Analysis of a global reactive transport model and results for the MoMaS benchmark

Jocelyne Erhel, Souhila Sabit

Analysis of a global reactive transport model and results for the MoMaS benchmark

Jocelyne Erhel[a,*], Souhila Sabit[a]

[a]*Inria, Campus de Beaulieu, 35042 Rennes Cedex, France*

Abstract

Reactive transport models are very useful for groundwater studies such as water quality, safety analysis of waste disposal, remediation, and so on. The MoMaS group defined a benchmark with several test cases. We present results obtained with a global method and show through these results the efficiency of our numerical model.

Keywords: reactive transport, global method, MoMaS benchmark.

1. Introduction

It is quite challenging to develop a numerical model for deep storage of nuclear waste. The time interval is very large (several thousands years), models are coupled and simulations must be accurate enough to be used for risk assessment. In most cases, chemistry must be included in models of deep geological storage. In addition to radioactive decay, chemical phenomna are numerous and include aqueous reactions, oxydo-reduction reactions, precipitation and dissolution reactions, ions exchanges, surface exchanges. These reactions can be either kinetic or at equilibrium.

Models must handle species which are in groundwater systems and take into account the mobile property of these species. It is thus necessary to consider a coupled model, where chemistry equations and radioactive decay are combined

---

*. Corresponding author
Email addresses: `jocelyne.erhel@inria.fr` (Jocelyne Erhel), `sabitsouhila@yahoo.fr` (Souhila Sabit)

with transport of contaminants. These models are partial differential equations (for transport, one equation for each species), and algebraic or differential equations (for chemistry, a system at each grid point).

The MoMaS benchmark was designed as a set of academic examples, in order to run experiments with several methods and software [2], [4]. Several authors participated in the exercise [1], [7], [13],[14], [15]. A synthetic comparison of their results indicate that, for this benchmark, the fastest results were obtained with global approaches [3].

In this paper, we show original results obtained with a global approach for the so-called 2D easy test case of the MoMaS benchmark. The model proposed in MoMaS is based on the introduction of total analytical concentrations, thanks to the linearity of the transport equation. It is a set of Partial Algebraic Differential Equations. We use the method developed in [6, 8] and improved in [10, 18], where we first discretize in space, using a Finite Difference scheme, then discretize in time, using an implicit multistep scheme, of the BDF family (Backward Differentiation Formula). Thus at each time step, we have to solve nonlinear equations which we keep coupled.

Compared with [8], three improvements were brought in [10, 18, 9]. First, a substitution technique, similar to global DSA methods, allows reducing the size of nonlinear or linear systems. The differential variables are kept in the semi-discrete system, in order to use adaptive time steps and adaptive Jacobian updates. Logarithmic variables are very convenient to ensure the positivity of the concentrations and to compute the derivatives, but they can lead to severely ill-conditioned Jacobian matrices [18]. Thus, the second improvement consists in using non logarithmic variables, at the price of ensuring positivity during nonlinear iterations. In the benchmark, the first component is inert, so the third improvement is to remove this component from the coupled equations.

The paper is organized as follows. In section 2, we describe the mathematical model of the MoMaS test case studied and provide our simulation results for flow and transport of the five main species. The numerical method is defined in section 3, as well as three versions of our software GRT3D [18]. Finally,

2

we analyse in section 4 the efficiency of the improvements brought to the original global method. We discuss these results and outline future work in the concluding section 5.

2. MoMaS benchmark

The MoMaS group studies mathematical models and numerical simulations for nuclear waste disposal. A set of test cases were defined for transport reactive problems [4]. Here, we make experiments with the so-called easy test case, in 2D. The computational domain is a rectangle with two porous media, see Figure 1. All dimensions are normalized, with length unit L and time unit T.
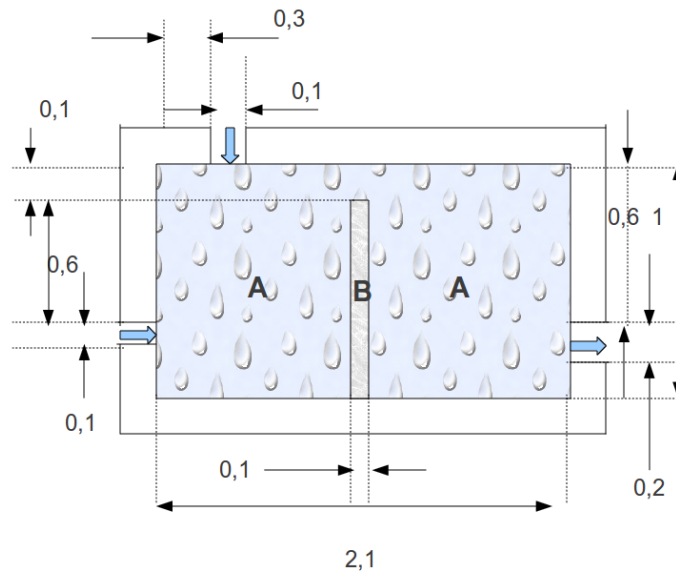

FIGURE 1: Computational domain for MoMaS Benchmark (dimensions are in the unit length L).

2.1. Flow simulations

The benchmark considers a steady saturated one-phase flow, with no source term. Flow is governed by Darcy's law and mass conservation, giving mathe-

matical equations (1) where the pressure h and the Darcy's velocity q are the unknowns and **K** is the hydraulic conductivity.

$$\begin{cases} q = -\mathbf{K}\nabla h, \\ \nabla q = 0. \end{cases} \tag{1}$$

Darcy's velocity is related to the porosity ε and to the pore velocity v by $q = \varepsilon v$.

