n\), with the following successive application of the transport and chemical operators in a sequential approach:
\[ u^n = u_n + \Delta t (L(u_n) + UQ) \]

\[
\begin{cases}
  u^n = Uc_{n+1} \\
  \Phi_{u}(c_{n+1}) = 0
\end{cases}
\]

\[ r_{k_{n+1}} = \Phi_{c}(c_{n+1}) \]

\[ u_{n+1} = u^n + \Delta t USE_k r_{k_{n+1}} \]

The transport operator (11) is applied to the components. Then the chemical operator is applied with the updated mobile components for speciation between fixed and solute concentrations. In the specific case where chemical reactions are all at equilibrium and no kinetics is involved, a TC formulation is used to fully decouple (de Dieuleveult et al., 2009).

In such case the decoupling does not then rely on operator splitting, but on a block Gauss-Seidel method. When the stability conditions of the explicit integration are too much constraining, implicit schemes should be used instead within a sequential iterative approach (Carrayrou et al., 2004; de Dieuleveult and Erhel, 2010; Yeh and Tripathi, 1989):

\[ u_{n+1} = u_n + \Delta t [L(u_{n+1}) + \Theta_{SE_k} r_{k_{n+1}} + UQ] \]

\[
\begin{cases}
  u_{n+1} = Uc_{n+1} \\
  \Phi_{u}(c_{n+1}) = 0
\end{cases}
\]

\[ r_{k_{n+1}} = \Phi_{c}(c_{n+1}) \]

Classical Picard's method have been extensively used to solve such kind of problems:
\[ u_{n+1}^{k+1} = u_n + \Delta t [L(u_{n+1}^{k+1}) + \theta S_{n+1}^k + UQ] \]

\[
\begin{align*}
\{ u_{n+1}^{k+1} &= Uc_{n+1}^{k+1} \\
\phi_c(u_{n+1}^{k+1}) &= 0
\end{align*}
\]

(15)

where \( k \) is the index of the Picard iteration method instantiated by:

\[ u_{n+1}^{k+1} = u_n \]

(16)

We recall the necessity to check the consistency of the temporal integration scheme with the Operator Splitting method chosen. With this decomposition, explicit first-order scheme naturally leads to sequential non-iterative approach. The implicit first-order scheme requires a sequential iterative approach. Other choices are possible and might reduce errors depending on the chemical system (Barry et al., 1996). As it should be possible to test and benchmark them at a reduced development cost, we use a generic decoupling formalism that can be used to implement a broad range of schemes.

2.3. Generic operator splitting implementation

The reactive transport system can be generically split in two operators. Using the formalism of Gasda et al. (2011), equation (1) can be written as:
where $z$ is the unknown, $L_1$ and $L_2$ can be equation (11) and (12), respectively. Other decomposition are possible, e.g. the transport operator can be subdivided into an advection and a diffusion-dispersion operator (Clement et al., 1998), or one operator might contain advection-reaction and the other diffusion (Liu and Ewing, 2005). Each operator will be solved separately for a splitting time step $\Delta t^i = t^{i+1} - t^i$ using adapted numerical methods.

The generic operator splitting methods implemented into the Toolbox are the sequential splitting, additive splitting, Strang splitting, symmetrically weighted splitting, and alternating method (Appendix A). Assuming exact integration of the operators and homogeneous boundary conditions in equation (18), the first two have a first-order temporal truncation error, and the following three a second-order one (Hundsdorfer and Verwer, 2013). Since the operators are usually solved using numerical methods, the global order of such approaches might be modified because of the order of the numerical methods used for each operator (Barry et al., 1996; Csomós and Faragó, 2008). The alternating splitting increases the order of the sequential splitting if the time steps are small enough (Simpson and Landman, 2008; Valocchi and Malmstead, 1992).

3. Operator splitting implementation and software organization

We provide in TReacLab an object-oriented toolbox for the non-intrusive operator splitting methods of the previous section. TReacLab is organized along three main components for coupling transport and reactivity, and proceeds in three pre-processing, processing and post-processing phases (Figure 1). These three components correspond to the three
well-identified coupler, transport and chemistry classes. The three classes are fully segmented and exchange information through interfaces. Segmentation ensures that any of the three coupler, transport and chemistry classes can be replaced without modifications of any of the two other ones. The solution of the reactive transport problem after spatial discretization eventually consists in the temporal integration with the chosen OS technique, which iteratively calls transport and geochemical solvers through interfaces (Figure 1, middle row). This is the core of the simulation that we identify as the processing phase. It is generic and does not require at run time any further specification of transport, reactivity and coupler methods. Standard error management techniques are used to stop the algorithm when any of the integration method of the three classes fails, stopping the running process and returning adapted error messages.
**Figure 1:** General software organization of TRacLab with the three coupler, transport and chemistry classes in columns, and the three pre-processing, processing, and post-processing phases in rows. Generic components represented in black are the organization and the coupler class. External software for transport and chemistry are represented in blue with hatched line (cannot be modified). Red boxes highlight the instantiation and interface methods that must be developed when connecting new transport or chemistry software.

The processing phase can be generic because all specifications of the coupler, transport, and chemistry classes are performed in a pre-processing phase (Figure 1, first row). The pre-processing phase consists in the instantiation of the coupler, transport and chemical classes, in the preparation of the interfaces that will transfer information and in the specifications of the initial conditions. As detailed in Appendix B, instantiations are code dependent. Instantiation can be done externally for example with the definition of a transport
or chemical problem through the graphical user interface of software like COMSOL or PHREEQC. It can also be done internally by a method within TReacLab specifying the inputs and parameters to existing interfaces like IPhreeqc (Charlton and Parkhurst, 2011), PhreeqcRM (Parkhurst and Wissmeier, 2015), or COMSOL livelink (COMSOL, 2010). Even when instantiation is complex, it remains independent for each of the three classes. Cross-dependencies and feedback between transport and reactivity like density-driven flows with reacting species are not supported at this stage, although they may be important in some applications like CO$_2$ sequestration (Abarca et al., 2013).

