N° d'ordre : ANNÉE 2014

UNIVERSITÉ DE RENNES 1

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Mathématique et Applications

École doctorale Matisse

présentée par

Souhila SABIT

préparée à l'unité de recherche INRIA Rennes Bretagne Atlantique
Institut National de Recherche en Informatique et Automatique
Composante Universitaire ISTIC

Les méthodes numériques de transport réactif

Thèse soutenue à Rennes
le 27 mai 2014

devant le jury composé de :

Damien TROMEUR-DERVOUT
PROFESSEUR, UNIVERSITÉ DE LYON
/ *Rapporteur*

Carole ROSIER
PROFESSEUR, UNIVERSITÉ DU LITTORAL CÔTE D'OPALE
/ *Rapporteur*

Alexandre Ern
PROFESSEUR, ECOLE DES PONTS PARISTECH
/ *Examinateur*

Benoit Cochepin
INGÉNIEUR, ANDRA
/ *Examinateur*

Jocelyne ERHEL
DIRECTRICE DE RECHERCHE INRIA RENNES
/ *Directrice de thèse*

Édouard Canot
CHARGÉ DE RECHERCHE CNRS, IRISA RENNES
/ *Co-directeur de thèse*

Remerciements

Je tiens à remercier les responsables de l'Andra et les responsables de centre de recherche INRIA pour l'opportunité de préparer ma thèse et finir mes travaux.

Je remercie également Madame Jocelyne Erhel pour l'aide et les conseils concernant les missions évoquées dans cette thèse.

Je remercie vivement Monsieur Édouard Canot qui m'a aidé et m'a orienté pendant ces années, pour ses précieux conseils.

Je remercie Alexander Ern, Professeur à l'École des Ponts ParisTech pour avoir présidé le jury de ma soutenance.

Je tiens à remercier également Damien Tromeur-Dervout, Professeur à l'université de Lyon et Carole Rosier, Professeur à l'université du Littoral Côte d'Opale d'avoir accepté d'être les rapporteurs et Monsieur Benoit Cochepin, Ingénieur à l'Andra.

Je n'oublie pas de remercier beaucoup Nadir Soualem, Ingénieur pendant deux années au sein de notre équipe et qui m'a aidé pendant ma première année.

Je remercie toute ma famille qui m'a beaucoup soutenu.

Je remercie chaleureusement tous mes amis qui m'ont aidé à passer ces années de thèse.

Je remercie finalement toute l'équipe SAGE.

Table des matières

Remerciements **1**

Table des matières **3**

Introduction **7**

1 Modèle de chimie **11**
1.1 Réactions chimiques et espèces chimiques 11
1.1.1 Molalité d'une espèce, concentration et activité 12
1.1.2 Composants et espèces secondaires 13
1.2 Réactions aqueuses et surfaciques 14
1.2.1 Lois d'action de masse 14
1.2.2 Tableau de Morel 14
1.3 Réactions de précipitation-dissolution 15
1.3.1 Modèle simplifié de précipitation-dissolution 15
1.3.2 Modèle général de précipitation-dissolution 16
1.4 Lois de conservation de la masse 17
1.4.1 Système chimique 17
1.5 Jacobien du système chimique 18
1.5.1 Composant de type traceur 20
1.6 Variables logarithmiques 21
1.6.1 Système chimique avec logarithmes 21
1.6.2 Jacobien du modèle simplifié avec logarithmes 22
1.6.3 Composant de type traceur avec logarithmes 23
1.6.4 Composant proche de 0 avec logarithme 24
1.7 La neutralité électrique 25
1.8 Conclusion 26

2 Transport réactif **27**
2.1 Modèle de transport 27
2.1.1 Loi de Darcy 27
2.1.2 L'écoulement (équation de type diffusion) 28
2.1.3 L'advection 29
2.1.4 La dispersion 29

2.1.5 L'équation d'advection-dispersion-réaction . . . 29
2.2 Couplage entre chimie et transport . . . 29
2.3 Méthodes Numériques . . . 31
2.3.1 Discrétisation spatiale . . . 31
2.3.2 La méthode séquentielle non itérative SNIA . . . 32
2.3.3 La méthode globale avec approche DAE (GDAE) . . . 32
2.4 Réduction du système linéaire . . . 34
2.5 Réduction du système non linéaire . . . 36
2.5.1 Réduction du système différentiel avec l'élimination de C . . . 36
2.5.2 Réduction du système non linéaire avec l'élimination de T . . . 37
2.6 Conclusion . . . 38

3 Suite Logicielle GRT3D (Global Reactive Transport 3D) 39
3.1 Introduction . . . 39
3.2 Description de GRT3D version 1 . . . 40
3.2.1 Composants logiciels . . . 40
3.2.2 SIA1D (séquentielle itérative en 1D) . . . 41
3.2.3 GDAE1D (équations Algébro-différentielles) . . . 41
3.2.4 GDAE3D . . . 41
3.3 Refactoring . . . 41
3.3.1 Difficultés rencontrées . . . 41
3.3.2 Restructuration sous forme de bibliothèques . . . 42
3.3.2.1 sundials_sparse_column . . . 42
3.3.2.2 sundials_sparse_row . . . 42
3.3.2.3 MODFLOW-96 . . . 43
3.3.2.4 MT3D . . . 43
3.3.3 Modifications des points d'entrée . . . 44
3.3.3.1 SIA1D . . . 44
3.3.3.2 GDAE1D . . . 44
3.3.3.3 GDAE3D . . . 44
3.4 Description version 2 . . . 45
3.4.1 GRT3DRL . . . 45
3.4.2 GRT3DRSL . . . 46
3.4.3 SNIA . . . 47
3.4.4 GRT3DRSLC . . . 47
3.5 Tests . . . 48
3.5.1 aqueux-adv (1D) . . . 48
3.5.2 ex11-adv (1D) . . . 49
3.5.3 Benchmark-adv (1D) . . . 52
3.5.4 Benchmark-disp (1D) . . . 55
3.6 Conclusion . . . 58

4 Cas test Andra 2D **59**
4.1 Description . . . 59
4.1.1 Les conditions initiales . . . 62
4.2 La solution analytique . . . 62
4.3 Comparaison des trois versions de GRT3D . . . 63
4.3.1 Précision . . . 63
4.3.2 Temps de calcul . . . 65
4.4 La solution . . . 68
4.5 Comparaison entre GRT3DRSL et SNIA . . . 71
4.6 Conclusion . . . 73

5 Cas test Benchmark 2D **75**
5.1 Description . . . 75
5.1.1 Les caractéristiques physiques . . . 76
5.1.2 Le système chimique du cas test Benchmark . . . 77
5.1.3 Les conditions initiales et les conditions aux limites . . . 78
5.2 Expériences numériques . . . 78
5.3 Comparaison des trois versions de GRT3D . . . 79
5.4 Résultats . . . 81
5.5 Analyse du temps de calcul . . . 85
5.5.1 Temps total . . . 85
5.5.2 Temps de calcul cumulé . . . 86
5.6 Analyse du conditionnement de la Jacobienne . . . 87
5.7 Analyse des sorties de Sundials . . . 90
5.7.1 Nombre de pas de temps internes . . . 90
5.7.2 Nombre d'échecs du test d'erreur . . . 92
5.7.3 Nombre d'échecs de la Jacobienne . . . 94
5.8 Analyse des étapes linéaires . . . 95
5.8.1 Nombre de résolutions et de factorisations . . . 95
5.8.2 Temps de calcul des résolutions et factorisations . . . 97
5.8.3 Pourcentages des temps de calcul des étapes linéaires . . . 99
5.9 Conclusion . . . 100

Annexes **105**
A Installation de GRT3D . . . 105
A.1 Contrôle de version . . . 105
A.2 Variables d'environnement . . . 105
A.3 Sundials-2.3.0 [1] et Suitesparse . . . 105
A.4 Sundials_sparse-2.3.0 et Sundials_sparse_reduction-2.3.0 . . . 105
A.5 PGPLOT . . . 106
A.6 SIA1D, GDAE1D, GDAE3D, GRT3DRL, GRT3DRSL 106
A.7 MODFLOW-96 et MT3D-5.2.0 . . . 106
B Les fichiers d'entrée . . . 107
B.1 Fichiers d'entrée de GRT3D . . . 107

B.2 Fichiers d'entrée de MODFLOW 109
B.3 Fichiers d'entrée de MT3D 113
C Exécution (en ligne de commande) 118
C.1 Exécution de MODFLOW et MT3D 118
C.2 Exécution de SIA1D 118
C.3 Exécution de GDAE1D 119
C.4 Exécution de GDAE3D 119
D Fichiers de sortie 120

Bibliographie **128**

Table des figures **129**

Introduction

L'eau est un élément très important dans la vie des humains, des animaux et des plantes. Seulement une petite partie est utilisée comme eau potable et cette partie est menacée par diverses raisons : la pollution domestique, les rejets industriels, l'utilisation de pesticides et d'engrais dans l'agriculture, les pluies acides ...
On doit donc veiller à ce que cette eau potable ne soit pas polluée.
En particulier, la pollution par des éléments radioactifs doit être absolument évitée car elle contaminerait toutes les chaînes alimentaires. Compte tenu de la dangerosité de ces déchets radioactifs, il est nécessaire de les confiner en lieu sûr. Le niveau de radioactivité ne permet pas la décharge sans contrôle dans l'environnement.
Les déchets radioactifs sont divers et leurs caractéristiques diffèrent d'un déchet à l'autre : nature physique et chimique, niveau et type de radioactivité, durée de vie ...
En France, environ 2 kg de déchets radioactifs sont produits par an et par habitant[2].
En comparaison :

- Les déchets industriels représentent 2.500 kg par an et par habitant.
- Les déchets ménagers 360 kg par an et par habitant.

Le stockage des déchets radioactifs en France est lié à deux critères importants :

1. Le niveau de radioactivité : le niveau de radioactivité correspond à la quantité de rayonnement émis par les éléments radioactifs (radionucléides ou radioisotopes) contenus dans les déchets. Il y a 4 niveaux d'activités différents : haute activité (HA), moyenne activité (MA), faible activité (FA) et très faible activité (TFA).
2. La période radioactive : elle quantifie le temps au bout duquel l'activité initiale d'un radionucléide est divisée par deux, il existe une période courte (inférieure ou égale à 31 ans) et une période longue (supérieure à 31 ans). Il y a aussi le cas particulier des radionucléides utilisés pour les besoins de diagnostic en médecine, période très courte (inférieure à 100 jours).

Classification des déchets radioactifs français :

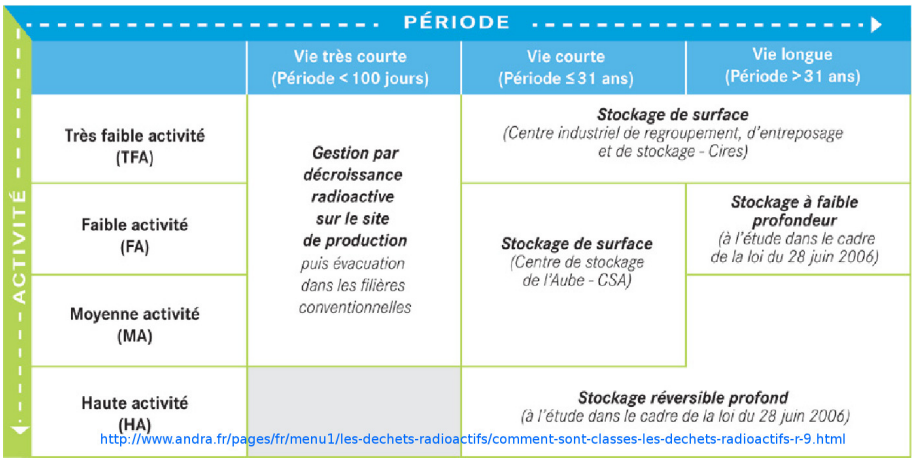


FIGURE 1 – Classification des déchets radioactifs français en fonction de leur mode de gestion

À partir du tableau dans la Figure 1, on peut classer les déchets en cinq catégories ; avec des modes de gestion différents :

- Les déchets de haute activité (HA) et de moyenne activité à vie longue (MA-VL) : ce sont les déchets provenant du cœur du réacteur nucléaire à haut rayonnement. Un stockage géologique profond est étudié.
- Les déchets de faible activité à vie longue (FA-VL). Un stockage à faible profondeur est à l'étude.
- Les déchets de faible et moyenne activité à vie courte (FMA-VC) : ce sont les déchets technologiques (gants, filtres, vêtements de protection, outillages, etc.). Un stockage en surface est opérationnel et un autre est fermé et surveillé.
- Les déchets à vie courte ou longue de très faible activité (TFA) : ce sont les déchets qui ont été contaminés provenant du démantèlement des matières dans les installations nucléaires : granulats, béton, ferraille ...
  Un stockage en surface est opérationnel.
- Les déchets à vie courte. Ils sont gérés sur le site de production.

Les études liées aux déchets de haute activité HA et de moyenne activité à vie longue MA-VL sont effectuées par l'Andra (Agence Nationale pour la gestion des Déchets RAdioactifs).
L'objectif du stockage profond est de confiner les déchets à la fois par des ouvrages de stockage et par une barrière géologique naturelle. L'Andra est chargée de concevoir et d'implanter un centre de stockage pour les déchets HA et MA-VL, appelé Cigéo (Centre Industriel de stockage Géologique).
Une partie des études menées par l'Andra concerne la modélisation mathématique et la simulation numérique parce que l'échelles de temps inaccessibles à l'expérimentation

et la modélisation participe au dimensionnement du stockage. Il s'agit de comprendre, décrire et prédire la dynamique des phénomènes dans les sites de stockage. Plusieurs phénomènes couplés sont modélisés : Thermique, Hydraulique, Mécanique, Chimique (THMC).
Dans cette thèse, nous nous intéressons aux méthodes numériques pour simuler l'évaluation Hydraulique et Chimique.