Medium A has a high conductivity and a low porosity, whereas medium B has a low conductivity and a high porosity, see Table 1.

	Medium A	Medium B
Porosity ε	0.25	0.5
Conductivity K (L.T^{-1})	10^{-2}	10^{-5}

TABLE 1: Flow conditions for MoMaS benchmark.

Boundary conditions must be prescribed to complete the PDE system. At outflow, the pressure is given by $h = 1L$. At both inflows, the flow velocity is given by $q = 2.25 \times 10^{-2} L.T^{-1}$, whereas no flow condition is applied at other boundaries.

We simulate the flow equations with the software MODFLOW [11]. Figures 2 et 3 show the velocity and the pressure computed with a mesh of 40×84 cells [6], [18].

2.2. Transport simulations

Transport of mass in groundwater is governed by advection and dispersion. There is no source term in the benchmark test. The concentration c of an inert solute is the solution of the PDE (2) expressing a mass conservation law.

$$\varepsilon \frac{\partial c}{\partial t} = \nabla.(D\ \nabla c) - \nabla.(q\ c), \tag{2}$$

where the dispersion tensor D is given by

$$D = \varepsilon d_m I + \alpha_T \|q\| I + (\alpha_L - \alpha_T) \frac{qq^T}{\|q\|}.$$
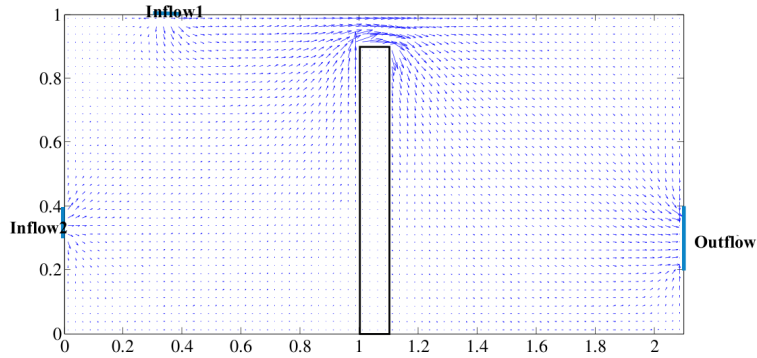
FIGURE 2: Velocity field with a mesh of 40×84 cells.

FIGURE 3: Pressure head with a mesh of 40×84 cells.

We consider the advective test case of the MoMaS benchmark, without molecular diffusion, see Table 2.

Initial conditions at time $t = 0T$ are applied to the geochemistry system, see Table 5. The final time is $t = 6000T$.

At impermeable boundaries, a no total flux condition is imposed. At ouflow, a zero concentration gradient is imposed. At both inflows, concentration is prescribed, with values given in Table 6. Injection occurs during a first period of time until $t = 5000T$, followed by a leaching period until the end.

5

	Medium A	Medium B
Molecular diffusion d_m (L^2T^{-1})	0	0
Longitudinal Dispersion $\alpha_L(L)$	10^{-2}	6×10^{-2}
Transverse Dispersion $\alpha_T(L)$	10^{-3}	6×10^{-3}

Table 2: Transport conditions for MoMaS benchmark.

The first chemical component is a spectator ion, which behaves like a inert solute. For this component, we can simulate the transport equations with the software MT3D [20]. Figure 4 represents the concentration at various times, obtained with a fine mesh of 80×168 cells.

2.3. Geochemistry simulations

In the MoMaS test case, the geochemical system has $N_c = 4$ aqueous components c_j and $N_s = 1$ fixed component s_j. They react with $N_\alpha = 5$ aqueous secondary species α_i and $N_\beta = 2$ fixed secondary species β_i. There is no precipitation dissolution. All the coefficients of activities are equal to 1 and the variables $c_j, s_j, \alpha_i, \beta_i$ are the concentrations of the species.

Stoichiometric coefficients and equilibrium constants are given in Table 3. In the original benchmark, the constant $K_{\alpha 5}$ is equal to 10^{+35} but we replace it by 10^{+6}. Indeed, with such a very large constant, the chemical nonlinear system is highly ill-conditioned and inaccuracies may appear for some values of c and s. For example, we succeeded to run 1D simulations with $K_{\alpha 5} = 10^{+35}$ [18], but we experienced numerical difficulties in 2D.

Table 3 is summarized in Table 4 by using algebraic notations, with the matrices S, A, B and the vectors $c, s, \alpha, \beta, K_\alpha, K_s$.

Secondary species can be computed thanks to the mass action laws (3) describing the chemical reactions. Moreover, the concentrations of the components

6

	c_1	c_2	c_3	c_4	s	K
α_1	0	-1	0	0	0	10^{-12}
α_2	0	1	1	0	0	1
α_3	0	-1	0	1	0	1
α_4	0	-4	1	3	0	0.1
α_5	0	4	3	1	0	10^{+6}
β_1	0	3	1	0	1	10^{+6}
β_2	0	-3	0	1	2	10^{-1}

TABLE 3: Stoichiometric coefficients and equilibrium constants for MoMaS benchmark (with $K_{\alpha 5}$ modified).

	c	s	K
α	S	0	K_α
β	A	B	K_s

TABLE 4: Algebraic representation of Table 3.

must be positive.