Pre-processing phase specifies the initial conditions and transfers them to the coupler in charge of starting the numerical integration. Post-processing is generic and only consists in formatting and storing output concentrations and solver performances (Figure 1, bottom row). Specifications are all restricted to the instantiation of the software and interface in the pre-processing phase while processing and post-processing remain fully generic. Connections between specific algorithms and generic structures are done by interfaces. Appendix B provides a detailed description of the transport and chemistry classes, defining the interfaces to the external codes.

4. Examples and benchmarks

The three following examples validate the methods and illustrate the implementation presented in sections 2 and 3. The three of them are based on a 1D hydraulically homogeneous system with steady-state flow and uniform dispersion (equation (2)). The examples are compared visually against analytical solution or well-know numerical software. Moreover, we show a convergence study for the first case being the reference solution the numerical solution with finest time resolution.
The four examples display evolving degrees of complexity both in terms of chemical systems and in terms of software called for transport and reactivity, software versions are given in Table 1. The first example is a single-species transport with first-order decay. The transport solver is COMSOL and the chemical solver is a simple analytical solution. This example is used to assess the different coupling algorithms implemented and to check the implementation of the interface with COMSOL. The second example is an equilibrium precipitation/dissolution chemical system in a 1D hydraulically homogeneous system. Chemical solver is IPhreeqc. Several solvers have been compared for the transport solver, both to check IPhreeqc interface implementation and to evaluate the effect of the transport solver. The third example is the most advanced in terms of chemistry. Chemical reactions are partly in equilibrium and partly kinetically controlled. They involve precipitation and dissolution reactions. The chemical code is PhreeqcRM. It is used in combination with COMSOL as transport solver. The last problem face a 2D unsaturated system where transport is modeled by Richards equation and solved by COMSOL. Chemistry is solved by PhreeqcRM. These four test cases have been chosen to check the implementation and assess the coupling methods developed. They are also simple enough from the development point of view to be taken as starting points to model more advanced chemical systems and transport conditions.
Table 1: Software versions.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB</td>
<td>R2013b</td>
</tr>
<tr>
<td>COMSOL</td>
<td>4.3b</td>
</tr>
<tr>
<td>PHREEQC</td>
<td>3.3.7</td>
</tr>
<tr>
<td>IPhreeqc</td>
<td>3.3.7</td>
</tr>
<tr>
<td>PhreeqcRM</td>
<td>3.3.9</td>
</tr>
</tbody>
</table>

4.1. Single-species transport with first-order decay

A single-species transport with first-order decay using different OS methods is compared to an analytical solution (Van Genuchten and Alves, 1982). The reactive transport system contains a single solute species of concentration $c$:

$$\frac{\partial c}{\partial t} = L(c) - kc. \quad (18)$$

where $L$ is given by equation (2). Equation (18) can straightforwardly be separated into transport and chemistry operators corresponding to the two right-hand side terms.

At time 0, the solute concentration is 0 in the domain ($c(x, t=0) = 0$). The concentration at the left boundary is constant and equal to 1 mol/m$^3$ ($c(x = 0, t) = 1$ mol/m$^3$). The boundary condition on the right side of the domain is a perfectly absorbing condition ($c(x = x_{\text{max}}, t) = 0$).

Parameters are derived from Steefel and MacQuarrie (1996) and given in Table 2. The solver for transport is COMSOL and an analytical solution is used for the first-order decay. Solute concentration progressively invades the domain from the left boundary with a smooth profile resulting from the combination of dispersion and decay (Figure 2). Second-order methods perform much better than first-order methods as expected. Errors are more pronounced at the inlet boundary condition on the left side of the domain where the concentration is higher.
(Steefel and MacQuarrie, 1996; Valocchi and Malmstead, 1992). The sequential splitting method with the transport operator performed first overestimates the amount of reaction for the whole domain since it considers that all incoming solute is getting in without decay for the full first time step. If the sequence of operators is exchanged, namely first chemistry is solved, and then transport is solved, the amount of reaction is underestimated. The second-order alternating splitting, which alternates between transport-chemistry and chemistry-transport steps, shows strong improvement with compensations between overestimation in the first application of the chemical operator and underestimation in the second application of the chemical operator (Simpson and Landman, 2008; Valocchi and Malmstead, 1992).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu$ [m/y]</td>
<td>100</td>
</tr>
<tr>
<td>$D$ [m$^2$/y]</td>
<td>20</td>
</tr>
<tr>
<td>$k$ [y$^{-1}$]</td>
<td>100</td>
</tr>
<tr>
<td>$x_{\text{max}}$ [m]</td>
<td>6</td>
</tr>
<tr>
<td>$\Delta x$ [m]</td>
<td>0.4</td>
</tr>
<tr>
<td>$\Delta t$ [y]</td>
<td>$4 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 2: Parameters for the single-species transport with first-order decay benchmark. $\nu$ is the velocity, $D$ is the dispersion coefficient, $k$ is the decay rate, $x_{\text{max}}$ is the length of the 1D column, $\Delta x$ is the grid size, and $\Delta t$ is the time step.
Figure 2: Comparison of first- and second-order OS for the single-species transport with first-order decay at $t = 0.5$ y. Parameters are given in Table 2. Analytical solution is derived from Van Genuchten and Alves (1982).