La modélisation du transport réactif du contaminants en milieu poreux est un problème complexe cumulant les difficultés de la modélisation du transport avec celles de la modélisation de la chimie et surtout du couplage entre les deux. Cette modélisation conduit à un système d'équations aux dérivées partielles et algébriques dont les inconnues sont les quantités d'espèces chimiques. Plusieurs méthodes ont été développées pour la résolution de ce type de problème [14, 23] : Méthodes séquentielles non itératives (SNIA), Méthodes séquentielles itératives (SIA), approches globales d'équations différentielles ordinaires (ODE) [7] et approches globales de substitution direct (DSA)[27]...

Nous avons choisi de résoudre ce problème par une approche globale GDAE [12] : l'utilisation d'une méthode de lignes[39], correspondant à la discrétisation en espace seulement, conduit à un système différentiel algébrique (DAE) qui est résolu par un solveur. Ce solveur repose sur une méthode implicite, à ordre et pas variables, qui requiert à chaque pas de temps la résolution d'un grand système non linéaire associé à une matrice jacobienne. Cette méthode est implémentée dans un logiciel qui s'appelle GRT3D (Transport Réactif Global en 3D) [12].

Le travail de cette thèse consiste à améliorer l'approche GDAE, du point de vue de la performance, de la stabilité et de la robustesse. En ce qui concerne la performance, la version d'origine de GRT3D souffrait d'une certaine lenteur, à tel point que certaines simulations en haute résolution demandaient plusieurs jours de calculs ; une réduction analytique opérée sur le système linéaire à résoudre liée à la jacobienne du système a permis une grande amélioration, comme le montre les résultats décrits dans la thèse ; l'élimination du traceur, dans une moindre mesure, a contribué elle aussi à réduire le temps de calcul. En ce qui concerne la stabilité et la robustesse de l'approche GDAE, on a montré que l'utilisation du logarithme des concentrations, assez fréquente dans des logiciels similaires et destinée principalement à éviter des valeurs négatives, conduisait dans certains cas à des matrices jacobiennes singulières ; la suppression de ces logarithmes, donc un retour aux valeurs de concentrations brutes, a permis non seulement d'éliminer ces singularités mais aussi de contribuer partiellement à un accroissement de la performance du code.

De plus, nous avons aussi enrichi les possibilités de GRT3D, par la prise en compte complète des équations de précipitation-dissolution permettant ainsi l'apparition ou la disparition d'une espèce précipitée, chose qui n'était pas possible avec la version d'origine du logiciel. Cette prise est compte est modélisée par un problème de complémentarité [35, 31], résolu par une méthode de Newton semismooth [22].

Plan de la thèse

Ce document est organisé en cinq parties :
Le premier chapitre présente les équations générales des réactions chimiques en milieu poreux ; les notions de composants et d'espèces secondaires y sont également expliquées. Ce chapitre contient également l'analyse et la comparaison de la jacobienne du système, suivant qu'on utilise les logarithmes ou pas.
Le deuxième chapitre présente le système de transport qui est basé sur la loi de Darcy, l'advection et la dispersion. Ensuite, on présente les deux méthodes numériques utilisées pendant ce travail :

- L'approche séquentielle non itérative SNIA est une méthode basée sur le schéma d'Euler explicite. À chaque pas de temps, on résout explicitement l'équation de transport et on utilise ces calculs comme données pour le système chimique, résolu dans chaque maille de façon indépendante.
- L'approche globale (GDAE) est basée sur un schéma en temps implicite BDF (Backward Differentiation Formulas), qui conduit à un système non linéaire couplé à chaque pas de temps, résolu globalement par une méthode de Newton.

Puis, on définit le système non linéaire réduit à chaque itération de Newton.
Dans le troisième chapitre, on décrit le logiciel GRT3D et les modifications apportées. On définit en particulier les entrées, les sorties et la manière d'exécuter les différentes versions. Quelques résultats numériques pour des cas en 1D terminent le chapitre.

Le chapitre 4 présente un cas test Andra en 2D et les résultats obtenus avec les versions de logiciel GRT3D. Nous montrons la réduction du temps d'exécution, étudions l'utilisation de la mémoire et la précision.
On présente les réaction chimiques de ce cas, les espèces secondaires aqueuses et fixées, les précipités et les coefficients stœchiométriques avec les concentrations initiales et les caractéristiques de transport (l'advection, dispersion, porosité, conditions aux limites). Ensuite, on donne la solution analytique pour un composant et on compare cette solution avec nos résultats pour plusieurs maillages.
On présente les mêmes critères pour un cas test Benchmark en 2D dans le chapitre 5. Dans ce cas on n'a pas la solution analytique donc on compare les résultats à ceux obtenus avec le logiciel GRT3DRSL pour le même maillage. On analyse le conditionnement de la matrice jacobienne et on compare les sorties du sundials pour chacune des trois versions GRT3D, GRT3DRL (Réduction et avec Logarithme) et GRT3DRSL (Réduction et Sans Logarithme) .
Finalement, on conclut ce travail et on donne des perspectives pour le futur travail.

Chapitre 1

Modèle de chimie

Dans ce chapitre, nous présentons un modèle de réactions chimiques dans un milieu poreux proposé par plusieurs chercheurs [12, 7, 27]. Nous rappelons les réactions chimiques, les notations utilisées dans cette thèse, les lois d'action de masse et de conservation de masse, ainsi que la loi de conservation de la charge électrique.
D'autre part, on définit le modèle généralisé du problème de précipitation-dissolution avec un problème de complémentarité qui permet l'apparition ou la disparition d'une espèce précipitée.
Ce point particulier est une nouveauté pour le logiciel GRT3D [12]. Pour le traiter numériquement, nous avons choisi d'utiliser la méthode de Newton semismooth.
Une analyse détaillée de la matrice jacobienne, écrite de manière analytique aussi bien sous la forme primitive (sans logarithme) que sous la forme utilisée par plusieurs codes (avec logarithme), permet de montrer que cette matrice est parfois singulière dans le cas avec logarithme.

1.1 Réactions chimiques et espèces chimiques

Au cours du temps, les espèces peuvent être modifiées entre elles, on peut exprimer ces modifications sous forme de réactions chimiques qui sont écrites sous la forme suivante :

$$\sum_{i=1}^{N_E} \nu_{r,i} E_i \rightleftarrows 0,\ r = 1, ...N_r \tag{1.1}$$

avec $\nu_{r,i}$ le coefficient stœchiométrique de l'espèce E_i pendant la réaction r. Ce système contient N_E espèces chimiques provoquant N_r réactions.
La vitesse d'une réaction se définit comme étant la variation de la concentration d'un réactif ou d'un produit par unité de temps.
À partir de la vitesse de la réaction chimique, on définit deux grandes familles de réactions [46, 5, 36] :

1. L'approche à l'équilibre : un système est dit à l'équilibre si les variables d'état (Température, l'activité chimique, la vitesse de la réaction ...) ont même valeur en tout point du système et de façon constante dans le temps.

2. L'approche cinétique : la vitesse de la réaction est variée au cours du temps.

Deux modèles expriment les réactions chimiques à l'équilibre ou les réactions chimiques cinétiques[5, 45] :

— Le modèle généré par l'utilisation des lois d'action de masse, qui est très utilisé [23, 14, 37, 17, 18] ...
— Le modèle basé sur la minimisation de l'enthalpie libre [43, 24, 15].

1.1.1 Molalité d'une espèce, concentration et activité

La molalité d'un espèce

La molalité d'un soluté est la quantité de soluté (en moles) dans une solution divisée par la masse du solvant (en kilogramme) :

$$m_i = \frac{\text{quantité de soluté (mole)}}{\text{masse de solvant (kg)}}$$

Les concentrations

La concentration molaire ou la molarité est le nombre de moles de soluté par litre de solution

$$c_i = \frac{\text{quantité de soluté (mole)}}{\text{volume de la solution (L)}}$$

Puisque les concentrations sont des valeurs petites et que le solvant est l'eau (la masse d'un 1 litre d'eau vaut presque 1 kg), alors numériquement la concentration d'une espèce en solution est égale à sa molalité, mais ces deux grandeurs n'ont pas la mêmes dimensions.

L'activité

L'activité chimique d'une espèce chimique correspond à la concentration active de cette espèce et elle dépend du type de cette espèce.

L'activité d'une espèce aqueuse

L'activité d'une espèce aqueuse est liée à sa molalité m_i

$$a_i = \gamma_i * \frac{m_i}{m_i^0}$$

où γ_i est le coefficient d'activité (sans dimension) et m_i^0 est la molalité de référence (égale à 1 mol/kg H_2O). Si Les concentrations sont petites, on peut supposer que le coefficient d'activité γ_i vaut 1. Pour les milieux plus concentrés, on peut approcher γ_i par la loi de Debye-Hückel [6] :

$$\log \gamma_i = -\frac{A\ z_i^2 \sqrt{I}}{1 + b_i B \sqrt{I}}$$

où z_i est la charge électrique, A et B sont des coefficients dépendant de la température et b_i est la taille de l'ion. I est la force ionique définie par :

$$I = -\frac{1}{2}\sum m_i \; z_i^2$$

Il y a aussi l'équation de Davies [10], le modèle de B-Dot [33] et d'autres approches [44, 34].

L'activité d'une espèce fixée

Pour les espèces fixées, il n'y a pas un modèle uniforme pour calculer l'activité et différents modèles peuvent être utilisés [5].

L'activité d'une espèce précipitée

L'activité est toujours égale à 1.

L'activité de l'eau

L'activité de l'eau vaut presque 1 pour les solutions diluées et dans les autres cas, elle est approchée par exemple par la loi de Raoult

$$a_{H_2O} = 1 - 0.017 \sum_{n_i}^{N_{aq}} \frac{n_i}{W_{aq}}$$

où W_{aq} est la masse de solvant dans la solution aqueuse, N_{aq} est le nombre d'espèce aqueuse dans le système.

1.1.2 Composants et espèces secondaires

Dans le système chimique, nous avons un grand nombre d'inconnues. Pour cela, nous utilisons certains critères et des hypothèses pour réduire la taille du système.
On suppose que les réactions chimiques sont indépendantes. Même dans le cas contraire, on peut réduire le système jusqu'à trouver l'indépendance de ces réactions. Cette hypothèse et les lois d'actions de masse permettent d'exprimer certaines inconnues, que l'on nomme espèces secondaires, en fonction des autres inconnues, que l'on nomme composants.
À partir de l'activité de chaque espèce, on peut choisir les espèces aqueuses, les espèces fixées et les espèces précipitées. Puisque l'activité des espèces précipitées ne dépend pas du nombre de moles du précipité, on ne peut pas l'exprimer en fonction d'autres inconnues. Donc, l'espèce précipitée reste inconnue dans le système réduit.
On note respectivement c, s et p les concentrations des composants aqueux, fixés et des espèces précipitées. Par ailleurs on note respectivement α et β les concentrations des espèces secondaires aqueuses et fixées.

1.2 Réactions aqueuses et surfaciques

Nous avons étudié les réactions à l'équilibre et utilisé les lois d'action de masse, donc on peut traduire cette hypothèse par le système suivant :

$$\prod_{i=1}^{N_E} a_i^{\nu_{r,i}} = K_r,\ r = 1, N_r \tag{1.2}$$

où a_i est l'activité de l'espèce E_i, $\nu_{r,i}$ sont les coefficients stœchiométriques et K_r est la constante d'équilibre de la réaction r.
Dans notre étude, les concentrations ont des valeurs petites et le solvant est l'eau. Donc, nous avons $a_i = c_i$.

1.2.1 Lois d'action de masse

On suppose que le système contient N_c composants aqueux, N_s composants fixés, N_p espèces précipitées, N_α réactions aqueuses et N_β réactions surfaciques.
Les lois d'action de masse et les notations précédentes permettent de définir le système chimique suivant :

$$\begin{cases} \alpha_i(c) = K_{c,i} \prod_{j=1}^{N_c} c_j^{S_{ij}}, & i = 1, N_\alpha \\ \beta_i(c,s) = K_{s,i} \prod_{j=1}^{N_c} c_j^{A_{ij}} \prod_{j=1}^{N_s} s_j^{B_{ij}}, & i = 1, N_\beta \end{cases} \tag{1.3}$$

où $S \in \mathbf{Z}^{N_\alpha \times N_c}$, $A \in \mathbf{Z}^{N_\beta \times N_c}$ et $B \in \mathbf{Z}^{N_\beta \times N_s}$ sont les matrices des coefficients stœchiométriques. $K_c \in \mathbf{R}^{N_\alpha}$ et $K_s \in \mathbf{R}^{N_\beta}$ sont les constantes des réactions.

1.2.2 Tableau de Morel

Avec les définitions et les notations précédentes, on définit le tableau de Morel [28] suivant, de façon condensée (et matricielle), qui exprime chaque espèce α_i (ou β_i) en fonction des c_j et s_j :

composant / espèce secondaire	c	s
α	S	0
β	A	B

TABLE 1.1 – Tableau de Morel

1.3 Réactions de précipitation-dissolution

En général, dans un système réactif avec des réactions de précipitation-dissolution à l'équilibre, on c'est pas si les minéraux sont présents ou non dans le système. La présence d'un minéral signifie que le fluide est saturé à l'équilibre, et si le fluide est sous-saturé, le minéral correspondant est complètement dissous.
Dans le cas où tous les minéraux sont présents, l'équilibre peut être simplement exprimé comme un système non linéaire d'équations algébriques qui est résolu par une méthode de Newton standard. Mais en pratique, cette connaissance n'est pas disponible, donc on a plusieurs possibilités :

1. Le fluide est saturé et le minéral est présent, c'est-à-dire

$$\begin{cases} 0 &= (1 - K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}) \quad i = 1, N_p \\ \\ 0 &< \qquad p_i \end{cases} \tag{1.4}$$

   où E_{ij} sont les coefficients stœchiométriques des réactions de précipitation.