$$\begin{cases} \alpha_i(c) = K_{\alpha i} \prod_{j=1}^{N_c} c_j^{S_{ij}}, \quad i = 1, \ldots N_\alpha, \\ \beta_i(c,s) = K_{\beta i} \prod_{j=1}^{N_c} c_j^{A_{ij}} \prod_{j=1}^{N_s} s_j^{B_{ij}}, \quad i = 1, \ldots N_\beta, \\ c_j \geq 0, \quad j = 1, \ldots N_c, \\ s_j \geq 0, \quad j = 1, \ldots N_s. \end{cases} \quad (3)$$

Then the model applies the mass conservation law to the chemical system.

$$\begin{cases} c + S^T \alpha(c) + A^T \beta(c,s) = T, \\ s + B^T \beta(c,s) = W, \end{cases} \quad (4)$$

where T and W are respectively the total analytical concentrations for mobile and fixed components. In a closed system, these quantities are known, but they vary in time and space when the species are transported by water.

7

2.4. Reactive transport

Transport is now governed not only by advection and dispersion, but also by reaction. A mass conservation equation can be written for each component and each secondary species, where the reaction terms are unknowns in the context of reactions at equilibrium. However, because the dispersion tensor is the same for all the species and the transport terms are linear, it is possible to compute a linear combination of all the equations. This ends up to mass conservation equations (5) applied to the total analytical concentrations T, where the reaction terms cancel and disappear. Moreover, the total analytical concentrations W remain constant and given.

$$\varepsilon \frac{\partial T_j}{\partial t} = \nabla.(D \ \nabla C_j(c)) - \nabla.(q \ C_j(c)), \ j = 1, \ldots N_c, \tag{5}$$

where $C(c) = c + S^T \alpha(c)$ is the total mobile concentration.

Equations (5) are coupled with the mass action laws (3) and the mass conservation laws (4) written at each point of the computational domain.

Initial conditions for the 5 components are given in Table 5. It can be noted that $T_3 = 0$ implies that $c_3 = \alpha_2 = \alpha_4 = \alpha_5 = \beta_1 = 0$, because all the associated stoichiometric coefficients are strictly positive. Therefore, the equilibrium constants $K_{\alpha 2}, K_{\alpha 4}, K_{\alpha 5}, K_{\beta 1}$ have no mathematical effect on the initial equilibrium. Thus the original initial conditions of components remained the same when we changed the value of $K_{\alpha 5}$.

Boundary conditions at both inflows are given in Table 6 for the injection and leaching periods. Again, it can be noted that $T_3 = 0$ during the leaching period, and $T_4 = 0$ during the injection period. Since the stoichiometric coefficients associated to c_4 are also strictly positive, $c_4 = \alpha_3 = \alpha_4 = \alpha_5 = \beta_2 = 0$ during the injection period. Thus, at inflow, the equilibrium constants $K_{\alpha 4}, K_{\alpha 5}, K_{\beta 2}$ have no mathematical effect on the equilibrium.

Figure 5 shows the concentrations of the aqueous components c_4 at various times, using a mesh of size 80×168. They cannot be compared exactly to the results of the literature [2, 13, 14, 15], because we changed the constant $K_{\alpha 5}$,

	T_1	T_2	T_3	T_4	W
Medium A	0	-2	0	2	1
Medium B	0	-2	0	2	10

TABLE 5: Initial conditions for MoMaS benchmark.

	T_1	T_2	T_3	T_4	W
Injection $t \in [0, 5000]$	0.3	0.3	0.3	0	0
Leaching $t \in [5000, 6000]$	0	-2	0	2	0

TABLE 6: Inflow conditions for MoMaS benchmark.

but they are very close. More results can be found in [18].

3. Numerical method

Equations (3),(4),(5) form a PDAE system composed of algebraic and PDE equations. We assume that it is well-posed.

Following a method of lines, we first discretize in space, using a finite difference scheme implemented in the software MT3D. We define a regular mesh of the domain with a rectangle of $N_x \times N_y$ cells. We get a DAE semi-discrete system, where the algebraic chemistry equations are written at each point of the mesh [6, 8].

Because the system is stiff, we use an implicit scheme, involving a global nonlinear system at each time step, coupling transport and chemistry equations. We use a BDF scheme implemented in the software SUNDIALS [12]. At each time step, the nonlinear system is solved with a Newton's method. We provide the function of the DAE equation and its derivative, whereas the module IDA of SUNDIALS provides the implicit scheme.

Time discretization comes with an adaptive time step, which controls both the accuracy of the approximation and the convergence of Newton's iterations. It allows choosing large time steps when possible and saves CPU time.

9

We solve the linearized equations with a sparse direct solver, implemented in the software library UMFPACK [5], which we have interfaced with SUN-DIALS. We thus factorize the Jacobian matrix and use the triangular factors to solve the linearized equations. The software keeps the Jacobian of linearized Newton's iterations frozen while convergence is fast enough, saving updates and factorizations thus CPU time.

We implemented three versions of our method in the software suite GRT3D [19, 10, 18]. Mass action laws can be linearized by introducing logarithmic variables, if they are strictly positive. We introduce $\hat{c} = \log(c)$, $\hat{s} = \log(s)$, so that equation (3) becomes (written with matrices and vectors) :

$$
\begin{cases}
\alpha(\hat{c}) = \exp\left(\log K_\alpha + S\ \hat{c}\right), \\
\beta(\hat{c}, \hat{s}) = \exp\left(\log K_\beta + A\ \hat{c} + B\ \hat{s}\right),
\end{cases}
$$

and c, s are replaced by $\exp(\hat{c}), \exp(\hat{s})$ in equation (4).