The error at time $t = 0.5$ y is taken as the quadratic relative difference over the domain of the finest time step of the numerical solution and the numerical solutions for the corresponding time step, $c_{NF}$ and $c_N$ respectively:

$$
\|e\|_2 = \sqrt{\sum_{i=1}^{N_{F}} \left( \frac{c_{NF}(i) - c_N(i)}{c_{NF}(i)} \right)^2}.
$$

Table 3 displays the values for evolving time steps and shows that all methods converge with the time. The reference finest time step for each method has been $\Delta t = 2 \times 10^{-4}$ s (i.e. $c_{NF}(0)$ value). While all methods perform well, the sequential method is more accurate than the additive one and second-order methods are overall more accurate than first-order methods. The performance on convergence arranged on descending order is given by Strang, symmetrically weighted splitting, alternating, sequential and additive.
<table>
<thead>
<tr>
<th>$\Delta x = 0.4 \text{ m}$</th>
<th>$\Delta t (y)$</th>
<th>$1^{\text{st}}$ order</th>
<th>$2^{\text{nd}}$ order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Additive</td>
<td>Sequential</td>
<td>Alternating OS</td>
</tr>
<tr>
<td>$4 \times 10^{-3}$</td>
<td>1.107</td>
<td>0.1667</td>
<td>0.075</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.032</td>
</tr>
<tr>
<td>$2 \times 10^{-3}$</td>
<td>0.514</td>
<td>0.079</td>
<td>0.032</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.026</td>
</tr>
<tr>
<td>$4 \times 10^{-4}$</td>
<td>0.114</td>
<td>0.019</td>
<td>0.029</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.031</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.029</td>
</tr>
</tbody>
</table>

Table 3: Error $\|\varepsilon\|_2$ of equation (19) for the single-species transport with first-order decay with different OS methods and splitting time steps.

4.2. Calcite dissolution

Calcite dissolution and dolomite formation has become a classical benchmark for reactive transport problems with sharp precipitation/dissolution fronts (Beyer et al., 2012; Engesgaard and Kipp, 1992; Prommer et al., 1999). Progressive introduction of magnesium calcium in a domain at equilibrium between calcium carbonate in solution and calcite (CaCO$_3$) dissolves the calcite and precipitates dolomite (CaMg(CO$_3$)$_2$). This chemical system has been modeled with the physical and chemical parameters given by Table 4, Table 5, and Table 6. Chemical concentrations are initially homogeneous. At the initial time ($t = 0$), the chemical system is destabilized with the introduction of magnesium instead of calcium at the upper boundary condition ($x = 0$), inducing the dissolution/precipitation process. The boundary condition at the downstream limit ($x_{\text{max}}$) is a simple outflow of the solutes.

Here, we show how transport solvers can be applied and validate our interface to IPhreeqc. IPhreeqc performs the computation of components, aqueous speciation, precipitation and dissolution reactions (Charlton and Parkhurst, 2011). The database used is 'NAPSI_290502(260802).dat'. Transport is solved either with COMSOL Multiphysics (COMSOL, 2012), with a finite difference spatial discretization and forward Euler time
integration, derived from built-in pdepe function of MATLAB (Skeel and Berzins, 1990). Transport and chemistry are coupled through the simple sequential approach of equations (A.1)-(A.3). PHREEQC is independently run as 1D reactive transport solver for general comparison.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nu$ [m/s]</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>$D$ [m$^2$/s]</td>
<td>$6.7 \times 10^{-8}$</td>
</tr>
<tr>
<td>$x_{\text{max}}$ [m]</td>
<td>0.25</td>
</tr>
<tr>
<td>$\Delta x$ [m]</td>
<td>0.01</td>
</tr>
<tr>
<td>$\Delta t$ [s]</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 4: Physical parameters for the calcite dissolution benchmark. $\nu$ is the average velocity, $D$ is the dispersion coefficient, $x_{\text{max}}$ is the maximum length of the column, $\Delta x$ is the grid size, and $\Delta t$ is the time step.

<table>
<thead>
<tr>
<th>Chemical Component and Species</th>
<th>Initial value</th>
<th>Boundary value at $x=0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ca [mol/L]</td>
<td>$1.23 \times 10^{-4}$</td>
<td>0</td>
</tr>
<tr>
<td>C [mol/L]</td>
<td>$1.23 \times 10^{-4}$</td>
<td>0</td>
</tr>
<tr>
<td>Cl [mol/L]</td>
<td>0</td>
<td>$2 \times 10^{-3}$</td>
</tr>
<tr>
<td>Mg [mol/L]</td>
<td>0</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>pH [-]</td>
<td>9.91</td>
<td>7</td>
</tr>
<tr>
<td>Calcite [mol/L]</td>
<td>$2 \times 10^{-4}$</td>
<td>-</td>
</tr>
<tr>
<td>Dolomite [mol/L]</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5: Calcite dissolution benchmark initial and boundary values for aqueous components and mineral species. In PHREEQC, components are called elements.
### Homogeneous reactions

<table>
<thead>
<tr>
<th>Reaction</th>
<th>log (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2H^+ + 2e^- \leftrightarrow H_2$</td>
<td>-3.1055</td>
</tr>
<tr>
<td>$2H_2O - 4H^+ - 4e^- \leftrightarrow O_2$</td>
<td>-85.9862</td>
</tr>
<tr>
<td>$HCO_3^- + 9H^+ + 8e^- - 3H_2O \leftrightarrow CH_4$</td>
<td>27.8493</td>
</tr>
<tr>
<td>$H_2O - H^+ \leftrightarrow OH^-$</td>
<td>-13.9995</td>
</tr>
<tr>
<td>$H^+ - H_2O + HCO_3^- \leftrightarrow CO_2$</td>
<td>6.3519</td>
</tr>
<tr>
<td>$HCO_3^- - H^+ \leftrightarrow CO_3^{2-}$</td>
<td>-10.3289</td>
</tr>
<tr>
<td>$Ca^{2+} - H^+ + HCO_3^- \leftrightarrow CaCO_3$</td>
<td>-7.1048</td>
</tr>
<tr>
<td>$Ca^{2+} + HCO_3^- \leftrightarrow CaHCO_3^+$</td>
<td>1.1057</td>
</tr>
<tr>
<td>$Ca^{2+} + H_2O - H^+ \leftrightarrow CaOH^+$</td>
<td>-12.78</td>
</tr>
<tr>
<td>$Mg^{2+} - H^+ + HCO_3^- \leftrightarrow MgCO_3$</td>
<td>-7.3492</td>
</tr>
<tr>
<td>$Mg^{2+} + HCO_3^- \leftrightarrow MgHCO_3^+$</td>
<td>1.0682</td>
</tr>
<tr>
<td>$Mg^{2+} + H_2O - H^+ \leftrightarrow MgOH^+$</td>
<td>-11.44</td>
</tr>
</tbody>
</table>