2. Le fluide est sous-saturé et le minéral est entièrement dissous, c'est-à-dire

$$\begin{cases} 0 &\leq (1 - K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}) \quad i = 1, N_p \\ \\ 0 &= \qquad p_i \end{cases} \tag{1.5}$$

En général la formule qui regroupe les deux cas précédents est la suivante :

$$\begin{cases} 0 \leq (1 - K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}) \quad \text{et} \quad p_i = 0, \quad i = 1, N_p \\ \\ \text{ou} \\ \\ 0 = (1 - K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}) \quad \text{et} \quad 0 \leq p_i, \quad i = 1, N_p \end{cases} \tag{1.6}$$

1.3.1 Modèle simplifié de précipitation-dissolution

Dans un premier travail, on va étudier le cas où le minéral est présent et le fluide est saturé par rapport à la réaction de précipitation-dissolution correspondante [12]. c'est-à-dire :

$$\begin{cases} 1 &= K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}, \quad i = 1, N_p \\ \\ 0 &< p_i, \qquad\qquad i = 1, N_p \end{cases} \tag{1.7}$$

1.3.2 Modèle général de précipitation-dissolution

Dans un cas général, on peut écrire le système (1.6) sous la forme suivante :

$$\begin{cases} 0 = (1 - K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}) \times p_i, & i = 1, N_p \\ \text{avec} \\ 0 \leq (1 - K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}) \quad \text{et} \quad 0 \leq p_i, & i = 1, N_p \end{cases} \tag{1.8}$$

Ce problème est appelé un problème de complémentarité (PC)[41, 23].
On va appliquer une méthode qui élimine les inégalités dans (1.8). Pour cela, on va trouver une fonction φ qui vérifie :

$$\varphi(a, b) = 0 \iff ab = 0 \text{ et } a \geq 0 \text{ et } b \geq 0 \tag{1.9}$$

Cette fonction s'appelle fonction de la complémentarité ou C-fonction.
Il y a plusieurs types de fonction de complémentarité comme :

1. La fonction minimum $\varphi_{min}(a, b) = min(a, b)$ [19, 31].
2. La fonction Fischer-Burmeister $\varphi_{FB}(a, b) = \sqrt{a^2 + b^2} - a - b$ [13].
3. Fonction différentielle par morceaux [42].

On peut remplacer l'équation (1.8) par $\varphi(\Pi_i(c), p_i) = 0$ avec

$$\Pi_i(c) = 1 - K_{p,i} \prod_{j=1}^{N_c} c_j^{E_{ij}}$$

On résout le problème d'équilibre chimique par une méthode de Newton. Il faut remarquer que le gradient de φ n'est pas défini de façon unique partout. La fonction φ fait partie de la classe des fonctions semismooth.
La méthode de Newton semismooth est l'application de la méthode de Newton à une fonction semismooth, où à chaque étape de Newton, un élément quelconque du B-sous-différentiel est utilisé à la place de la jacobienne classique.
Dans notre étude, on va utiliser la fonction φ_{min}, alors pour $i = 1, N_p$, on a :

$$\varphi_{min}(\Pi_i(c), p_i) = \begin{cases} \Pi_i(c) & \text{si} \quad \Pi_i(c) \leq p_i \\ p_i & \text{si} \quad \Pi_i(c) > p_i \end{cases} \tag{1.10}$$

Alors la jacobienne est définie de cette manière :

$$\frac{\partial \varphi_{min}(\Pi_i(c), p_i)}{\partial c_j} = \begin{cases} \dfrac{\partial \Pi_i(c)}{\partial c_j} & \text{si} \quad \Pi_i(c) \leq p_i \\ 0 & \text{si} \quad \Pi_i(c) > p_i \end{cases} \tag{1.11}$$

Et

$$\frac{\partial \varphi_{min}(\Pi_i(c), p_i)}{\partial p_i} = \begin{cases} 0 & \text{si} \quad \Pi_i(c) \leq p_i \\ \\ 1 & \text{si} \quad \Pi_i(c) > p_i \end{cases} \tag{1.12}$$

où la dérivée de Π_i par rapport à c_j est calculée dans l'équation suivante :

$$\frac{\partial \Pi_i}{\partial c_j} = - \begin{pmatrix} K_{p,j} \prod_{l=1 \ \& \ l \neq j}^{N_c} c_l^{E_{il}} & \text{Si } c_j = 0 \text{ et } E_{ij} = 1 \\ 0 & \text{Si } c_j = 0 \text{ et } E_{ij} > 1 \text{ ou } E_{ij} = 0 \\ \infty & \text{Si } c_j = 0 \text{ et } E_{ij} < 0 \\ \frac{E_{ij}}{c_j} K_{p,j} \prod_{l=1}^{N_c} c_l^{E_{il}} & \text{Si } c_j \neq 0 \end{pmatrix} \tag{1.13}$$

1.4 Lois de conservation de la masse

Puisque les réactions sont à l'équilibre, on peut définir les concentrations totales pour chaque composant [9, 37, 29]. Ces concentrations totales se décomposent en total aqueux T_c, et total fixé T_s et sont exprimées de façon suivante :

$$\begin{cases} T_c(c,s,p) &= c + S^T \alpha(c) + A^T \beta(c,s) + E^T p \\ T_s(c,s) &= s + B^T \beta(c,s) \end{cases} \tag{1.14}$$

où $\alpha(c)$ et $\beta(c,s)$ sont définis dans l'équation (1.3).
Les lois de conservation de masse sont alors écrites sous la forme suivante :

$$\begin{cases} T &= T_c(c,s,p) \\ W &= T_s(c,s) \end{cases} \tag{1.15}$$

où T et W sont des données du modèle chimique, dans le cas d'un système fermé. Par contre, dans le cas d'un système ouvert, T et W sont des variables obtenues par couplage avec un autre modèle.

1.4.1 Système chimique

Le système chimique est un système non linéaire de $(N_c+N_s+N_p)$ équations avec des contraintes. Pour le modèle simplifié, on suppose que les concentrations des minéraux sont strictement positives ($p_i > 0$), ce qui conduit au modèle chimique suivant :

$$\begin{cases} T - T_c(c,s,p) &= 0 \\ W - T_s(c,s) &= 0 \\ -\Pi(c) &= 0 \\ c &\geq 0 \\ s &\geq 0 \\ p &> 0 \end{cases} \tag{1.16}$$

Nous utilisons la méthode de Newton pour résoudre ce problème non linéaire, en supposant que le système a une solution unique et que la Jacobienne n'est pas singulière. La méthode de Newton est adaptée pour satisfaire les contraintes. Pour le modèle généralisé, le modèle chimique est le suivant :

$$\begin{cases} T - T_c(c,s,p) &= 0 \\ W - T_s(c,s) &= 0 \\ \Pi(c) \times p &= 0 \\ c &\geq 0 \\ s &\geq 0 \\ p &\geq 0 \\ \Pi(c) &\geq 0 \end{cases} \tag{1.17}$$

Nous utilisons la méthode de Newton semismooth avec la fonction de complémentarité min. Donc le système (1.17) s'écrit sous la forme suivante :

$$\begin{cases} T - T_c(c,s,p) &= 0 \\ W - T_s(c,s) &= 0 \\ \varphi_{min}(\Pi(c),p) &= 0 \\ c &\geq 0 \\ s &\geq 0 \\ p &\geq 0 \\ \Pi(c) &\geq 0 \end{cases} \tag{1.18}$$

La méthode de Newton est adaptée pour satisfaire les contraintes.

1.5 Jacobien du système chimique

Dans cette partie, nous écrivons de façon détaillée la matrice jacobienne du modèle généralisé. Nous observons que le calcul des dérivées des espèces secondaires est assez compliqué. Ensuite, nous analysons le cas particulier d'un traceur et montrons que la ligne associée dans la Jacobienne est celle de l'identité.
La Jacobienne du système (1.18) s'écrit sous la forme suivante :

$$J(c,s,p) = \begin{pmatrix} -\dfrac{\partial T_c}{\partial c} & -\dfrac{\partial T_c}{\partial s} & -\dfrac{\partial T_c}{\partial p} \\ -\dfrac{\partial T_s}{\partial c} & -\dfrac{\partial T_s}{\partial s} & 0 \\ \dfrac{\partial \varphi_{min}}{\partial c} & 0 & \dfrac{\partial \varphi_{min}}{\partial p} \end{pmatrix}$$

Les dérivées de φ_{min} sont définies dans la section (1.3.2) et les dérivées des concentrations totales sont calculées à partir de (1.14) et des dérivées des espèces secondaires α

et β, donc

$$J(c,s,p) = \begin{pmatrix} -I - S^T\dfrac{\partial \alpha}{\partial c} - A^T\dfrac{\partial \beta}{\partial c} & -A^T\dfrac{\partial \beta}{\partial s} & -E^T \\ -B^T\dfrac{\partial \beta}{\partial c} & -I - B^T\dfrac{\partial \beta}{\partial s} & 0 \\ \dfrac{\partial \varphi_{min}}{\partial c} & 0 & \dfrac{\partial \varphi_{min}}{\partial p} \end{pmatrix} \tag{1.19}$$

Les coefficients de la matrice Jacobienne peuvent s'écrire sous la forme suivante :

$$\begin{cases} \dfrac{\partial T_{c,i}}{\partial c_j} = \delta_{ij} + \displaystyle\sum_{k=1}^{N_\alpha} S_{ki}\frac{\partial \alpha_k}{\partial c_j} + \sum_{k=1}^{N_\beta} A_{ki}\frac{\partial \beta_k}{\partial c_j}, & i = 1,...,N_c, \quad j = 1,...,N_c \\ \dfrac{\partial T_{c,i}}{\partial s_j} = \displaystyle\sum_{k=1}^{N_\beta} A_{ki}\frac{\partial \beta_k}{\partial s_j}, & i = 1,...,N_c, \quad j = 1,...,N_s \\ \dfrac{\partial T_{c,i}}{\partial p_j} = E_{ji}, & i = 1,...,N_c, \quad j = 1,...,N_p \\ \dfrac{\partial T_{s,i}}{\partial c_j} = \displaystyle\sum_{k=1}^{N_\beta} B_{ki}\frac{\partial \beta_k}{\partial c_j}, & i = 1,...,N_s, \quad j = 1,...,N_c \\ \dfrac{\partial T_{s,i}}{\partial s_j} = \delta_{ij} + \displaystyle\sum_{k=1}^{N_\beta} B_{ki}\frac{\partial \beta_k}{\partial s_j}, & i = 1,...,N_s, \quad j = 1,...,N_s \end{cases} \tag{1.20}$$

Les dérivées des concentrations des espèces secondaires s'écrivent :

$$\frac{\partial \alpha_k}{\partial c_j} = \begin{cases} K_{c,k}\displaystyle\prod_{l=1\ \&\ l\neq j}^{N_c} c_l^{S_{kl}} & \text{Si } c_j = 0 \text{ et } S_{kj} = 1 \\ 0 & \text{Si } c_j = 0 \text{ et } S_{kj} > 1 \text{ ou } S_{kj} = 0 \\ \infty & \text{Si } c_j = 0 \text{ et } S_{kj} < 0 \\ S_{kj}\dfrac{\alpha_k}{c_j} & \text{Si } c_j \neq 0 \end{cases} \tag{1.21}$$

et

$$\frac{\partial \beta_k}{\partial c_j} = \begin{cases} K_{s,k}\displaystyle\prod_{l=1\&\ l\neq j}^{N_c} c_l^{A_{kl}} \prod_{l=1}^{N_s} s_l^{B_{kl}} & \text{Si } c_j = 0 \text{ et } A_{kj} = 1 \\ 0 & \text{Si } c_j = 0 \text{ et } A_{kj} > 1 \text{ ou } A_{kj} = 0 \\ \infty & \text{Si } c_j = 0 \text{ et } A_{kj} < 0 \\ A_{kj}\dfrac{\beta_k}{c_j} & \text{Si } c_j \neq 0 \end{cases} \tag{1.22}$$

$$\frac{\partial \beta_k}{\partial s_j} = \left(\begin{array}{ll} K_{s,k} \prod_{l=1}^{N_c} c_l^{A_{kl}} \prod_{l=1 \ \& \ l \neq j}^{N_s} s_l^{B_{kl}} & \text{Si } s_j = 0 \text{ et } B_{kj} = 1 \\ 0 & \text{Si } s_j = 0 \text{ et } B_{kj} > 1 \text{ ou } B_{kj} = 0 \\ \infty & \text{Si } s_j = 0 \text{ et } B_{kj} < 0 \\ B_{kj} \dfrac{\beta_k}{s_j} & \text{Si } s_j \neq 0 \end{array} \right) \tag{1.23}$$

Ces calculs sont assez compliqués et impliquent des tests. En pratique, le test $c = 0$ est remplacé par $c \leq \varepsilon$, où ε est une quantité fixée proche de la précision machine.

1.5.1 Composant de type traceur

Une espèce chimique est un traceur s'il ne réagit pas avec les autres espèces i.e

1. Si l'espèce chimique est un composant aqueux c_i, alors

$$\left\{ \begin{array}{rcll} S_{ki} & = & 0, & k = 1, ..., N_\alpha \\ A_{ki} & = & 0, & k = 1, ..., N_\beta \\ E_{ji} & = & 0, & j = 1, ..., N_p \end{array} \right. \tag{1.24}$$

2. Si l'espèce chimique est un composant fixé s_i, alors :

$$B_{ki} = 0, \quad k = 1, ..., N_\beta \tag{1.25}$$

Nous montrons que la présence d'un traceur ne perturbe pas la qualité du Jacobien.