In a first version, called GRT3D, we used as primary variables the total T, the total mobile C and the components concentrations $\hat{c}, \hat{s}$, so that we had $(3N_c + N_s)$ variables at each cell. In a second version, called GRT3DRL, we reduced the size of linearized systems by using a substitution approach in the linear systems to be solved. We eliminated the variables T and C and kept only the variables $\hat{c}, \hat{s}$, thus $(N_c + N_s)$ variables at each cell. This technique saves CPU time by factorizing a smaller matrix. In a third version, called GRT3DRSL, we used the same algorithms, but with the non logarithmic variables c, s. We also eliminated the variables T and C in the linearized equations. In this case, we have to ensure the positivity of concentrations. This is done in Newton's method, where approximate concentrations are forced to be positive at each iteration.

4. Performance analysis

In this section, we analyze the accuracy and the computational time of our simulations, using the three versions of our software. Initial and boundary conditions are changed from 0 to 10^{-10} when necessary for logarithmic variables

(GRT3D and GRT3DRL). All experiments are done in sequential, on a Intel Xeon computer, with 24 MB of RAM and with 12 MB of cache memory.

The rectangle is meshed by a regular grid of $N_m = N_x \times N_y$ cells. Simulations are done with three meshes : a coarse mesh with $N_m = 20 \times 42$ cells, an intermediate mesh with $N_m = 40 \times 84$ cells, and a fine mesh with $N_m = 80 \times 168$ cells.

Here we show a selection of the results which can be found in [18].

4.1. Accuracy

Mesh	with the first component		without the first component	
	GRT3D	GRT3DRL	GRT3D	GRT3DRL
20x42	6.402528E-08	3.112242E-09	3.227338E-08	2.568312E-09
40x84	6.432335E-08	6.467349E-09	3.040179E-08	3.034366E-09
80x168	5.802496E-08	5.286914E-09	6.091884E-08	5.727954E-09

TABLE 7: Comparison of accuracy for the component c_2. The difference with GRT3DRSL is computed from (6).

For each component, we compare the concentrations obtained with the three versions of the software, keeping or removing the inert component. The algorithms are mathematically equivalent but not numerically, therefore we compare their accuracy. We take as reference the result obtained with GRT3DRSL, that is the version with substitution and without logarithms. For each component c_i or s_i, we compute the difference between this reference and the results obtained with other versions, in the following way :

$$E_{c_i} = \sqrt{\frac{1}{N_m N_t} \sum_{n,j} (\tilde{c}_i(m_j, t_n) - c_i(m_j, t_n))^2}, \qquad (6)$$

where m_j is the cell number and t_n is the timestep number, with N_m cells and N_t timesteps, $\tilde{c}_i$ is the reference concentration and c_i is the other concentration.

11

Table 7 shows the deviations for the component c_2, those for the other components can be found in [18]. Since these differences are quite small for the three meshes and for all versions, we conclude that we do not introduce numerical artefacts by using substitution or by using logarithms.

4.2. System size and total CPU time

Mesh	with the first component		without the first component	
	GRT3D	GRT3DRL	GRT3D	GRT3DRL
20x42	10920	4200	8400	3360
40x84	43680	16800	33600	13440
80x168	174720	67200	134400	53760

TABLE 8: System size using GRT3D and GRT3DRL with and without the first inert component.

The reduction of the system size is given in Table 8. If we keep the inert component and apply substitution, the system size is reduced from $(3N_c + N_s) \times N_m = 13N_m$ in GRT3D to $(N_c + N_s) \times N_m = 5N_m$ in GRT3DRL and GRT3DRSL. If we remove the inert component, we get $(N_c - 1)$ aqueous components, and obtain a reduction thanks to substitution from $10N_m$ to $4N_m$.

Mesh	GRT3D	GRT3DRL	GRT3DRSL
20x42	52 m 52 s	27 m 16 s	19 m 6 s
40x84	9 h 52 m 24 s	4 h 44 m 40 s	3 h 9 m 8 s
80x168	5 j 15 h 12 m 15 s	3 j 11 h 24 m 23 s	2 j 2 h 12 m 5 s

TABLE 9: CPU time of the three versions with the first inert component.

Clearly, reducing the system size has a direct impact on the CPU time. The total CPU time for the three versions is given in Table 9 for computations with the first inert component and in Table 10 without it.

Mesh	GRT3D	GRT3DRL	GRT3DRSL
20x42	40 m 20 s	19 m 1 s	13 m 49 s
40x84	7 h 20 m 22 s	4 h 4 m 1 s	2 h 12 m 45 s
80x168	2 j 18 h 22 m 46 s	2 j 6 h 56 m 42 s	1 j 8 h 20 m 46 s

TABLE 10: CPU time of the three versions without the first inert component.

CPU time decreases also when variables are not logarithmic. In order to analyze these effects, we use the measures taken at each external timestep of the software SUNDIALS. In Figure 6, we plot the CPU time versus the simulated time, for the six simulations (three versions of software, with and without the first component), and for the fine mesh. During all the simulated time interval, the substitution approach, as well as the elimination of the first component, reduce the CPU time. Also, the use of non logarithmic variables is much more efficient.

For all the simulations, the CPU time increases rapidly until about $t = 1000T$ then it levels off until time $t = 5000T$ and increases again rapidly when the inflow boundary conditions change. We analyse the outputs of the software IDA in order to explain this behavior.