### Homogeneous reactions

| Calcite                                                                 | 1.849    |
| $CaCO_3 \leftrightarrow Ca^{2+} - H^+ + HCO_3^-$                         |          |

| Dolomite                                                                | 4.118    |
| $CaMg(CO_3)_2 \leftrightarrow Ca^{2+} + Mg^{2+} - 2H^+ + 2HCO_3^-$     |          |
Table 6: Chemical system of the calcite dissolution benchmark. The upper part comprises the homogeneous equations and the lower part the heterogeneous reactions. The first column shows the equilibrium reactions and the second one the logarithms of equilibrium constants.

Figure 3 and Figure 4 display aqueous and mineral equivalent concentrations at time $t = 10^4$ s. As magnesium and chloride get in the domain (Figure 3b and Figure 3d), calcite progressively dissolves and is replaced by dolomite as expected (Figure 4). Some of the calcium remains in solution and is flushed out (Figure 3a and Figure 3c). Because of the subsequent absence of calcium in solution, dolomite dissolves again with some increase of calcium in solution (Figure 3a and Figure 3c). The three different transport solvers give the same tendency as the PHREEQC solution.
Figure 3: Aqueous concentration profiles at time $t = 10^4$ s.

Figure 4: Dolomite and calcite equivalent concentration profiles with open and filled symbols respectively at time $t = 10^4$ s.

Although COMSOL leads to good results, it is more than one order of magnitude slower than the two other transport methods (Table 7). We checked that this large difference in performances does not come from the numerical method but from the large time required for COMSOL to start and stop when called numerous times externally. While this might not be
an issue for large transport problems for which limitations will rather come from transport
operator, it is a constrain for smaller tests and benchmarks.

<table>
<thead>
<tr>
<th>Software Coupling</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPhreeqc + COMSOL</td>
<td>668 s</td>
</tr>
<tr>
<td>IPhreeqc + FD script</td>
<td>24 s</td>
</tr>
<tr>
<td>IPhreeqc + pdepe</td>
<td>40 s</td>
</tr>
</tbody>
</table>

*Table 7: Time performance for the calcite dissolution benchmark using a sequential operator splitting.*

Whatever the coupling method, the consistency with PHREEQC is overall good. Although
COMSOL uses, as default, implicit time integration schemes for solving the transport
equation instead of the required explicit method, it still compares well with PHREEQC and
the other software couplings. Indeed, the sequential non iterative method requires an explicit
time integration for transport (equation (13)). It is not the case for COMSOL which uses (as
default) a backward differentiation formula temporal integration scheme, which order varies
with the internal time step adaptation (COMSOL, 2012). It thus introduces an additional error
in the coupling scheme (de Dieuleveult *et al.*, 2009). However, by using such stable and
accurate temporal integrations, it enhances the robustness of the transport scheme.

**4.3. Mixed equilibrium-kinetic system**

We simulate the progressive increase of dissolved species in an atmospheric water infiltrating
a granitic bedrock. This test case is derived from Nardi *et al.* (2014). The hydraulic properties
of the system are found in Table 8. The infiltrating water has much lower concentrations of
dissolved species than the resident water. It interacts with five minerals (Table 9). It is in
equilibrium with calcite. The four other minerals k-feldspar, illite, albite and pyrite are subject
to kinetically controlled dissolution with rates ranging from $10^{-13}$ to $10^{-11}$ mol/s. All
parameters and rate laws of the simulation are provided in the PHREEQC file of iCP (Nardi et al., 2014). The infiltrating water dissolves calcite to maintain equilibrium, increasing both the concentration of calcium and the pH of the solution. Other minerals also dissolve and increase the concentrations of Al and K in solution, however at a much slower rate because of the kinetic control of the reactions. pH is eventually buffered by the dissolution of illite and pyrite.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nu ) [m/s]</td>
<td>( 2.78 \times 10^{-6} ) m/s</td>
</tr>
<tr>
<td>( D ) [m/s]</td>
<td>( 5.55 \times 10^{-9} ) m²/s</td>
</tr>
<tr>
<td>( x_{\text{max}} ) [m]</td>
<td>0.08 m</td>
</tr>
<tr>
<td>( \Delta x ) [m]</td>
<td>( 10^{-3} ) m</td>
</tr>
<tr>
<td>( \Delta t ) [s]</td>
<td>720 s</td>
</tr>
</tbody>
</table>

Table 8: Parameters for mixed equilibrium-kinetic benchmark. \( \nu \) is the average velocity, \( D \) is the dispersion coefficient, \( x_{\text{max}} \) is the maximum length of the column, \( \Delta x \) is the grid size, and \( \Delta t \) is the splitting time.
Table 9: Aqueous components and mineral species for mixed equilibrium-kinetic benchmark.