Proposition 1.1 *Si le traceur est un composant aqueux, les dérivées associée à $T_{c,i}$ sont définies de la façon suivante :*

$$\left\{ \begin{array}{rcl} \dfrac{\partial T_{c,i}}{\partial c_j} & = & \delta_{ij} \\ \dfrac{\partial T_{c,i}}{\partial s_j} & = & 0 \\ \dfrac{\partial T_{c,i}}{\partial p_j} & = & 0 \end{array} \right. \tag{1.26}$$

Ainsi, la ligne associée à $T_{c,i}$ est égale à la ligne correspondante de la matrice identité. Si le traceur est un composant fixé, les dérivées associée à $T_{s,i}$ sont définies de la façon suivante :

$$\left\{ \begin{array}{rcl} \dfrac{\partial T_{s,i}}{\partial c_j} & = & 0 \\ \dfrac{\partial T_{s,i}}{\partial s_j} & = & \delta_{ij} \end{array} \right. \tag{1.27}$$

Ainsi, la ligne associée à $T_{s,i}$ est égale à la ligne correspondante de la matrice identité.

Preuve 1.1 *Nous appliquons la définition d'un traceur dans l'expression (1.24 et 1.25) et les dérivées des concentrations totales dans l'expression(1.20).*

Remarque 1.1 *Puisque le traceur ne réagit pas avec les autres espèces, on peut l'enlever du système chimique. On a tout simplement* $c_i = T_{c,i}$ *ou* $s_i = T_{s,i}$.

1.6 Variables logarithmiques

En supposant que $c > 0$ et $s > 0$, on peut faire le changement de variable suivant : $lc = \log(c)$ et $ls = \log(s)$.
Cette forme logarithmique est utilisée dans plusieurs logiciels comme SPECY [9], MIN3P [26, 25], RETRASO [29], RICHY [20, 23, 21], parce que cette forme assure la positivité des concentrations. De plus, cette forme facilite les calculs des dérivées, parce qu'elle linéarise les équations : au lieu de calculer la dérivée des produits et des puissances, on calcule la dérivée des sommes. Avec ce changement de variable, on peut écrire les lois d'actions de masse sous la forme logarithmique suivante :

$$\left\{\begin{array}{rcl} \alpha(lc) & = & \exp[\log K_c \ + \ S^T \ lc] \\ \beta(lc, ls) & = & \exp[\log K_s \ + \ A^T \ lc + \ B^T \ ls] \end{array}\right. \tag{1.28}$$

Et la loi de conservation de masse sous la forme suivante :

$$\left\{\begin{array}{rcl} T_c(lc, ls, p) & = & \exp(lc) + S^T\alpha(lc) + A^T\beta(lc, ls) + E^T p \\ T_s(lc, ls) & = & \exp(ls) + B^T\beta(lc, ls) \end{array}\right. \tag{1.29}$$

1.6.1 Système chimique avec logarithmes

Pour le modèle simplifié l'équation de précipitation-dissolution s'écrit sous la forme suivante :

$$\left\{\begin{array}{rcll} 0 & = & \log K_p + E^T \ lc & \\ & & & \\ 0 & < & p_i, & i = 1, N_p \end{array}\right. \tag{1.30}$$

Soit $\varphi(lc) = \log K_p + E^T \ lc$. Donc le système chimique s'écrit :

$$\left\{\begin{array}{rcl} T - \exp(lc) + S^T\alpha(lc) + A^T\beta(lc, ls) + E^T p & = & 0 \\ W - \exp(ls) + B^T\beta(lc, ls) & = & 0 \\ \varphi(lc) & = & 0 \\ p > 0, \ c = \exp(lc), \ s = \exp(ls) & & \end{array}\right. \tag{1.31}$$

Ce modèle satisfait par hypothèse les contraintes sur c et s mais interdit que les concentrations soient nulles.

1.6.2 Jacobien du modèle simplifié avec logarithmes

Dans cette section, on montre la facilité de calcul de la matrice Jacobienne mais dans la section suivante, on démontre la singularité de cette matrice quand les concentrations sont proches de 0.
La matrice jacobienne est calculée de façon analogue à la section (1.5)

$$J_l(lc, ls, p) = \begin{pmatrix} -\dfrac{\partial T_c}{\partial lc} & -\dfrac{\partial T_c}{\partial ls} & -\dfrac{\partial T_c}{\partial p} \\ -\dfrac{\partial T_s}{\partial lc} & -\dfrac{\partial T_s}{\partial ls} & 0 \\ \dfrac{\partial \varphi}{\partial lc} & 0 & 0 \end{pmatrix}$$

Les coefficients de la matrice Jacobienne peuvent s'écrire sous la forme suivante :

$$\begin{cases} \dfrac{\partial T_{c,i}}{\partial (lc_j)} &= \delta_{ij} \exp(lc_j) + \displaystyle\sum_{k=1}^{N_\alpha} S_{ki} \frac{\partial \alpha_k}{\partial (lc_j)} + \sum_{k=1}^{N_\beta} A_{ki} \frac{\partial \beta_k}{\partial (lc_j)} \\ \dfrac{\partial T_{c,i}}{\partial (ls_j)} &= \displaystyle\sum_{k=1}^{N_\beta} A_{ki} \frac{\partial \beta_k}{\partial (ls_j)} \\ \dfrac{\partial T_{c,i}}{\partial p_j} &= E_{ji} \\ \dfrac{\partial T_{s,i}}{\partial (lc_j)} &= \displaystyle\sum_{k=1}^{N_\beta} B_{ki} \frac{\partial \beta_k}{\partial (lc_j)} \\ \dfrac{\partial T_{s,i}}{\partial (ls_j)} &= \delta_{ij} \exp(ls_j) + \displaystyle\sum_{k=1}^{N_\beta} B_{ki} \frac{\partial \beta_k}{\partial (ls_j)} \end{cases} \tag{1.32}$$

Et

$$\begin{cases} \dfrac{\partial \alpha_k}{\partial (lc_j)} &= S_{kj}\ \alpha_k \\ \dfrac{\partial \beta_k}{\partial (lc_j)} &= A_{kj}\ \beta_k \\ \dfrac{\partial \beta_k}{\partial (ls_j)} &= B_{kj}\ \beta_k \\ \dfrac{\partial \varphi_i}{\partial (lc_j)} &= E_{ij} \end{cases} \tag{1.33}$$

On peut écrire les dérivées des concentrations des espèces secondaires sous la forme matricielle suivante :

$$\left\{\begin{array}{lcl} \dfrac{\partial \alpha}{\partial lc} & = & \mathrm{diag}(\alpha)\ S \\ \dfrac{\partial \beta}{\partial lc} & = & \mathrm{diag}(\beta)\ A \\ \dfrac{\partial \beta}{\partial ls} & = & \mathrm{diag}(\beta)\ B \\ \dfrac{\partial \varphi}{\partial lc} & = & E \end{array}\right. \tag{1.34}$$

La matrice Jacobienne s'écrit sous la forme suivante :

$$J_l = \left(\begin{array}{ccc} -\mathrm{diag}(\exp(lc)) - S^T\ \mathrm{diag}(\alpha)\ S - A^T\ \mathrm{diag}(\beta)\ A & -A^T\ \mathrm{diag}(\beta)\ B & -E \\ -B^T\ \mathrm{diag}(\beta)\ A & -\mathrm{diag}(\exp(ls)) - B^T\ \mathrm{diag}(\beta)\ B & 0 \\ E & 0 & 0 \end{array}\right)$$

1.6.3 Composant de type traceur avec logarithmes

Dans cette partie, nous analysons, comme précédemment, l'impact d'un traceur sur le Jacobien. Nous montrons que la matrice est proche de la singularité si la concentration du traceur est proche de 0.

Proposition 1.2 *Soit c_i un composant aqueux de type traceur. Les dérivées associées à $T_{c,i}$ sont données par l'équation suivante :*

$$\left\{\begin{array}{lcl} \dfrac{\partial T_{c,i}}{\partial (lc_j)} & = & \delta_{ij} \exp(lc_j) \\ \dfrac{\partial T_{c,i}}{\partial (ls_j)} & = & 0 \\ \dfrac{\partial T_{c,i}}{\partial p_j} & = & 0 \end{array}\right. \tag{1.35}$$

Soit s_i un composant fixé de type traceur, alors :

$$\left\{\begin{array}{lcl} \dfrac{\partial T_{s,i}}{\partial (lc_j)} & = & 0 \\ \dfrac{\partial T_{s,i}}{\partial (ls_j)} & = & \delta_{ij} \exp(ls_j) \end{array}\right. \tag{1.36}$$

Si la concentration d'un traceur est proche de 0, la matrice Jacobienne est proche de la singularité.

Preuve 1.2 Premier point : à partir de la définition de traceur dans l'équation (1.24) (respectivement (1.25)) et la structure de la dérivée associée à $T_{c,i}$ dans l'équation (1.32), nous déduisons le résultat (1.35) (respectivement (1.36)).
Deuxième point : Si la concentration est proche de 0, alors $\lim_{c_i \longrightarrow 0}(\exp(lc_i)) = 0$.

Donc $\lim_{c_i \longrightarrow 0} \dfrac{\partial T_{c,i}}{\partial lc_i} = 0$.

Dans le cas d'un composant fixé $\lim_{s_i \longrightarrow 0}(\exp(ls_i)) = 0$. Donc $\lim_{s_i \longrightarrow 0} \dfrac{\partial T_{s,i}}{\partial ls_i} = 0$.
Alors la ligne associée au traceur dans la matrice tend vers 0, et la matrice Jacobienne est proche de la singularité.

1.6.4 Composant proche de 0 avec logarithme

Nous avons ainsi montré un comportement très différent des modèles avec et sans logarithmes. Dans le cas d'un traceur, grâce à la remarque (1.1), il est facile de lever la difficulté.
Nous analysons maintenant l'impact d'un composant quelconque sur le Jacobien. Nous montrons que si la concentration de ce composant est très petite, alors le Jacobien est proche de la singularité.

Proposition 1.3 *On suppose que les $E_{ji} = 0,\ j = 1, ..., N_p$ pour un composant c_i, que $S_{ki} \geq 0$ et $A_{ki} \geq 0$. Pour ce composant, la ligne associée à $T_{c,i}$ tend vers 0 si c_i tend vers 0 et la matrice J_l est proche de la singularité.*
Pour un composant fixé s_i, on suppose que $B_{ki} \geq 0$. Alors la ligne associée à $T_{s,i}$ tend vers 0 si s_i tend vers 0 et la matrice J_l est singulière.

Preuve 1.3 Pour un composant aqueux c_i, la démonstration est liée à la valeur des coefficients stœchiométriques S_{ki} et A_{ki}. Puisque la concentration de c_i est proche de 0, alors $\lim_{c_i \longrightarrow 0} lc_i = -\infty$.

a - Si $S_{ki} > 0$, nous avons $\lim_{c_i \longrightarrow 0} \alpha_k(lc) = 0$
Alors $\lim_{c_i \longrightarrow 0} S_{ki}\ \alpha_k = 0$

b - Si $A_{ki} > 0$, nous avons $\lim_{c_i \longrightarrow 0} \beta_k(lc) = 0$
Alors $\lim_{c_i \longrightarrow 0} A_{ki}\ \beta_k = 0$

c - Si $S_{ki} = 0$, nous avons $S_{ki}\ \alpha_k = 0$.

d - Si $A_{ki} = 0$, nous avons $A_{ki}\ \beta_k = 0$.

Puisque nous avons supposé que $E_{ji} = 0$, $S_{ki} \geq 0$ et $A_{ki} \geq 0$ alors $S_{ki}\ \alpha_k$ est nul ou tend vers 0 et aussi $A_{ki}\ \beta_k$. Donc :

$$\begin{cases} \dfrac{\partial T_{c,i}}{\partial (lc_j)} & \longrightarrow & 0 \\ \dfrac{\partial T_{c,i}}{\partial (ls_j)} & \longrightarrow & 0 \\ \dfrac{\partial T_{c,i}}{\partial p_j} & = & 0 \end{cases} \tag{1.37}$$

Alors la matrice Jacobienne est singulière.
Pour un composant fixé s_i, la démonstration est liée à la valeur des coefficients stœchiométriques B_{ki}.
Puisque la concentration de s_i est proche de 0, alors $\lim\limits_{s_i \longrightarrow 0} ls_i = -\infty$.

a - Si $B_{ki} > 0$, nous avons $\lim\limits_{s_i \longrightarrow 0} \beta_k(lc, ls) = 0$
Alors $\lim\limits_{s_i \longrightarrow 0} B_{ki}\ \beta_k = 0$

b - Si $B_{ki} = 0$, nous avons $B_{ki}\ \beta_k = 0$.

Puisque nous avons supposé que $B_{ki} \geq 0$ alors $B_{ki}\ \beta_k$ est nul ou tend vers 0. Donc :

$$\begin{cases} \dfrac{\partial T_{s,i}}{\partial (lc_j)} & \longrightarrow & 0 \\ \dfrac{\partial T_{s,i}}{\partial (ls_j)} & \longrightarrow & 0 \\ \dfrac{\partial T_{s,i}}{\partial p_j} & = & 0 \end{cases} \tag{1.38}$$

Alors la matrice Jacobienne est singulière.

Remarque 1.2 *Si $c_i \longrightarrow 0$ et si $S_{ki} < 0$, alors on peut avoir $\alpha_k(c) \longrightarrow \infty$; c'est pourquoi on suppose que $S_{ki} \geq 0$.*
De même pour A_{ki} et B_{ki}.

Remarque 1.3 *Si $E_{ji} > 0$ et $c_i = 0$ alors $\Pi_i(c) = 1$ mais dans le modèle simplifié $\Pi_i(c) = 0$ puisque $p_i > 0$. Donc si $c_i = 0$, nous avons $E_{ji} = 0,\ j = 1, ...N_p$*

1.7 La neutralité électrique

Dans une solution aqueuse, on peut supposé en première approximation qu'il y a la même quantité de charges positives que de charges négatives et cela s'écrit sous la forme suivante :

$$\sum_i z_i m_i = 0$$

ou z_i est la charge de l'espèce E_i. Cette équation montre que les concentrations totales sont liées par une condition linéaire.