4.3. Effect of adaptive time step

The IDA solver in the software SUNDIALS adjusts automatically the time step, in order to control both the accuracy of time discretization and the convergence of Newton's iterations. In Figure 7, we plot the number of time steps taken during the whole simulation. Here, we show only results without the first component and for the fine mesh, since other results are very similar.

As expected, the CPU time is strongly correlated with the number of time steps. The initial and boundary conditions are difficult to handle and require small time steps during a period of time until larger time steps can be taken. These results demonstrate the efficiency of adaptive time step.

13

4.4. Effect of modified Newton's iterations

The number of time steps has a direct impact on the number of Newton's iterations which can be measured by the number of linear system solvings. It is also correlated to the number of updates of the Jacobian matrix, which can be measured by the number of matrix factorizations. Indeed, in our simulations, we use a direct sparse linear solver implemented in the software UMFPACK. Thus each time the Jacobian is updated, it has to be factorized in order to solve the linear systems associated to this new Jacobian.

In Figure 8, we plot the number of linear solvings and matrix factorizations, for the three versions and for the fine mesh. Theses numbers are in logarithmic scale. Here, the first component is not considered, similar results are obtained when including it.

Again, we observe a sharp increase of these numbers at the beginning of the simulation and a small jump after leaching. At time $t = 0$, the number of solvings as well as the number of factorizations are slightly smaller when reducing the system size, from GRT3D to GRT3DRL. They are clearly smaller when using non logarithmic variables. This result is probably due to a better condition number of the Jacobian matrix at time $t = 0$.

For the three versions, the number of matrix factorizations is much smaller than the number of linear system solvings. This result demonstrates the efficiency of adaptive modified Newton's iterations. Indeed, the Jacobian is kept frozen during several time steps and during nonlinear iterations, so that factorization does not occur as often as solving.

4.5. Algorithmic complexity of Newton's iterations

It is well-known that the algorithmic complexity of sparse matrix factorization is much higher than the complexity of sparse system solving, which involves sparse triangular matrices. Thus, in order to fully measure the efficiency of the adaptive update of the Jacobian matrix, we have to analyze the CPU time of the Newton's iterations.

14

In Figure 9, we plot the CPU time (in logarithmic scale) of the linear solvings and matrix factorizations, for the three versions and the fine mesh. We see again the effect of the numerical difficulties at time $t = 0$ and the differences between the three versions.

Whereas the number of factorizations is smaller than the number of solvings, the CPU time does not behave in the same way. For the small and intermediate meshes, factorization time is smaller than solving time but it becomes larger than solving time for the fine mesh. Because the algorithmic complexity of the factorization is higher, the CPU time increases much faster when the mesh is refined.

This is also illustrated by Figure 10, where we plot the percentage of time spent in linear steps in function of the mesh size, measured by the number of cells. These percentages are computed versus the total CPU time, at the end of the simulations. Computations are done here with the version without logarithms (GRT3DRSL) and without the first component c_1.

Factorization time becomes relatively more important as the mesh size increases and eventually the lines intersect roughly at the size of the fine mesh. It can also be noted that the addition of both percentages increases with the mesh size. For the fine mesh, more than 90% of CPU time is spent in the Newton's iterations. Therefore, it is really important to reduce the number of iterations and their computational cost.

5. Concluding remarks

In this paper, we have proposed a global approach for simulating reactive transport equations, where transport is coupled with geochemistry. The global method is based on a DAE formulation of the semi-discrete system obtained after space discretization. An implicit time scheme ensures stability and provides an adaptive time step with an adaptive update of the Jacobian matrix. This feature is very efficient as illustrated in our numerical results with the MoMaS benchmark. In order to use this adaptive time scheme, we keep the differential

15

variables, but we eliminate them at the linear level, thus reducing the system size. This is also very efficient as demonstrated in our results. We could also apply the substitution at the nonlinear level [10, 18]. In the MoMaS benchmark, some concentrations are set to 0 at the intial time and at inflow boundaries. When using logarithmic variables and very small initial concentrations instead of 0, this leads to ill-conditioned systems. It appears that it is more efficient to use nonlogarithmic variables for the test case studied here.

We use a sparse direct solver which first factorizes the Jacobian matrix, then solves two triangular systems. This second step has a much smaller complexity than the first one. Thanks to the adaptive update of the Jacobian, the computational cost of the factorization remains low for the meshes considered here. However, it is clear that this cost will eventually dominate for larger systems. Therefore, we investigate parallel iterative solvers in order to tackle 3D problems [17].

The test case considered here defines chemistry at equilibrium, without precipitation or dissolution. These reactions are very challenging because minerals can appear and disappear. We study some mathematical issues of this problem [16]. We also plan to include kinetic reactions.

Acknowledgements

References

[1] L. Amir, M. Kern, A global method for coupling transport with chemistry in heterogeneous porous media, Computational Geosciences 14 (2010) 465–481. URL : http://dx.doi.org/10.1007/s10596-009-9162-x. doi :10.1007/s10596-009-9162-x.

[2] J. Carrayrou, Looking for some reference solutions for the reactive transport benchmark of momas with specy, Computational Geos-

16

ciences 14 (2010) 393–403. URL : `http://dx.doi.org/10.1007/`
`s10596-009-9161-y`. doi :`10.1007/s10596-009-9161-y`.

[3] J. Carrayrou, J. Hoffmann, P. Knabner, S. Kräutle, C. de Dieuleveult, J. Erhel, J.V. der Lee, V. Lagneau, K. Mayer, K. MacQuarrie, Comparison of numerical methods for simulating strongly non-linear and heterogeneous reactive transport problems. the momas benchmark case., Computational Geosciences 14 (2010) 483–502.