To simulate this set of reactions, we have chosen PhreeqcRM to assess the flexibility of TReaCLab. Transport is simulated with COMSOL to benefit from the accurate transport solver, it uses a variable order (between 1 and 5) backward differentiation formula. In the
presence of both kinetically controlled and equilibrium reactions, both the quality of the transport and reactive integrations and coupling issues may be critical. We choose a simple sequential OS method with the successive integration of transport and reactivity. The results obtained by the coupling of COMSOL and PhreeqcRM are close to the solution given by PHREEQC alone for the dissolved species and kinetically dissolving minerals (Figure 5). The time step of the coupled PhreeqcRM and COMSOL integration has been taken smaller than the characteristic mesh scale transport time and reactive time at least for the kinetical reaction to ensure accurate integrations. The most difficult quantity to get accurately is the calcium concentration because calcite is at equilibrium. The time step must be reduced to recover a steeper reactive front (Figure 6).

This more advanced test shows that the computational load should be well balanced between the coupler, transport and chemistry methods. While coupling is the critical component in cases of equilibrium reactions and may even require highly integrated coupling strategies like global implicit methods (Hoffmann et al., 2010; Saaltink et al., 2001), it is not the case for kinetically controlled reactions. In this case of mixed equilibrium kinetic reaction, elementary coupling and accurate transport and reactive solvers can be efficient with small enough time steps where sharp reaction fronts are involved.
Figure 5: Comparison of results between the coupling of PhreeqcRM and COMSOL and PHREEQC observed for the mixed equilibrium-kinetic benchmark at the output of the column.
Figure 6: Quantity of dissolved calcite with PhreeqcRM and COMSOL for two different splitting time steps $\Delta t = 720 \, s$, $360 \, s$ and $90 \, s$. PHREEQC independently is used as reference.

4.4. Pesticide infiltration

The following benchmark concerns the infiltration in an unsaturated soil column of a carbamate insecticide (Aldicarb) (MIKE(DHI), 2016; Multiphysics, 2008; Šimůnek et al., 1994; Wissmeier and Barry, 2011). The soil column is a 2D axisymmetric cylinder made up of two layers with a smaller hydraulic conductivity in the upper layer but higher saturation. Transport is modeled by Richards' equation and solved by COMSOL (Figure 7). Aldicarb is transported downwards and sideways from the infiltration (top of the column from $r = 0 \, m$ to $r = 0.25 \, m$). Chemistry is described by first-order decay chain reactions (Figure 8), being only mobile Aldicarb, Aldicarb sulfoxide and Aldicarb sulfone (i.e. the other species are fix species). These system of ordinary differential equations is solved by PhreeqcRM.
Figure 7: Soil column geometry and mesh.

\[
\begin{align*}
\text{Aldicarb} & \quad K_1 = 0.2 \text{ d}^{-1} \quad \rightarrow \quad \text{Aldicarb oxime} \\
\text{Aldicarb sulfoxide} & \quad K_2 = 0.36 \text{ d}^{-1} \\
\text{Aldicarb sulfoxide} & \quad K_3 = 0.01 \text{ d}^{-1} \quad \rightarrow \quad \text{Aldicarb sulfoxide oxime} \\
\text{Aldicarb sulfone} & \quad K_4 = 0.024 \text{ d}^{-1} \\
\text{Aldicarb sulfone} & \quad K_5 = 0.052 \text{ d}^{-1} \quad \rightarrow \quad \text{Aldicarb sulfone oxime}
\end{align*}
\]

Figure 8: Aldicarb reaction chain.
The simulation time is 8 days with a splitting time step of 0.05 days. The number of nodes is 3936 nodes. Figure 9a and Figure 9b show the concentration in the soil column of Aldicarb and Aldicarb sulfone, respectively. Aldicarb disappears fast from the domain since its kinetic constant are fast in comparison to the kinetic constants of the daughter species. Therefore, Aldicarb (and also Aldicarb oxime) are presented close to the infiltration condition. On the contrary, the other daughter species (Aldicarb sulfoxide, aldicarb sulfone, aldicarb sulfoxide oxime, aldicarb sulfone oxime) have a similar distribution in the domain. Figure 9c and 9d show the concentration of Aldicarb and Aldicarb sulfone when \( r = 0 \) m for the different OS methods and COMSOL alone. It is possible to see a good agreement between all the methods, although a discrepancy between the methods and COMSOL is observable. The discrepancy is related to the OS error and the chosen integration time scheme for the chemistry step.
Figure 9: a) Aldicarb contour plot after 8 days, b) Aldicarb oxime contour plot after 8 days, c) Concentration aldicarb at r = 0 m for all the methods and Comsol, d) Concentration aldicarb oxime at r = 0 m for all the methods and Comsol.

5. Discussion

As shown by many previous studies and by the four examples of the previous section, reactive transport problems can be solved by a wide diversity of transport, chemistry, and operator splitting methods. No method is currently accepted as systematically more accurate and efficient than any other. Integration of the transport and chemistry operators in PHREEQC using more appropriate splitting with advection-reaction on one side and diffusion-reaction on
the other side leads to better resolution of chemical fronts as shown in the second and third cases (Parkhurst and Appelo, 1999), but such front can be obtained by smaller time steps. It is not only the integration but also the successive improvements of the methods that lead to significantly more accurate schemes. TReacLab results remain however close, displaying the same overall behavior both on solute and mineral concentrations. The interest of fully segmented reactive transport implementations like in TReacLab is not motivated by the accuracy and should not be used when other more integrated and optimized software are appropriate and freely available.