1.8 Conclusion

Dans ce chapitre, nous avons défini un modèle chimique incluant des réactions, à l'équilibre, sans réactions cinétiques.
Ce modèle est un système algébrique d'équations non linéaires avec contraintes d'inégalité. Dans une version simplifiée, nous supposons que les minéraux sont présents, ce qui permet de calculer le Jacobien du système. Dans la version générale, les minéraux peuvent être dissous. Nous utilisons un problème de complémentarité et une fonction semismooth pour calculer le Jacobien.
D'autre part, nous étudions l'impact de concentrations nulles sur le conditionnement du Jacobien. Nous montrons, que dans le cas d'un traceur, la ligne associée est celle de l'identité. Toutefois, ce modèle impose des contraintes de positivité des concentrations. De plus, le calcul du Jacobien est compliqué c'est pourquoi nous définissons aussi un modèle simplifié avec des variables logarithmiques. Nous montrons que dans le cas d'un traceur ou d'un composant à très faible concentration, la ligne associée du Jacobien est presque nulle.

Chapitre 2

Transport réactif

Ce chapitre présente la modélisation du transport réactif en milieu poreux saturé. On suppose que le flux est stationnaire et la vitesse du fluide est calculée à l'avance par une simulation d'écoulement. La porosité est constante et le transport réactif lié aux deux phénomènes advection et dispersion ; de plus, les coefficients de dispersion ne dépendent pas de l'espèce.
On couple ici les deux modèles de chimie et de transport, cette modélisation conduit à un système d'équations aux dérivées partielles et algébriques. Ensuite nous décrivons les méthodes numériques utilisées pour résoudre ce type de problème, une méthode implicite et une méthode explicite non itérative. À la fin, on réduit la taille du système résolu à chaque itération. Ensuite, on réduit le nombre des inconnues du système non linéaire à chaque pas temps.

2.1 Modèle de transport

2.1.1 Loi de Darcy

La loi de Darcy est utilisée pour calculer les écoulements souterrains de l'eau ou d'un liquide. Cette loi contrôle l'écoulement stationnaire d'un fluide incompressible caractérisé par une viscosité dynamique μ et une masse volumique ρ au travers d'un milieu poreux caractérisé par un coefficient de perméabilité K, elle permet de mesurer et modéliser le débit d'eau pouvant s'écouler à travers un sol de section A et de longueur L [11, 46].

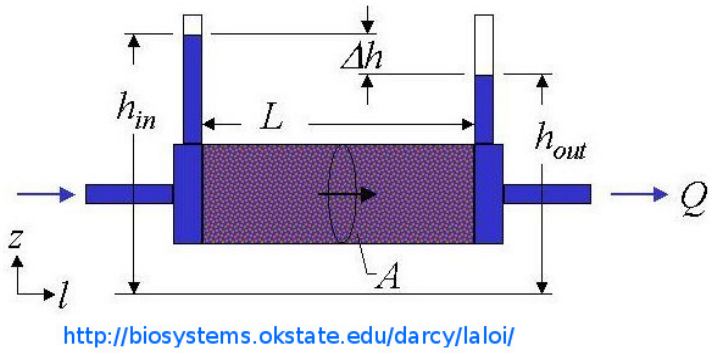


FIGURE 2.1 – section de sable soumis à un débit de fluide Q

La vitesse de Darcy généralisée pour un cas tridimensionnel est donnée par l'équation suivante :

$$q = \frac{Q}{A} = -\mathbf{K}\nabla h \tag{2.1}$$

avec :

- q la vitesse de Darcy $[LT^{-1}]$.
- $\mathbf{K}$ la conductivité hydraulique $[LT^{-1}]$.
- h la charge piézomètrique $[L]$.

La conductivité hydraulique $\mathbf{K}$ dépend des propriétés du milieu poreux et s'exprime par la loi suivante :

$$K = \frac{k.\rho.g}{\mu}$$

où :

- k est la perméabilité intrinsèque du milieu poreux$[L^2]$.
- ρ est la masse volumique du fluide $[ML^{-3}]$.
- g est l'accélération de la pesanteur $[LT^{-2}]$.
- μ est la viscosité dynamique du fluide $[ML^{-1}T^{-1}]$.

La vitesse de Darcy q est liée à la porosité ε et la vitesse d'infiltration v :

$$q = \varepsilon v$$

2.1.2 L'écoulement (équation de type diffusion)

L'équation classique en hydrogéologie (Lois dynamiques)[46] s'écrit sous la forme suivante :

$$S_s \frac{\partial h}{\partial t} + \nabla q = q_s \tag{2.2}$$

où S_s est le coefficient d'emmagasinement spécifique et q_s est le terme algébrique source. En régime permanent, on obtient :

$$\nabla q = q_s$$

2.1.3 L'advection

L'équation de transport par advection s'écrit sous la forme suivante :

$$-\nabla.(q\ c) = \varepsilon \frac{\partial c}{\partial t} \tag{2.3}$$

où c est la concentration d'espèce aqueuse.

2.1.4 La dispersion

L'équation de transport par dispersion s'écrit sous la forme suivante :

$$\nabla.(D\ \nabla c) = \varepsilon \frac{\partial c}{\partial t} \tag{2.4}$$

avec

$$D = \varepsilon d_m I + \alpha_T \|q\| I + (\alpha_L - \alpha_T)\frac{qq^T}{\|q\|}$$

— d_m est le coefficient de diffusion moléculaire $[L^2T^{-1}]$.
— α_L est le coefficient de dispersion longitudinale $[L]$.
— α_T est le coefficient de dispersion transversal $[L]$.

2.1.5 L'équation d'advection-dispersion-réaction

On suppose que la porosité ε ne varie pas en temps mais peut varier en espace. Pour un soluté, l'équation de transport s'écrit sous la forme suivante :

$$\varepsilon \frac{\partial c}{\partial t} = \nabla.(D\ \nabla c) - \nabla.(q\ c) + q_s c_s + \varepsilon R \tag{2.5}$$

où

— q_s est le terme puit-source intervenant dans l'équation d'écoulement.
— c_s est la concentration puisée ou injectée par le puit ou la source.
— R est le terme de réaction chimique.

2.2 Couplage entre chimie et transport

Dans ce travail, on suppose que les espèces fixées sont immobiles et que le terme algébrique source de ces espèces est nul. Alors l'équation de transport pour chaque espèce chimique est définie par le système suivant :

$$\left\{\begin{array}{lcll}
\varepsilon \dfrac{\partial c_i}{\partial t} & = & \nabla.(D\ \nabla c_i) - \nabla.(q\ c_i) + q_s c_c(i) + \varepsilon R_{c,i} & i = 1, N_c \\
\varepsilon \dfrac{\partial \alpha_i}{\partial t} & = & \nabla.(D\ \nabla \alpha_i) - \nabla.(q\ \alpha_i) + q_s c_\alpha(i) + \varepsilon R_{\alpha,i} & i = 1, N_\alpha \\
\varepsilon \dfrac{\partial s_i}{\partial t} & = & \varepsilon R_{s,i} & i = 1, N_s \\
\varepsilon \dfrac{\partial \beta_i}{\partial t} & = & \varepsilon R_{\beta,i} & i = 1, N_\beta \\
\varepsilon \dfrac{\partial p_i}{\partial t} & = & \varepsilon R_{p,i} & i = 1, N_p
\end{array}\right. \tag{2.6}$$

où $R_c, R_s, R_\alpha, R_\beta$ et R_p sont les taux de production des différentes espèces.
On suppose que l'opérateur de transport est indépendant de l'espèce transportée et est donc linéaire[40]. Sous l'hypothèse où les réactions chimiques sont à l'équilibre, la concentration totale est fixée dans chaque réaction et, à partir de l'équation (1.14) définie dans le premier chapitre, nous obtenons le système suivant :

$$\left\{\begin{array}{lcll} \varepsilon\dfrac{\partial T_i}{\partial t} & = & \nabla.(D\ \nabla C_i) - \nabla.(q\ C_i) + q_s C_{s,i}, & i = 1, Nc \\ \varepsilon\dfrac{\partial W_i}{\partial t} & = & 0 & i = 1, Ns \end{array}\right. \tag{2.7}$$

où C_i est la partie aqueuse de la concentration totale T_i et $C_{s,i}$ est la concentration totale puisée.
La deuxième équation de système (2.7) signifie que les concentrations totales fixées ne changent pas pendant le transport des espèces, donc il reste la première équation différentielle.
Ces équations aux dérivées partielles sont écrites dans un domaine $\Omega \in \mathbb{R}^3$, alors des conditions aux bords sont ajoutées au système ; Il y a plusieurs types de conditions aux limites [46, 12] :

- Condition de Dirichlet : la concentration est imposée le long de la frontière.
- Condition de Neumann : le gradient de la concentration est imposé le long de la frontière.
- Condition de Cauchy : la concentration et le gradient de concentration sont tous les deux imposés le long de la frontière.

On ajoute aussi les valeurs des concentrations totales aqueuses T à l'instant initial. Ce système avec les conditions initiales et les conditions aux limites est couplé avec l'équation de chimie (1.16) définie dans le premier chapitre. Les inconnues supplémentaires sont les concentrations c, s, p définies en chaque point de l'espace et à chaque instant.
Pour avoir un système bien posé, on ajoute l'équation $C(c) = c + S^T\alpha$ qui exprime la variable C en fonction des inconnues c.
Soit

$$\Phi(c, s, p) = \begin{pmatrix} T - T_c(c, s, p) \\ W - T_s(c, s) \\ \varphi(c, p) \end{pmatrix} \tag{2.8}$$

où $\varphi(c, p) = -\Pi(c)$ dans le modèle simplifié et $\varphi_{\min}(\Pi(c), p)$ dans le modèle général, T_c et T_s sont définis au chapitre 1. Alors nous avons le système suivant :

$$\left\{\begin{array}{lcl} \varepsilon\dfrac{\partial T}{\partial t} & = & \nabla.(D\ \nabla C) - \nabla.(q\ C) + Q \\ \Phi(c, s, p) & = & 0 \\ C & = & C_c(c) \\ & + & \text{conditions aux limites} \\ & + & \text{conditions initiales} \\ & + & \text{contraintes de positivité} \end{array}\right. \tag{2.9}$$

avec $C_c(c) = c + S^T\alpha$ et Q est le terme source.

Dans le système (2.9), on choisit de conserver les inconnues T et C. Dans l'approche DSA (approche de substitution directe[27]), C est remplacé par $C_c(c)$ dans l'équation de transport.
Cette substitution permet de réduire le nombre d'inconnues, au prix d'un couplage fort entre transport et chimie. Nous l'étudions au section 2.5.1.
Parfois, T est aussi remplacé par $T_c(c, s, p)$. Cette substitution n'est pas judicieuse car elle introduit une complexité non linéaire dans la dérivée en temps.

2.3 Méthodes Numériques

La modélisation du transport réactif en milieu poreux conduit ainsi au système d'équations aux dérivées partielles et algébriques (2.9) dont les inconnues sont les quantités des espèces chimiques.
La plupart des modèles numériques utilisent une méthode des lignes, qui induit un système semi-discret algébro-différentiel, après discrétisation en espace [8].
Cette discrétisation est faite par un schéma Eulérien, de type volumes finis ou différences finies. Ensuite, il faut discrétiser en temps. Dans ce chapitre, on présente deux méthodes : la méthode globale avec approche DAE (GDAE) et la méthode séquentielle non itérative (SNIA).

2.3.1 Discrétisation spatiale

La discrétisation spatiale choisie doit éviter les artefacts numériques tels qu'oscillations ou diffusion artificielle. L'opérateur de transport discret s'écrit $LC - G$, où L est une matrice carrée d'ordre N_m, le nombre de points spatiaux et où G représente les conditions aux limites.

Les variables du modèle numérique de transport réactif sont stockées dans des tableaux, par exemple $T = (T_1, \ldots, T_{N_m})$. La colonne $T_j, j = 1, \ldots, N_m$ contient toutes les composantes en un point et la ligne $T_i, i = 1, \ldots, N_c$ contient une composante en tous les points.
Les autres variables c, s, p, C, Q, G et W suivent le même stockage de T avec la même taille, avec $s \in \mathbb{R}^{N_s \times N_m}$, $p \in \mathbb{R}^{N_p \times N_m}$ et $W \in \mathbb{R}^{N_s \times N_m}$.
Avec ces notations, le système semi-discret de transport s'écrit

$$\left\{\begin{array}{lcll} M\frac{dT_i}{dt} - LC_i & = & Q_i + G_i, & i = 1, \ldots, N_c, \\ T_j - T_{c,j}(c_j, s_j, p_j) & = & 0, & j = 1, \ldots, N_m, \\ W_j - T_{s,j}(c_j, s_j) & = & 0, & j = 1, \ldots, N_m, \\ \varphi_j(c_j, p_j) & = & 0, & j = 1, \ldots, N_m, \\ C_j & = & C_c(c_j), & j = 1, \ldots, N_m, \\ \text{condition initiale pour } T. & & & \\ \text{contraintes .} & & & \end{array}\right. \tag{2.10}$$

où $M = \mathbf{diag}(\varepsilon)$.

2.3.2 La méthode séquentielle non itérative SNIA

Les méthodes séquentielles découplent les deux opérateurs de chimie et de transport. Le système chimique est résolu indépendamment sur chaque point du maillage. L'approche séquentielle non itérative SNIA est une méthode basée sur le schéma d'Euler explicite et s'écrit très simplement, avec un pas de temps Δt :

$$\begin{cases} MT_i^{n+1} - \Delta t L C_i^n & = & MT_i^n + \Delta t(Q_i^n + G_i^n), & i = 1, \ldots, N_c, \\ T_j^{n+1} - T_{c,j}(c_j^{n+1}, s_j^{n+1}, p_j^{n+1}) & = & 0, & j = 1, \ldots, N_m, \\ W_j^{n+1} - T_{s,j}(c_j^{n+1}, s_j^{n+1}) & = & 0, & j = 1, \ldots, N_m, \\ \varphi_j(c_j^{n+1}, p_j^{n+1}) & = & 0, & j = 1, \ldots, N_m, \\ C_j^{n+1} & = & C_c(c_j^{n+1}), & j = 1, \ldots, N_m, \end{cases}$$

À chaque pas de temps, on calcule explicitement et de façon indépendante les totaux T_i, puis on résout dans chaque maille de façon indépendante la spéciation chimique. Les contraintes de stabilité imposent de choisir un pas de temps Δt petit, lié au pas d'espace Δx.