[4] J. Carrayrou, M. Kern, P. Knabner, Reactive transport benchmark of momas, Computational Geosciences 14 (2010) 385–392. URL : `http://dx.doi.org/10.1007/s10596-009-9157-7`. doi :`10.1007/s10596-009-9157-7`.

[5] T. Davis, Algorithm 832 : Umfpack, an unsymmetric-pattern multifrontal method, ACM Transactions on Mathematical Software 30 (2004) 196–199.

[6] C. de Dieuleveult, Un modèle numérique global et performant pour le couplage géochimie-transport, Ph.D. thesis, University of Rennes 1, 2008.

[7] C. de Dieuleveult, J. Erhel, A global approach to reactive transport : application to the momas benchmark, Computational Geosciences 14 (2010) 451–464. doi :`DOI:10.1007/s10596-009-9163-9`.

[8] C. de Dieuleveult, J. Erhel, M. Kern, A global strategy for solving reactive transport equations, Journal of Computational Physics 228 (2009) 6395–6410. URL : `http://www.sciencedirect.com/science/article/` `B6WHY-4WGDR6C-1/2/ada7965dcd6096984365876b64411966`. doi :`DOI:10.` `1016/j.jcp.2009.05.044`.

[9] J. Erhel, S. Sabit, A global reactive transport model applied to the momas benchmark, in : B. Amaziane, E. Ahusborde, D. Barrera, J. Ibanez-Perez, R. Romero-Zaliz, D. Sbibih (Eds.), Proceedings of the 6th International Conference on Approximation Methods and Numerical Modelling in Envi-

17

ronment and Natural Resources : MAMERN'15, EUG, Pau, France, 2015, pp. 303–326.

[10] J. Erhel, S. Sabit, C. de Dieuleveult, Computational science, engineering and technology series, Computational Science, Engineering and Technology Series, volume 31, Saxe-Coburg Publications, 2013, pp. 151–169. URL : http://hal.inria.fr/hal-00846544.

[11] A. Harbaugh, E. Banta, M. Hill, M. McDonald, MODFLOW-2000, the U.S. Geological Survey modular ground-water model – User guide to modularization concepts and the Ground-Water Flow Process, Open-File Report 00-92, U.S. Geological Survey, 2000.

[12] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, , C.S. Woodward, SUNDIALS : Suite of nonlinear and differential/algebraic equation solvers, ACM Transactions on Mathematical Software 31 (2005) 363–396. Also available as LLNL technical report UCRL-JP-200037.

[13] J. Hoffmann, S. Kräutle, P. Knabner, 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, Computational Geosciences 14 (2010) 421–433. URL : http://dx.doi.org/10.1007/s10596-009-9173-7. doi :10.1007/s10596-009-9173-7.

[14] V. Lagneau, J. Lee, Hytec results of the momas reactive transport benchmark, Computational Geosciences 14 (2010) 435–449. URL : http://dx.doi.org/10.1007/s10596-009-9159-5. doi :10.1007/s10596-009-9159-5.

[15] K. Mayer, K. MacQuarrie, Solution of the momas reactive transport benchmark with min3p - model formulation and simulation results, Computational Geosciences 14 (2010) 405–419. URL : http://dx.doi.org/10.1007/s10596-009-9158-6. doi :10.1007/s10596-009-9158-6.

18

[16] T. Migot, J. Erhel, Analyse mathématique de modèles géochimiques, Technical Report, Inria, 2014.

[17] F. Pacull, P. Gibert, S. Sabit, J. Erhel, D. Tromeur-Dervout, Parallel preconditioners for 3d global reactive transport, in : Parallel CFD 2014, Trondheim, Norway. Invited keynote lecture.

[18] S. Sabit, Les méthodes numériques de transport réactif, Ph.D. thesis, University of Rennes, 2014.

[19] S. Sabit, N. Soualem, Suite Logicielle GRT3D (Global Reactive Transport 3D), Rapport de contrat Andra, INRIA, 2011.