Despite their lower accuracy, fully segmented implementations may be useful in situations where flexibility is essential. It is the case when extensive modeling work has been performed in independent software environments for transport or chemistry, and extensions to reactive transport problems are required. Transport and chemistry solvers are then imposed and should be coupled with as few specific developments as possible. For example, COMSOL and PHREEQC have been interfaced here and in several other works because of their complementarity (Nardi et al., 2014; Nasir et al., 2014; Wissmeier and Barry, 2011). It is possible to specify advanced geometrical configurations in COMSOL through a convenient graphical user interface (Azad et al., 2016). PHREEQC provides advanced capacities for modeling complex geochemical systems with extensive database of reactions (Charlton and Parkhurst, 2011; Parkhurst and Wissmeier, 2015). In such cases, building the structure of the model may be the first and dominant issue in developing simulation capacities. That is when codes like TReacLab can provide practical bridges for reactive transport systems. The examples of section 4 however shows that they must be used with great care. Especially, the called software may have different temporal integration schemes than the explicit and implicit methods required by the SNIA and SIA coupling methods as discussed for the higher-order schemes of COMSOL in section 4. Using codes like COMSOL may enhance robustness at a
certain cost of accuracy. Thus, implementation capacity does not guarantee validity. Validity must be carefully checked and argued with other comparable cases or with appropriate convergence analysis.

Another targeted use of TReacLab concerns the development and test of new coupling methods or strategies. Operator splitting can be performed with various methods including for example adaptative time stepping (Belfort et al., 2007; Gasda et al., 2011). Global implicit approaches that separate geochemical and transport software might also be more widely tested providing the Jacobian of the chemical operator and taking into account current limitations such as the difficulties to model precipitation/dissolution reactions (Amir and Kern, 2010). TReacLab may then be used as a platform where interfaces to chemical and transport operators are available and have been tested and documented for other coupling methods.

These applications are possible because TReacLab is a fully free and open software that can be directly accessed and downloaded (https://github.com/TReacLab/TReacLab). The free and open use of TReacLab has been dominant in its development and in the choices made for its organization. The repository thus provides two main directories with sources and examples respectively. Sources are organized in four main categories for chemistry, transport, coupler and utilitaries. At the root of the chemistry, transport and coupler directories are the virtual classes as main entries. Examples of instantiations are provided in the subdirectories. Additional developments may take advantage of the documented examples provided at the different levels of the software.

**6. Conclusion**

We provide in the TReacLab code a fully segmented implementation of the coupling of independent geochemical and transport software. Coupling is based on a general expression of
the split-operator strategy with a set of classical methods. TReacLab should facilitate the
development of reactive transport simulation capacities for independent reactive and transport
software. Systematic comparison to the well-established PHREEQC model for uniform 1D
reactive transport cases shows a good agreement with TReacLab. Systematic comparison
against COMSOL for the 2D problem shows that full decoupling at the implementation level
has a cost in accuracy. Sharp dissolution fronts of thermodynamically controlled reactions
especially are generally smoothed in split-operator strategies. Steeper fronts might be
recovered with smaller splitting time steps at larger computational costs. Beyond the
implementation and the simulation capacity, consistency and validity of the numerical models
should be systematically assessed. TReacLab can be freely accessed and used to promote the
development of coupling methods and to provide additional modeling capacity for reactive
transport coupling in geological media.

Appendix A: Implemented operator splitting methods

We detail the mathematical formulation for the sequential splitting (Geiser, 2009):

$$\frac{\partial z^1}{\partial t} = L_1 z^1, \quad z^1(x, t^n) = z(x, t^n), \quad t^n \leq t \leq t^{n+1}. \quad (A.1)$$

$$\frac{\partial z^2}{\partial t} = L_2 z^2, \quad z^2(x, t^n) = z^1(x, t^{n+1}), \quad t^n \leq t \leq t^{n+1}. \quad (A.2)$$

$$z(x, t^{n+1}) = z^2(x, t^{n+1}). \quad (A.3)$$

the additive splitting (Faragó et al., 2008a; Faragó et al., 2008b):

$$\frac{\partial z^1}{\partial t} = L_1 z^1, \quad z^1(x, t^n) = z(x, t^n), \quad t^n \leq t \leq t^{n+1}. \quad (A.4)$$
\[
\frac{\partial Z^2}{\partial t} = \mathcal{L}_2 Z^2, \quad Z^2(x,t^n) = Z(x,t^n), \quad t^n \leq t \leq t^{n+1}, \quad (A.5)
\]

\[
Z(x,t^{n+1}) = Z^2(x,t^{n+1/2}) + Z^2(x,t^{n+1/2}) - Z(x,t^n), \quad (A.6)
\]

467 the Strang splitting (Strang, 1968):

\[
\frac{\partial Z^1}{\partial t} = \mathcal{L}_1 Z^1, \quad Z^1(x,t^n) = Z(x,t^n), \quad t^n \leq t \leq t^{n+1/2}, \quad (A.7)
\]

\[
\frac{\partial Z^2}{\partial t} = \mathcal{L}_2 Z^2, \quad Z^2(x,t^n) = Z^2(x,t^{n+1/2}), \quad t^n \leq t \leq t^{n+1}, \quad (A.8)
\]

\[
\frac{\partial Z^3}{\partial t} = \mathcal{L}_1 Z^3, \quad Z^3(x,t^{n+1/2}) = Z^2(x,t^{n+1}), \quad t^{n+1/2} \leq t \leq t^{n+1}, \quad (A.9)
\]

\[
Z(x,t^{n+1}) = Z^1(x,t^{n+1}). \quad (A.10)
\]
and the symmetrically weighted splitting (SWS) (Csomós et al., 2005):

\[ \frac{\partial Z^k}{\partial t} = \mathcal{L}_k Z^k, \quad Z^k(x, t^n) = Z(x, t^n), \quad t^n \leq t \leq t^{n+1}, \quad (A.11) \]

\[ \frac{\partial Z^k}{\partial x} = \mathcal{L}_2 Z^k, \quad Z^k(x, t^n) = Z^1(x, t^{n+1}), \quad t^n \leq t \leq t^{n+1}, \quad (A.12) \]

\[ \frac{\partial Z^{k-1}}{\partial t} = \mathcal{L}_2 Z^{k-1}, \quad Z^{k-1}(x, t^n) = Z(x, t^n), \quad t^n \leq t \leq t^{n+1}, \quad (A.13) \]

\[ \frac{\partial Z^{k-1}}{\partial x} = \mathcal{L}_1 Z^{k-1}, \quad Z^{k-1}(x, t^n) = Z^2(x, t^{n+1}), \quad t^n \leq t \leq t^{n+1}, \quad (A.14) \]

\[ Z(x, t^{n+1}) = \frac{Z^2(x, t^{n+2}) + Z^1(x, t^{n+1})}{2}. \quad (A.15) \]