2.3.3 La méthode globale avec approche DAE (GDAE)

Les méthodes globales sont basées sur un schéma en temps implicite, qui conduit à un système non linéaire couplé à chaque pas de temps, résolu globalement par une méthode de Newton.
Ces méthodes sont stables grâce au schéma implicite. Il y a trois types de méthodes globales : les approches basées sur les équations différentielles ordinaires (approches globales ODE)[7], les approches de substitution directe (approches globales DSA) [27] et les approches basées sur les équations différentielles algébriques (approches globales DAE), que nous présentons dans ce chapitre.
Le système semi-discret (2.10) est un système algébro-différentiel d'indice 1, dont le Jacobien peut être calculé [8, 16].
Dans ce travail, on introduit le produit de Kronecker $\otimes$ et la vectorisation d'une matrice **vec** ; la vectorisation d'une matrice est une transformation linéaire qui convertit la matrice dans un vecteur colonne ;
On peut définir les inconnues du système (2.10) de cette manière $Y = (\mathbf{vec}(T),\ X,\ \mathbf{vec}(C))$ avec $X = (\mathbf{vec}(c),\ \mathbf{vec}(s),\ \mathbf{vec}(p))$.
Ici, nous avons choisi une numérotation successive de c, puis s, puis p. Dans la section 2.5.1, nous étudions une numérotation groupée, soit **vec**(c,s,p).
Le système semi-discret (2.10) écrit sous la forme suivante :

$$F(t, Y, Y') = 0$$

avec

$$F\left(t, Y, Y'\right) = \begin{pmatrix} (M \otimes I_{N_c})\mathbf{vec}(T') - (L \otimes I_{N_c})(\mathbf{vec}(C)) - \mathbf{vec}(G + Q) \\ \mathbf{vec}(T) - \mathbf{vec}(T_c) \\ \mathbf{vec}(W) - \mathbf{vec}(T_s) \\ \mathbf{vec}(\varphi) \\ \mathbf{vec}(C) - \mathbf{vec}(C_c(X)) \end{pmatrix}. \quad (2.11)$$

La méthode GDAE utilise un schéma BDF, la dérivée en temps est approximée par la formule suivante :

$$Y'(t_{n+1}) \simeq \frac{a_0}{\Delta t} Y^{n+1} + \frac{1}{\Delta t} \overline{Y}^n \tag{2.12}$$

avec $\overline{Y}^n = \sum_{i=1}^{k} a_{n+1,i} Y^{n+1-i}$.
où k est l'ordre de la méthode.
Le système est transformé, à chaque pas de temps, en un système non linéaire :

$$F\left(t^{n+1}, Y^{n+1}, \frac{a_0}{\Delta t} Y^{n+1} + \frac{1}{\Delta t} \overline{Y}^n\right) = 0 \tag{2.13}$$

Il s'écrit, en omettant l'indice $(n+1)$:

$$\begin{cases} \frac{a_0}{\Delta t}(M \otimes I_{N_c})\mathbf{vec}(T) - (L \otimes I_{N_c})(\mathbf{vec}(C)) + \frac{1}{\Delta t}(M \otimes I_{N_c})\overline{T}^n - \mathbf{vec}(G+Q) = 0 \\ \mathbf{vec}(T) - \mathbf{vec}(T_c) = 0 \\ \mathbf{vec}(W) - \mathbf{vec}(T_s) = 0 \\ \mathbf{vec}(\varphi) = 0 \\ \mathbf{vec}(C) - \mathbf{vec}(C_c(X)) = 0 \end{cases}$$

Le Jacobien s'écrit sous la forme suivante :

$$J_G = \begin{pmatrix} \frac{a_0}{\Delta t} M \otimes I_{N_c} & 0 & -L \otimes I_{N_c} \\ N & J & 0 \\ 0 & -\frac{\partial(\mathbf{vec}C_c)}{\partial X} & I_{N_m N_c} \end{pmatrix}. \tag{2.14}$$

avec

$$N = \begin{pmatrix} I_{N_c N_m} \\ 0 \\ 0 \end{pmatrix}$$

Et J est la jacobienne du système chimique à tous les points de maillage.

$$J = \begin{pmatrix} -\mathbf{diag}(\frac{\partial T_c}{\partial c}) & -\mathbf{diag}(\frac{\partial T_c}{\partial s}) & -\mathbf{diag}(\frac{\partial T_c}{\partial p}) \\ -\mathbf{diag}(\frac{\partial T_s}{\partial c}) & -\mathbf{diag}(\frac{\partial T_s}{\partial s}) & 0 \\ \mathbf{diag}(\frac{\partial \varphi}{\partial c}) & 0 & \mathbf{diag}(\frac{\partial \varphi}{\partial p}) \end{pmatrix} \tag{2.15}$$

où les dérivées $\frac{\partial T_c}{\partial c}, \frac{\partial T_c}{\partial s}, \frac{\partial T_c}{\partial p}, \frac{\partial T_s}{\partial c}, \frac{\partial T_s}{\partial s}, \frac{\partial \varphi}{\partial c}$ et $\frac{\partial \varphi}{\partial p}$ sont définies dans le chapitre 1.
La dérivée partielle de C_c par rapport à X est calculée par :

$$\begin{aligned} \frac{\partial(\mathbf{vec}C_c)}{\partial X} &= (\mathbf{diag}(\frac{\partial C_c}{\partial c}) \;\; 0 \;\; 0) \\ &= (\mathbf{diag}(I + S^T \frac{\partial \alpha}{\partial c}) \;\; 0 \;\; 0) \end{aligned}$$

2.4 Réduction du système linéaire

La matrice jacobienne a une structure particulière, avec des blocs nuls et des blocs identités. Donc, il y a la possibilité de réduire la taille de système résolu à chaque itération de Newton. Autrement dit on peut appliquer la méthode de substitution à chaque système linéaire, en éliminant les inconnues T et C.
À chaque itération k, on résout le système suivant :

$$J_G(Y_k)\delta Y_k = -F_k \tag{2.16}$$

avec

$$\begin{aligned} F_k &= F(t^{n+1}, Y_k, \frac{a_0}{\Delta t}Y_k + \overline{Y}^n) \\ &= (F_T, F_c, F_s, F_p, F_C) \end{aligned} \tag{2.17}$$

et

$$\delta Y_k = (\delta T, \delta c, \delta s, \delta p, \delta C)$$

où

$$\begin{aligned} \delta T &= \mathbf{vec}(T_{k+1} - T_k) \\ \delta c &= \mathbf{vec}(c_{k+1} - c_k) \\ \delta s &= \mathbf{vec}(s_{k+1} - s_k) \\ \delta p &= \mathbf{vec}(p_{k+1} - p_k) \\ \delta C &= \mathbf{vec}(C_{k+1} - C_k) \end{aligned}$$

La première équation de (2.16) s'écrit

$$\frac{a_0}{\Delta t}(M \otimes I)\delta T - (L \otimes I)\delta C = -F_T$$

En utilisant le fait que $(M \otimes I)^{-1} = (M^{-1} \otimes I)$ et que $(M \otimes I)^{-1}(L \otimes I) = (M^{-1}L \otimes I)$, on transforme la première équation

$$\delta T - \frac{\Delta t}{a_0}(M^{-1}L \otimes I)\delta C = -\frac{\Delta t}{a_0}(M^{-1} \otimes I)F_T$$

La deuxième équation de (2.16) s'écrit

$$\delta T - \mathbf{diag}(\frac{\partial T_c}{\partial c})\delta c - \mathbf{diag}(\frac{\partial T_c}{\partial s})\delta s - \mathbf{diag}(\frac{\partial T_c}{\partial p})\delta p = -F_c$$

En combinant avec la deuxième équation, on élimine δT :

$$-\mathbf{diag}(\frac{\partial T_c}{\partial c})\delta c - \mathbf{diag}(\frac{\partial T_c}{\partial s})\delta s - \mathbf{diag}(\frac{\partial T_c}{\partial p})\delta p + \frac{\Delta t}{a_0}(M^{-1}L \otimes I)\delta C = -F_c + \frac{\Delta t}{a_0}(M^{-1} \otimes I)F_T \tag{2.18}$$

La dernière équation s'écrit

$$\delta C - \mathbf{diag}(\frac{\partial C_c}{\partial c})\delta c = -F_C$$

En utilisant la dernière équation, on élimine δC dans l'équation (2.18) :

$$\begin{array}{l}[-\mathbf{diag}(\frac{\partial T_c}{\partial c}) + \frac{\Delta t}{a_0}(M^{-1}L \otimes I)\mathbf{diag}(\frac{\partial C_c}{\partial c})]\delta c \; - \mathbf{diag}(\frac{\partial T_c}{\partial s})\delta s \\ -\mathbf{diag}(\frac{\partial T_c}{\partial p})\delta p = -(F_c - \frac{\Delta t}{a_0}(M^{-1} \otimes I)F_T - \frac{\Delta t}{a_0}(M^{-1}L \otimes I)F_C)\end{array} \tag{2.19}$$

On obtient donc le système réduit

$$J_R \delta X = -F_R$$

où

$$J_R = \begin{pmatrix} -\mathbf{diag}(\frac{\partial T_c}{\partial c}) + \frac{\Delta t}{a_0}(M^{-1}L \otimes I)\mathbf{diag}(\frac{\partial C_c}{\partial c}) & -\mathbf{diag}(\frac{\partial T_c}{\partial s}) & -\mathbf{diag}(\frac{\partial T_c}{\partial p}) \\ -\mathbf{diag}(\frac{\partial T_s}{\partial c}) & -\mathbf{diag}(\frac{\partial T_s}{\partial s}) & 0 \\ \mathbf{diag}(\frac{\partial \varphi}{\partial c}) & 0 & \mathbf{diag}(\frac{\partial \varphi}{\partial p}) \end{pmatrix} \tag{2.20}$$

et

$$F_R = \begin{pmatrix} F_c - \frac{\Delta t}{a_0}(M^{-1} \otimes I)F_T - \frac{\Delta t}{a_0}(M^{-1}L \otimes I)F_C \\ F_s \\ F_p \end{pmatrix} = \begin{pmatrix} F_{R,c} \\ F_s \\ F_p \end{pmatrix} \tag{2.21}$$

On a

$$F_T = \frac{a_0}{\Delta t}(M \otimes I_{N_c})\mathbf{vec}(T) - (L \otimes I_{N_c})(\mathbf{vec}(C)) + \frac{1}{\Delta t}(M \otimes I_{N_c})\overline{T}^n - \mathbf{vec}(G+Q)$$

donc

$$\frac{\Delta t}{a_0}(M^{-1} \otimes I_{N_c})F_T = \mathbf{vec}(T) - \frac{\Delta t}{a_0}(M^{-1}L \otimes I_{N_c})(\mathbf{vec}(C)) + \frac{1}{a_0}\overline{T}^n - \frac{\Delta t}{a_0}(M^{-1} \otimes I_{N_c})\mathbf{vec}(G+Q)$$

On a

$$F_C = \mathbf{vec}(C) - \mathbf{vec}(C_c(X))$$

donc

$$\frac{\Delta t}{a_0}(M^{-1}L \otimes I_{N_c})F_C = \frac{\Delta t}{a_0}(M^{-1}L \otimes I_{N_c})\mathbf{vec}(C) - \frac{\Delta t}{a_0}(M^{-1}L \otimes I_{N_c})\mathbf{vec}(C_c(X))$$

On a

$$F_c = \mathbf{vec}(T) - \mathbf{vec}(T_c)$$

donc

$$F_{R,c} = -\mathbf{vec}(T_c) + \frac{\Delta t}{a_0}(M^{-1}L \otimes I_{N_c})\mathbf{vec}(C_c(X)) - \frac{1}{a_0}\overline{T}^n + \frac{\Delta t}{a_0}(M^{-1} \otimes I_{N_c})\mathbf{vec}(G+Q)$$

On calcule δT et δC à partir de δc

$$\left\{ \begin{array}{lcl} \delta C & = & \mathbf{diag}(\dfrac{\partial C_c}{\partial c})\delta c - F_C \\ \delta T & = & \dfrac{\Delta t}{a_0}(M^{-1}L \otimes I)\delta C - \dfrac{\Delta t}{a_0}(M^{-1} \otimes I)F_T \end{array} \right. \tag{2.22}$$

En éliminant δT et δC, on obtient ainsi un système réduit avec δc, de taille $(N_c + N_s + N_p) \times N_m$ au lieu de $(3N_c + N_s + N_p) \times N_m$.