[20] C. Zheng, P. Wang, MT3DMS : a modular three-dimensional multi-species model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems : documentation and user's guide, SERDP-99-1, U.S. Army Engineer Research and Development Center, 1999.
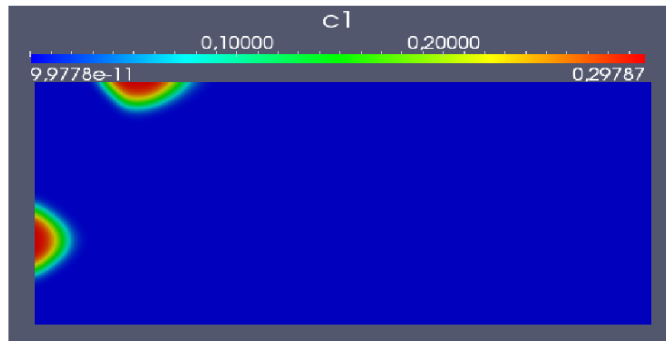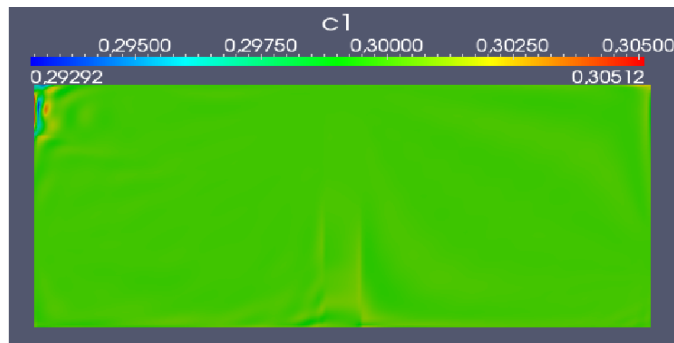
(a) Time $t = 1T$
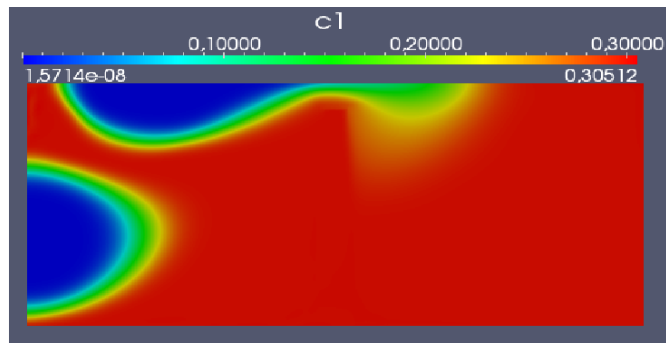
(b) Time $t = 4000T$
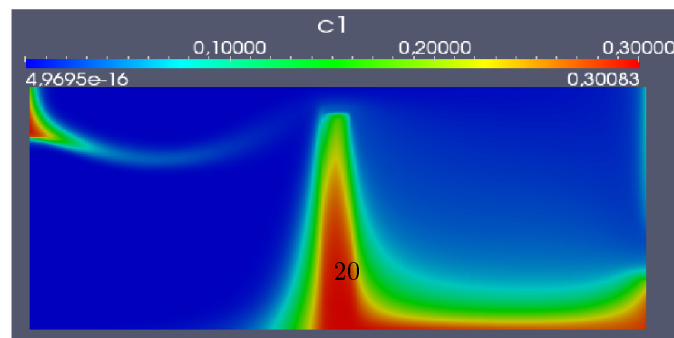
(c) Time $t = 5010T$

(d) Time $t = 5800T$

FIGURE 4: Concentrations of the inert solute c_1 at different times (with different scales). Simulations are done with a mesh of size 80×168.
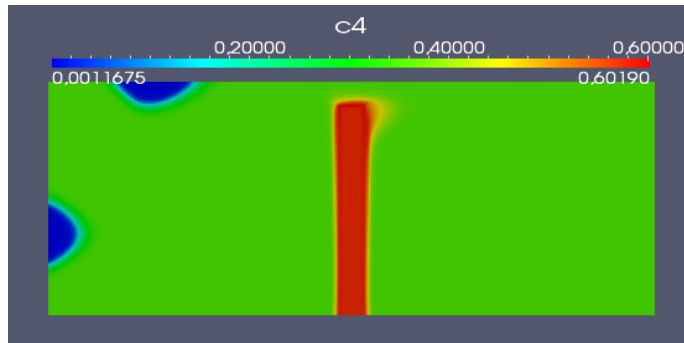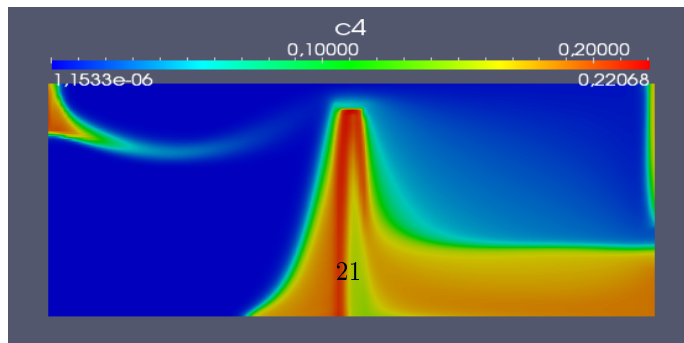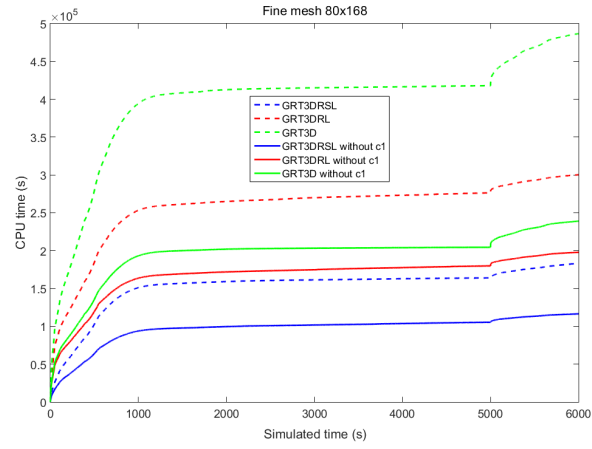
(a) Time $t = 1T$
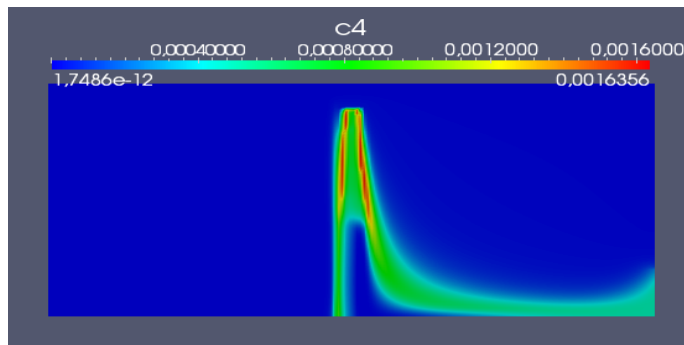
(b) Time $t = 4000T$
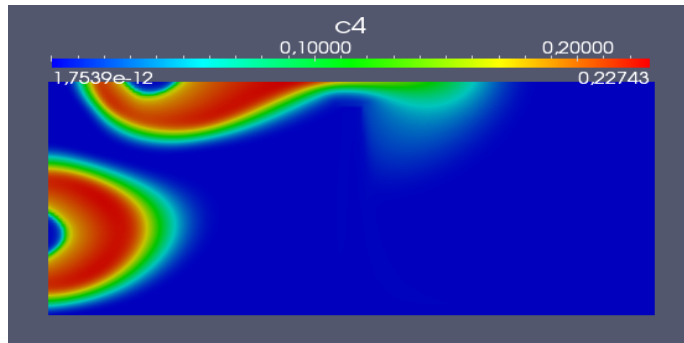
(c) Time $t = 5010T$