The alternating splitting algorithm (Valocchi and Malmstead, 1992) is based on a sequential splitting. It is defined by two successive splitting time steps with a permutation of the operator sequence between the splitting time steps.

**Appendix B: Complementary notes on software organization**

We successively describe the general toolbox organization, the coupler, transport and chemistry classes. We concretely show how operator splitting methods can be introduced and how other transport and geochemical codes can be connected.

**B.1 Coupling methods**

The coupler is at the center of TReacLab as it performs the temporal integration and calls the transport and chemistry solvers through the OS algorithm. In the pre-processing phase, it gets the initial conditions and the temporal constrains of the integration. It is also in charge of
storing the required results before formatting and outputting them in the post-processing phase. Because the coupler is at the core of the toolbox, its methods remain generic. Interactions with the transport and chemistry solvers are also fully generic thanks to template interfaces calling external software and managing the exchange of information. Calling external software relies on the so-called Solve_Engine method for both transport and chemistry software. Solve_Engine takes as inputs the concentration data and the time step over which the integration must be performed. It returns the updated concentrations, a flag to check the success of the integration and an error message in case of failure to activate and inform the error management procedure mentioned in the former section. The coupler is based on a fixed structure of concentration data. Whatever the structure of concentrations in the transport and chemical codes, the structure of concentrations within the coupler is always the same. It consists in a matrix with in columns chemical species and in rows the position within the domain (Figure B.1). The size of the matrix is equal to the number of cells times the number of chemical species and components passed through the coupler. Chemical species include solutes and fixed species. As this is the sole link between the chemical code and the coupler also in charge of temporary results storage for the post-processing, it must transfer all quantities necessary for the algorithm and for the later extraction. The format of the matrix is set in the pre-processing phase and it is fixed for the whole simulation. TReacLab does not support yet any modification of species number to transfer between codes. Even if some solute species are absent over some time of the simulation, they will be transferred. This choice does not limit the capacity of the software as long as the chemical system is known from the beginning but might have some consequences on its performance in cases where solute composition strongly evolves. The choice of generality and flexibility, here like in other places, has a cost in efficiency. All modifications of concentration format are eventually performed in the interfaces between the coupler and the transport and chemistry solvers.
Figure B.1: Concentration format internal to the coupler class. To ensure generality, this structure of concentration is always the same and does not depend on the external transport and chemistry software. Species concentration are given in columns and are passed to the transport software as such. Concentrations at given locations are stored in rows with both mobile and fixed species. They are transferred either line per line or globally to the chemistry software. Fixed species are transferred from the chemistry code to the coupler to enable their possible use in the post-processing phase for results and outputs.

Thanks to the template methods calling the transport and chemical solvers and to the generic concentration format, operator splitting methods can be simply implemented. These are not more than a combination of simple calls of solvers passing and updating concentration information. Several sequential non-iterative techniques have thus been implemented, as detailed in section 2.3.
Specifications of the coupler are thus the name of the coupling method necessary to switch to the corresponding method in the coupler class, the temporal constrains of the integration and a vector of additional parameters. Temporal constrains of the integration are not only the initial and final times of the integration but also the times at which the solution must be stored. All time related parameters are stored into a time class. Additional parameters may be tolerances for example when using sequential iterative approaches. Instantiation of the coupler class thus consists in providing the identifier of the chosen coupling technique, the time constrains in the time class (initial time, final time, time to save the results, OS time step) and the additional parameters possibly needed by the algorithm.

B.2 Geochemical solver

Geochemical codes widely differ by their principles, the type of reactivity they consider and their input/output formats and parameters. We propose to normalize some of their interface to simplify exchanges with the coupler. In any case of equilibrium or kinetic reactions or of a mixed combination of them, geochemical codes steadily take concentrations, reaction constants, rate parameters, reaction times, as inputs and return output concentrations. All specifications linked to the choice of components, primary and secondary species should be set in the geochemical code or in the interface so that the geochemical solver does not have to be modified and the coupler remains generic. Whether components are used or not, the definition of the chemical system is not unique. Even when components are used, several alternative and reliable definitions can be chosen (Fang et al., 2003; Hoffmann et al., 2012; Molins et al., 2004). Numerical and conceptual consistencies between the transport and chemical systems should thus be ensured externally before any implementation.

While solute concentrations are instantiated by the coupler and systematically passed to the geochemical solver, equilibrium and kinetic constants are considered as constant. They are
defined once for all in the pre-processing phase. For example in PHREEQC, chemical reactions and constants are already defined in databases like 'Phreeqc.dat' or 'llnl.dat'. Initialization of mineral quantities is done at the beginning of the simulation when setting the initial conditions through the coupler. The interface between the coupler and the geochemical solver is made up of the Solve_Engine that calls the geochemical solver and the methods that modify the concentration format. By default, the geochemical solver is instantiated and stored for each of the nodes of the computational grid for the whole domain of the simulation. Any data that are not passed to the coupler is, in general, kept in the instances of the geochemical code. Another option is provided by software that allow simultaneous computations for several independent batches like it is for example the case of PHREEQC. In such cases only one instance of the geochemical solver is necessary. Exchanges of data between the coupler and the geochemical solver are defined in the pre-processing phase and remain fixed for the whole duration of the simulation. It is precisely at this stage that components are derived through the algebraic operations of equation (6) and passed to the coupler. The coupler does not manage the transformation of concentration and species but just their transfer between the transport and geochemical solvers. The use of components does not fundamentally change the calling sequence of the geochemical operator but modifies its interface to the coupler. Components may be specified by the geochemical code like in PHREEQC or by the user in the pre-processing phase by loading the matrix of $U$ (equation (6)). In this latter case, components are defined by the user in the pre-processing phase and are computed by the interface that adapts the information to be passed through the coupler to the transport solver.