2.5 Réduction du système non linéaire

2.5.1 Réduction du système différentiel avec l'élimination de C

À partir de l'équation suivante :

$$C(X) - C_c(X) = 0$$

On peut éliminer l'inconnue C, donc on peut définir les inconnues du système de cette manière $Y = (\mathbf{vec}(T), \mathbf{vec}(X))$. Cette substitution induit un couplage transport- chimie dans la première équation. De plus, on choisit une numérotation groupée c'est-à-dire $\mathbf{vec}(X) = \mathbf{vec}(c, s, p)$.
Le système semi-discret (2.11) s'écrit sous la forme suivante

$$H(t, Y, Y') = 0$$

avec

$$H\left(t, Y, Y'\right) = \left(\begin{array}{c} (M \otimes I_{N_c})\mathbf{vec}(T') - (L \otimes I_{N_c})(\mathbf{vec}(C_c(X))) - \mathbf{vec}(G + Q) \\ \\ \mathbf{vec}(\Phi(c, s, p)) \end{array} \right) \tag{2.23}$$

On a un système algébro-différentiel de taille $(2N_c + N_s + N_p) \times N_m$ au lieu de $(3N_c + N_s + N_p) \times N_m$.
On utilise un schéma BDF alors on obtient à chaque pas de temps un système non linéaire similaire à (2.13), avec la fonction :

$$H(t^{n+1}, Y^{n+1}, \frac{a_0}{\Delta t}Y^{n+1} + \frac{1}{\Delta t}\overline{Y}^n) = 0 \tag{2.24}$$

$$H(t^{n+1}, Y^{n+1}, \frac{a_0}{\Delta t}Y^{n+1} + \frac{1}{\Delta t}\overline{Y}^n) =$$
$$\left(\begin{array}{c} \dfrac{a_0}{\Delta t}(M \otimes I_{N_c})\mathbf{vec}(T) - (L \otimes I_{N_c})(\mathbf{vec}(C_c(X))) + \dfrac{1}{\Delta t}(M \otimes I_{N_c})\mathbf{vec}\overline{T}^n - \mathbf{vec}(G + Q) \\ \\ \mathbf{vec}(\Phi(c, s, p)) \end{array} \right)$$

Le jacobien de ce système s'écrit sous la forme suivante :

$$J_{R,C} = \begin{pmatrix} \frac{a_0}{\Delta t}(M \otimes I_{N_c}) & -(L \otimes I_{N_c})(\mathbf{diag}\frac{\partial C_c(X_j)}{\partial X}) \\ I_{N_m} \otimes N & \mathbf{diag}(\Psi(X_j)) \end{pmatrix} \tag{2.25}$$

où $\mathbf{diag}(\Psi(X_j))$ est la matrice diagonale par blocs (ou bloc-diagonale), où les blocs $\Psi(X_j)$, qui sont les matrices jacobiennes de chimie définies en chaque point j du maillage et où

$$N = \begin{pmatrix} I_{N_c} \\ 0 \\ 0 \end{pmatrix}$$

Cette transformation est simple à mettre en œuvre dans le logiciel GRT3D.

2.5.2 Réduction du système non linéaire avec l'élimination de T

On peut éliminer C dans chaque système linéaire comme précédemment. Ici, on va éliminer T dans le système non linéaire (2.24), pour obtenir un système non linéaire avec $(N_c + N_s + N_p) \times N_m$ inconnues. On peut écrire l'équation $\mathbf{vec}(\Phi(X)) = 0$ sous la forme suivante :

$$\begin{aligned} &\mathbf{vec}(\phi(X)) + (I \otimes N)\mathbf{vec}(T) = 0 \\ &\phi(c,s,p) = \begin{pmatrix} -T_c(c,s,p) \\ -T_s(c,s) + W \\ \varphi(c,p) \end{pmatrix} \end{aligned} \tag{2.26}$$

On peut écrire la première équation sous la forme suivante :

$$\mathbf{vec}(T) - \frac{\Delta t}{a_0}(M^{-1}L \otimes I_{N_c})\mathbf{vec}(C_c(X)) + \frac{1}{a_0}\mathbf{vec}(\overline{T}^n) - \frac{\Delta t}{a_0}(M^{-1} \otimes I_{N_c})\mathbf{vec}(G+Q) = 0$$

donc, on a le système suivant :

$$\begin{aligned} \mathbf{vec}(\phi(X)) + \frac{\Delta t}{a_0}(M^{-1}L \otimes N)\mathbf{vec}(C_c(X)) - \frac{1}{a_0}(I_{N_m} \otimes N)\mathbf{vec}(\overline{T}^n) \\ + \frac{\Delta t}{a_0}(M^{-1} \otimes N)\mathbf{vec}(G+Q) = 0 \end{aligned} \tag{2.27}$$

Le jacobien de ce système est le suivant

$$J_{R,T}(X) = \mathbf{diag}(\Psi(X_j)) + \frac{\Delta t}{a_0}(M^{-1}L \otimes N)\mathbf{diag}(\frac{\partial(C_c(X_j))}{\partial X}) \tag{2.28}$$

Remarque 1 *Le Jacobien $J_{R,T}$ en (2.28) est, à une permutation près, le Jacobien J_R en (2.20).*

Proposition 2.1 *On supposant que la matrice du système chimique $\mathbf{diag}(\Psi(X_j))$ est inversible en tous point j, alors, pour un Δt assez petit, le jacobien $J_{R,T}$ est inversible.*

Preuve 2.1 *Puisque* $\mathbf{diag}(\Psi(X_j))$ *est inversible, on peut écrire la matrice* $J_{R,T}$ *sous la forme suivante :*

$$J_{R,T} = \mathbf{diag}(\Psi(X_j))[I + \frac{\Delta t}{a_0}[\mathbf{diag}(\Psi(X_j))]^{-1}(M^{-1}L \otimes N)\mathbf{diag}(\frac{\partial(C_c(X_j))}{\partial X})]$$

Si

$$\|\Delta t\| < \frac{a_0}{\|[\mathbf{diag}(\Psi(X_j))]^{-1}(M^{-1}L \otimes N)\mathbf{diag}(\frac{\partial(C_c(X_j))}{\partial X})]\|}$$

alors $I + \frac{\Delta t}{a_0}[\mathbf{diag}(\Psi(X_j))]^{-1}(M^{-1}L \otimes N)\mathbf{diag}(\frac{\partial(C_c(X_j))}{\partial X})$ *est inversible, et* $J_{R,T}$ *est également inversible.*

Cette transformation n'est pas immédiate dans le logiciel GRT3D.
En effet, cela nécessite de modifier l'interface avec IDA (est un package pour résoudre les systèmes d'équation différentielle algébrique (DAE), sous la forme $F(t, y, y') = 0$) au niveau non linéaire.

2.6 Conclusion

Dans ce chapitre, nous avons défini le modèle de transport qui est basé sur la loi de Darcy, l'écoulement, l'advection et la dispersion.
Ensuite, nous avons défini le modèle de couplage entre le transport et la chimie présentée dans le chapitre (1).
Nous avons présenté aussi les méthodes numériques SNIA et GDAE utilisées pour résoudre le système algébro- différentiel couplé. Pour l'approche GDAE qui est basée sur la méthode de Newton, nous avons défini la matrice Jacobienne du système.
La matrice Jacobienne a une structure particulière qui permet de réduire la taille du système linéaire à chaque itération de Newton, de $(3N_c + N_s + N_p) \times N_m$ à $(N_c + N_s + N_p) \times N_m$, en éliminant $2N_cN_m$ inconnues.
Nous avons ensuite appliqué cette élimination plus en avant : dans le système algébro-différentiel, on élimine N_cN_m inconnues. Puis on élimine de nouveau NcN_m inconnues dans le système non linéaire.
Le Jacobien est identique au précédent et inversible sous certaines hypothèses.
Dans les chapitres 4 et 5, nous appliquons les variantes de GDAE à deux cas tests et nous comparons les résultats à ceux de l'approche SNIA.

Chapitre 3

Suite Logicielle GRT3D (Global Reactive Transport 3D)

3.1 Introduction

GRT3D concerne le transport réactif dans un milieu poreux, c'est-à-dire le couplage entre le transport et la chimie (transport des espèces chimiques et en même temps des réactions chimiques entre ces espèces). La modélisation de ce problème conduit à une équation aux dérivées partielles algébrique (PDAE).
Pour calculer les concentrations de chaque composant numériquement, Caroline de Dieuleveult a développé pendant sa thèse [12] le logiciel GRT3D, version 1, qui contient trois parties :

1. SIA1D : Méthode Séquentielle Itérative en 1D. Discrétisation spatiale par volumes finis avec un module dédié.

2. GDAE1D : Méthode Globale par un système d'équations Différentielles Algébriques en 1D (discrétisation spatiale par le module volumes finis).

3. GDAE3D : Méthode Globale par un système d'équations Différentielles Algébriques en 1D, 2D et 3D (discrétisation spatiale par différences finies avec le logiciel MT3D).

Ce chapitre présente les différentes méthodes numériques et les bibliothèques utilisées (programmation en C et partiellement en Fortran).
Nous décrivons les différents points d'entrée SIA, GDAE1D, GDAE3D et la restructuration de la version 1. Puis nous décrivons les nouveaux points d'entrées dans GRT3DRL, GRT3DRSL, SNIA et GRT3DRSLC le version 2. Nous présentons la manière d'installer et d'utiliser les logiciels. Enfin, nous terminons par les résultats de quelques tests numériques.

3.2 Description de GRT3D version 1

3.2.1 Composants logiciels

La suite logicielle GRT3D a été codée en langage C. Elle contient 3 points d'entrée : SIA1D, GDAE1D et GDAE3D ; le compilateur utilisé est `GCC-4.4`. Le tableau 3.1 indique les logiciels libres utilisés, avec le numéro de version et le langage.

Bibliothèques	Version	Utilisation	Langage
MODFLOW-96[3]	3.2.0	MT3D	F90
MT3D	5.2.0	GDAE3D	F90
Sundials (IDA, KINSOL)	2.3.0	GRT3D	C
SuiteSparse	3.4.0	GRT3D	C
PGPLOT	5.2.2	SIA et GDAE1D	C

TABLE 3.1 – Logiciels utilisés au sein du projet GRT3D

Attention, les versions sundials-2.4, 2.5 et 2.6 ne fonctionnent pas, parce que certaines fonctions ont changé d'interface.
La figure 3.1 indique les liens entre les points d'entrée de GRT3D et les bibliothèques externes.

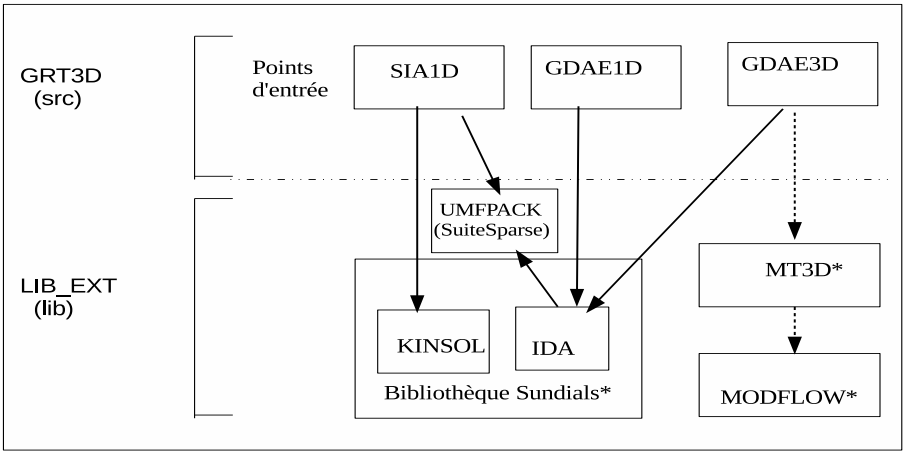


FIGURE 3.1 – Logiciel GRT3D (MODFLOW, MT3D et GDAE3D communiquent par des fichiers ; les autres interfaces sont des appels internes classiques. Le symbole * indique que les logiciels sont modifiés)

La suite logicielle contient aussi deux modules bibliothèques internes.
— Un module de transport en 1D avec une discrétisation par volumes finis.
— Un module de chimie contenant les fonctions du problème chimique et leurs dérivées. Dans cette version, le module utilise les logarithmes et le modèle simplifié de précipitation.

3.2.2 SIA1D (séquentielle itérative en 1D)

Les méthodes séquentielles permettent un découplage entre la résolution de la chimie et celle du transport. La discrétisation en temps utilise un schéma Euler implicite ; la méthode de Gauss-Seidel non linéaire est mise en œuvre pour le couplage.
Pour résoudre le problème de transport, on utilise le solveur linéaire direct UMFPACK.
On utilise le solveur KINSOL pour résoudre le problème chimique.
KINSOL est un solveur non linéaire pour les systèmes d'équations algébriques basé sur l'algorithme de Newton-Krylov.

3.2.3 GDAE1D (équations Algébro-différentielles)

Dans cette méthode, il s'agit d'écrire le modèle couplé comme un système algébro-différentiel et on résout le système avec le solveur IDA.
IDA est un package pour les systèmes algébro-différentiels de la forme $F(t, y, y') = 0$.
Il est complété par le module ida_sparse que nous avons développé pour traiter une matrice jacobienne creuse.
Le code GDAE1D est restreint au cas 1D et il utilise la matrice de transport calculée à partir de la discrétisation par volumes finis en espace. Il fournit à IDA la fonction et la Jacobienne après le couplage entre la chimie et le transport.

3.2.4 GDAE3D

C'est une extension du composant global à une, deux ou trois dimensions mais ici on utilise MT3D pour calculer la matrice de transport, avec un schéma de différences finies. On utilise deux fichiers qui sont créés par MT3D et MODFLOW :

1. A_RHS.dat (ou .bin) qui contient le nombre des composants, la matrice de transport, le second membre, les concentrations initiales et la porosité.
2. DXDYDZ.dat (ou .bin) qui contient les points de maillages du domaine.

3.3 Refactoring

3.3.1 Difficultés rencontrées

Chacun des composants cités précédemment utilisait une version de Sundials dont le code source a été modifié. Les principales difficultés ont été de répertorier les différentes modifications du code source de Sundials afin de mettre en place un système de bibliothèque extérieure [1] sans toucher au code original de Sundials.

Une autre difficulté a été de déterminer les versions finales du code GDAE3D, en effet trois versions de ce code étaient présentes ; nous avons fait des tests pour chacune d'entre elles.

L'utilisation des fichiers binaires issus de MT3D ne fonctionnait pas. Par ailleurs, le code n'offrait pas la possibilité d'utiliser des fichiers ascii ou binaires. L'utilisateur devait modifier le code en conséquence. Le passage en argument des fichiers de paramètres ne fonctionnait pas, il était nécessaire d'avoir tous les fichiers d'entrée dans un même répertoire, ce qui ne permettait pas la classification par chimie, maillage, solveur, etc

3.3.2 Restructuration sous forme de bibliothèques

La bibliothèque Sundials est maintenant utilisée sans modification. Pour cela, nous avons créé deux bibliothèques pour traiter des Jacobiens stockés avec un format de matrice creuse. La figure (3.1) montre les liens entre les points d'entrée de GRT3D et les bibliothèques Sundials.
Par contre, nous avons dû modifier légèrement Modflow et MT3D pour les interfacer avec GRT3D.