(d) Time $t = 5800T$

FIGURE 5: Concentrations of the solute c_4 at different times (with different scales). Simulations are done with a mesh of size 80×168.

FIGURE 6: CPU time during the simulation for the six variants.
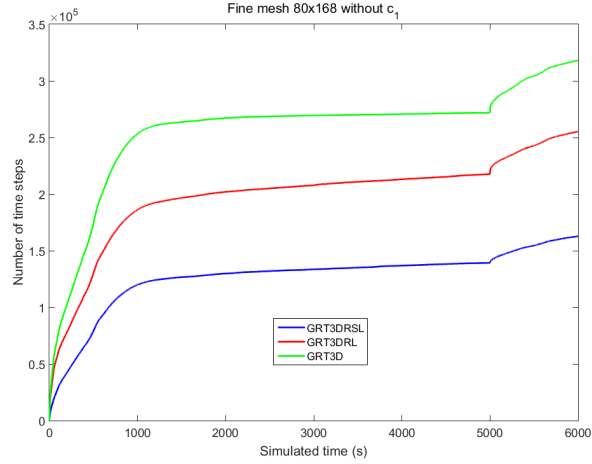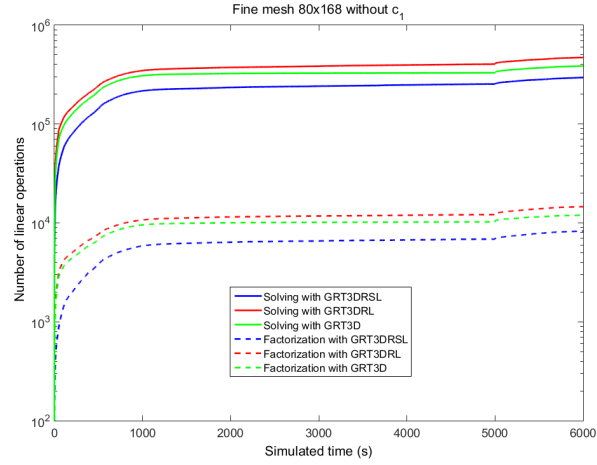

FIGURE 7: Number of time steps during the simulation.

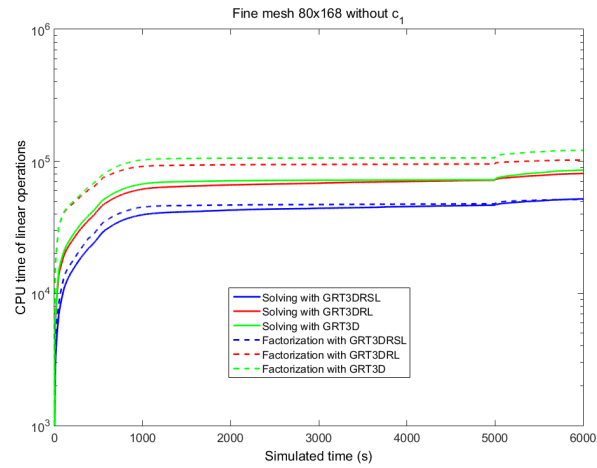FIGURE 8: Number of linear solvings and matrix factorizations during the simulation (logarithmic scale).

FIGURE 9: CPU time during the simulation of the linear system solvings and the matrix factorizations (logarithmic scale).
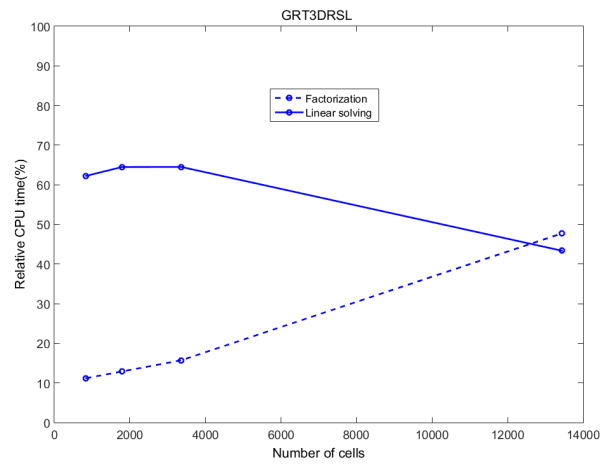
FIGURE 10: Percentage of CPU time in linear solvings and factorizations versus the mesh size using GRT3DRSL.