Connection of a new geochemical code requires essentially four operations. First, a new daughter class of the template chemistry class must be defined. It can be built up using, as template, one of the examples provided and described in the section 4. Second, an interface must be created to filter the required information given from the coupler to the Solve_Engine.
method. Third, an instantiation procedure should be provided whether it is internal or external to TReacLab. Fourth, the template Solve_Engine calling function of the geochemical solver must be written and optionally tested before being effectively used in reactive transport problems.

B.3 Transport solver

Despite the diversity of the transport mechanisms and numerical schemes to solve them, we provide here a basic interface designed mostly to address transport in a generic way. As previously stated, this approach assumes that transport parameters are not modified by the species concentration. This absence of feedback currently precludes density driven flows as well as permeability and porosity modifications due to precipitation or dissolution. TReacLab might be extended in this direction on the basis of slow evolutions of porosity or density. The transport operator relies on concentration independent parameters. We detail in the following the interaction between the coupler and the transport classes with the exchange of data and the instantiation of the transport solver. We will conclude this section with the development required to connect other transport codes.

While geochemical codes operate on species concentration on a given computational node, transport codes operate on a given species concentration over all the domain. In terms of data structure, each of the columns of the concentration array are successively transferred to the geochemical code and each of the rows (or linear combinations of rows) are given to the transport code (Figure B.1). The transport operator is thus iteratively called for each of the species or components explicitly specified in the interface between the coupler and the transport solver (Figure 1). The time range over which temporal integration should be performed and the identifiers of the transported species are also transferred to the transport solver. Species identification is essential when considering species sensitive diffusion
coefficient. The transport solver returns the updated concentration field at the final time of the
time range, an indicator of success or failure of the integration and a message to document
algorithm failures. The basic exchange of concentrations with the imposed integration times
are the sole requirements for the coupler to proceed.

All other parameters of the transport code should be set in the pre-processing phase, which
may become an important part of the eventual reactive transport code. In fact it does not cover
only the flow and transport parameters but more broadly the full structure of the domain, of
the computational grid, and of the boundary conditions. As for the geochemical code, the
transport code can be instantiated internally or externally. In case of internal definition, it
should contain at least the flow and transport properties, the morphology of the domain and
the structure of the computational grid (coordinates of the computational nodes). A default set
of classes is provided for 1D problems as templates for the morphology (domain definitions),
the computational grid (identification and coordinates of nodes and edges), the boundary
conditions (nature and values for boundary conditions) and the hydraulic and transport
properties. We recall as also said in section 2.2 that some operator splitting techniques might
impose limitations on the transport solver in terms of integration scheme or in terms of time
step (de Dieuleveult et al., 2009). Both the OS technique and the transport integration should
be chosen consistent.

Operations on the transport class are thus decomposed between the pre-processing and the
processing phases. Specifications of the operator with all necessary parameters is performed
in the pre-processing phase. Only generic exchanges of concentrations are needed in the
processing phase. Additional information would generally be needed externally to identify the
location of the computational nodes. More advanced information from the definition of the
domain, parameters and boundary conditions will be generally defined in the transport code
rather than in TReacLab. For example, Comsol or Modflow have their own grid definitions. They are complete and efficient. It may be straightforwardly extracted and cross-referenced with the results of TReacLab as long as the cell numbers correspond, a basic but necessary requirement. This choice is motivated by both the generality and the simplicity of TReacLab. It also highlights that TReacLab remains a coupler that transfers information and does not process in any way the relation of concentrations between cells like a transport operator does.

The methodological choice of handling the spatial dimension of the problem within the transport operator is not only operational. It is also ensuring the capacity to connect a wide range of transport codes with their own logic and structure. For example, the multi-physics software COMSOL has its own mesh generator methods and internal structures that should not be duplicated in TReacLab but interfaced. Connecting other codes would thus require reduced work as long as they can already be called from the same environment of development (here MATLAB) on a discretized time basis. More in details, any new transport code would require: 1) the development of the main calling function Solve_Engine to call it from the coupler 2) the adaptation of the concentration format in the interface methods that match the concentrations to the internal data structure of the external code, 3) the instantiation of the transport class and 4) the access to the coordinates of the computational nodes for outputs purposes. As for the geochemical code, implementation of the interface should be checked before any full reactive transport coupling. This can be completed within TReacLab by using an idle process instead of the geochemical code.

Acknowledgements: We acknowledge ANDRA and the ANR project H2MNO4 under the number ANR-12-MONU0012-01 for their financial support, Jocelyne Erhel and David L. Parkhurst for constant and fruitful discussions, and Javier Molinero for initial exchanges.
REFERENCES


Steefel, C.I., MacQuarrie, K.T., 1996. Approaches to modeling of reactive transport in porous media. Reviews in Mineralogy and Geochemistry 34, 85-129.


Highlights

- Object-oriented implementation of non-intrusive couplings for reactive transport
- Validation for 3 equilibrium and kinetically controlled 1D cases with PhreeqC and one 2D case with COMSOL.
- Illustration of implementation flexibility with different transport and reaction codes