3.3.2.1 sundials_sparse_column

On a créé la bibliothèque *libsundials_sparse_column* pour l'utiliser dans GDAE1D et on a ajouté *ida_sparse.c (.h)*, *ida_sparse_impl.h* et *sundials_sparse.c (.h)*, qui ne sont pas dans sundials. Cette bibliothèque regroupe les modifications pour utiliser une matrice jacobienne creuse. Nous avons défini un type sparseMat de matrice creuse. Il est défini comme un pointeur vers une structure avec les champs suivants :

- *nz* : nombre des coefficients non nul de la matrice.
- size : nombre des colonnes de la matrice.
- *Ap* est un vecteur de taille n + 1 (le nombre de lignes plus un). *Ap*[*j*] contient l'index dans le tableau Ap de la première entrée non nulle de la colonne j de la matrice originale. La colonne j de la matrice originale est composée des éléments de Ax depuis l'index Ap(j) jusqu'à l'index Ap(j+1)-1.
- *Ai* est un vecteur de taille *nz* . Il contient le numéro de la ligne de chaque élément de Ax.
- *Ax* est un vecteur de taille *nz*. Ce sont les valeurs de la matrice dans la position correspondant.

Ici le stockage de la matrice est effectué par colonne et correspond au format classique CSC (Compressed Sparse Column).

3.3.2.2 sundials_sparse_row

On a créé la librairie *libsundials_sparse_row* pour utiliser dans GDAE3D avec des modifications dans les fichiers *ida_sparse.{c, h}*,*ida_sparse_impl.h* et *sundials_sparse.{c, h}* et on a ajouté *sundials_sparseTri.{c, h}*, avec cette librairie on stocke la matrice jacobienne creuse par ligne (CSR : Compressed Sparse Row).

3.3.2.3 MODFLOW-96

MODFLOW est utilisé ici pour simuler un écoulement en milieu poreux et calculer la vitesse nécessaire au transport. On a fait les modifications suivantes :

- On ajoute DOUBLE PRECISION (A-H,O-Z) après chaque nom de subroutine et on remplace REAL par DOUBLE PRECISION dans tous les fichiers de MODFOW-96.
- On remplace ALOG par DLOG à la ligne 641 et 654 dans le fichiers *GFD1.FOR*
- On ajoute le fichier *add_ print.FOR* qui contient la création de *head.vtk* avec la subroutine PRINT_HEAD. Ce fichier contient les points de maillages et le flux à chaque point.
- On ajoute Makefile pour compiler MODFLOW (nom de l'exécutable : *mf96.exe*).
- On ajoute à la ligne 435
  CALL PRINT_HEAD(NLAY, NROW, NCOL, NLAY*NCOL*NROW, X(LCDELR), & X(LCDELC), X(LCHNEW)) et on commente la ligne 86 (CALL GETCL(FNAME) dans le fichier *MODFLOW96.FOR*.

3.3.2.4 MT3D

MT3D est utilisé ici pour calculer la matrice de transport. nous avons fait les modifications suivantes :

- On remplace NONE par DOUBLE PRECISION (A-H,O-Z) et REAL par DOUBLE PRECISION dans tous les fichiers de MT3D-5.2.0.
- On remplace 'FILESPEC.INC' par 'filespec.inc' à la ligne 22 dans les deux fichiers *mt_ utl5.for* et *mt_ btn5.for*
- On commente les deux lignes 269 et 272 dans le fichier *mt_ ssm5.for*
- On ajoute le fichier *print_ mt3d.c* qui contient la création des fichiers *A_ RHS.dat, DXDYDZ.dat ou (.bin)*
- On ajoute le fichier *add_ print.for* qui contient la création de *velocity.vtk* avec la subroutine PRINT_HEAD. Ce fichier contient les points de maillages et le flux à chaque point.
- On ajoute Makefile pour compiler MT3D-5.2.0 (nom de l'exécutable : *mt3d*)
- On ajoute à la ligne 2082
  CALL PRINT_FLUX(NLAY, NROW, NCOL, NODES, DELR, DELC, DH, QX, QY, QZ)
  dans fichier *mt_ adv5.for*
- On supprime $+DZ(J,I,K)$ à la ligne 712 de fichier *mt_ btn5.for*, on remplace $RHS(N) =$ .... par $RHS(N) = 0.00$; à al ligne 1439 et on supprime les lignes 1474-1476.
- On remplace *mt3dms5.for* par *mt3d_ file.for* (nouvelle version écrite par Caroline, avec trop de différences pour être détaillées ici).

3.3.3 Modifications des points d'entrée

Chacun des composants possède à présent un Makefile prenant en compte les diverses configurations de compilation (Debug, Release).

3.3.3.1 SIA1D

On a fait les modification suivantes :
On a supprimé les fichiers : *cas_qualif.c*, *cas_qualif_2.c*, *jacobienne.{c,h}*, *sparse_jac.c*, *Picard_modif.c*, *transport_temp2.c* et *chimistry_local_old.c* parce qu'ils ne sont pas utilisés.

3.3.3.2 GDAE1D

- On a supprimé les fichiers :
  cas_qualif.c, *cas_qualif_2.c*, *jacobienne.{c , h}*, *Makefile_files*, *Makefile_time*, *sparse_jac.c* et *sparse_jac_comprow.c* parce qu'ils ne sont pas utilisés.
- On a fait les mêmes modifications dans *coupling* et *initialisation* que ceux effectuée dans les fichiers *Picard.c* et *initialisation.c* de SIA1D.

3.3.3.3 GDAE3D

On utilise MT3D pour définir la matrice de transport mais on utilise les sundials_sparse_row.
Puisqu'on a les fichiers binaires et les fichiers Ascii, on ajoute un argument (–binary) pour l'exécutable avec les fichiers binaires. Alors, on a modifié les fichiers *init.c* et *coupling.c* dans le répertoire Global1D2D3D. On a fait les modifications suivantes dans les fichiers *print.c* et *couplage.c* :

- On a changé le format d'affichage des variables dans la fonction sundials_info_out et sundials_print dans fichier *print.c*.
- On a ajouté sundials_info_out et sundials_print dans *coupling.c* pour l'affichage dans *global.out*.

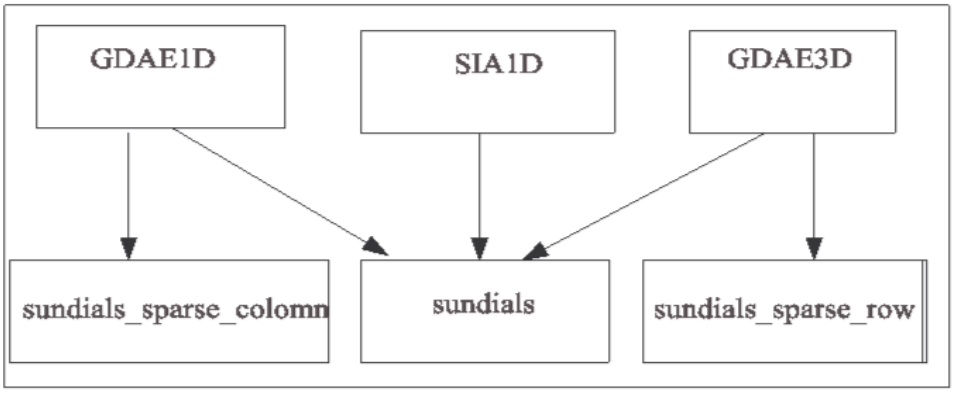


FIGURE 3.2 – Les points d'entrée et les sundials

3.4 Description version 2

Nous avons développé quatre nouveaux points d'entrée dans la suite GRT3D.
Le premier implémente la méthode séquentielle non itérative SNIA. Le deuxième, GRT3DRL, met en œuvre la réduction définie au chapitre 2. Le troisième, GRT3DRSL, n'utilise pas les logarithmes. Le quatrième, GRT3DRSLC, utilise le modèle général de précipitation.

3.4.1 GRT3DRL

Dans cette version, on utilise la forme réduite du système linéaire définie au chapitre 2, après élimination des variables T et C. Le couplage fait appel à de nouvelles fonctions qui calculent le problème réduit.
On a ajouté les fichiers suivants :

— reorder_bloc_11.c.h qui contient les deux fonctions suivants :

1. reorder_bloc_11_val pour calculer le premier bloc de la matrice Jacobienne suivant :
$$\mathbf{diag}(\frac{\partial T_c}{\partial c}) + \frac{\Delta t}{a_0}(M^{-1} \otimes I)\mathbf{diag}(\frac{\partial C_c}{\partial c}) \tag{3.1}$$
2. reorder_bloc_11_ind pour définir les indices de bloc défini dans l'équation (3.1) (Sparse_I et Sparse_J).

— SparseJss_Jac.c.h qui contient les fonctions suivantes :

1. SparseJss_Jac pour calculer la matrice Jacobienne réduite J_R.
2. SparseJss_Ind pour calculer les indices de la matrice Jacobienne J_R (Sparse_I et Sparse_J).
3. SparseJss_Fct pour calculer le second membre F_R du système réduit.
4. SparseJss_X pour calculer les autres inconnues à partit de δX : δT et δC_c

5. SparseJss_nz pour calculer le nombre des valeurs non nulles dans la matrice J_R.

De plus, nous avons créé la bibliothèque libsundials_sparse_reduction pour résoudre les systèmes linéaires. Donc, on remplace ida_sparse.{c, h} par ida_sparseRT.{c, h}, sundials_sparseTri.{c, h} par sundials_sparseRT.{c, h}. On définit les types et les fonctions suivantes :

- Type IDASparseRTJacFn : définit le type de la matrice Jacobienne réduite.
- Type IDASparseRTJacInd : définit le type de l'indice de la matrice Jacobienne réduite.
- Type IDASparseRTbssFn : définit le type du second membre réduit.
- Type IDASparseRTXFn : définit le type du vecteur global ($\delta Y = (\delta T_c, \delta X, \delta C_c)$).
- Fonction IDASparseRT : définit la fonction qui initialise les tailles des systèmes complet et réduit.
- Fonction IDASparseRTSetJacFn : fonction qui définit la matrice Jacobienne réduite.
- Fonction IDASparseRTSetbss : fonction qui définit le second membre.
- Fonction IDASparseRTSetXFn : fonction qui redéfinit le vecteur δY complète.
- Fonction IDASparseRTSetJacInd : fonction qui définit les indices de la matrice Jacobienne réduite.

3.4.2 GRT3DRSL

Dans cette version, on utilise toujours la forme réduite du système linéaire. Par contre, on utilise un nouveau module de chimie, où on n'applique pas les logarithmes. Nous avons donc développé un nouveau module pour calculer les lois de la chimie et leurs dérivées. De plus, nous avons ajouté les contraintes de positivité dans IDA. Nous avons ajouté des fonctions suivantes pour calculer les dérivées pour tous les points du maillage :

- get_global_dx pour calculer $\dfrac{\partial \alpha_k}{\partial c_j}$ dans le cas où $c_j = 0$ et $S_{kj} = 1$.
- get_global_dyc pour calculer $\dfrac{\partial \beta_k}{\partial c_j}$ dans le cas où $c_j = 0$ et $A_{kj} = 1$.
- get_global_dys pour calculer $\dfrac{\partial \beta_k}{\partial s_j}$ dans le cas où $s_j = 0$ et $B_{kj} = 1$.
- get_global_dep pour calculer $\dfrac{\partial \varphi}{\partial c}$ dans le cas où $c_j = 0$ et $E_{kj} = 1$.

3.4.3 SNIA

Dans cette version on met en œuvre la méthode SNIA définie dans le premier chapitre, on utilise le solveur KINSOL pour résoudre les problèmes de chimie :

$$\Phi(X_j^{n+1}) = (T_j^{n+1}, W_j^{n+1}, 1), \quad j = 1, \ldots, N_m$$

et les concentrations sont calculées sans le passage par le logarithme.
Dans cette version, on utilise Sundials standard sans modifications et on utilise MT3D comme la version GDAE3D pour calculer la matrice de transport.
On utilise les mêmes fonctions que celles définies dans la version GRT3DRSL pour calculer les lois de chimie et les dérivées des concentrations des espèces aqueuses, fixées, précipitées.
On appelle à chaque itération la fonction de l'équilibre pour calculer X à cette itération, après le calcul de T. Après avoir trouvé l'équilibre à chaque point du maillage, on appelle la fonction qui calcule C.

3.4.4 GRT3DRSLC

Dans cette version, on utilise le modèle général de précipitation-dissolution, avec la fonction min et la méthode de Newton semismooth. Nous avons développé de nouvelles fonctions pour les lois de précipitation-dissolution et leurs dérivées. De plus, nous devons ajouter des contraintes de positivité dans IDA. Nous avons ajouté la fonction suivante :

— dEp_cp pour calculer $\dfrac{\partial \varphi_{min}}{\partial p_j}$.

3.5 Tests

3.5.1 aqueux-adv (1D)

Dans ce cas, on a un seul composant aqueux donc pas de chimie. On considère un domaine 1D de taille 0.08m. Les conditions de transport sont les suivantes :
La porosité est égale à 1, l'advection vaut 2.77778×10^{-6}m/s et la dispersion est nulle.
La concentration initiale vaut 0.05 mol.l^{-1} et le temps final est 86400s.
La figure 3.3 représente le domaine 1D.

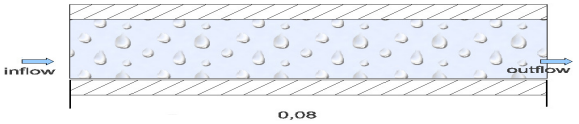


FIGURE 3.3 – Domaine 1D pour les cas tests aqueuse-adv et ex11-adv

La figure 3.4 montre les résultats de simulation à différents instants et différents pas de maillages.

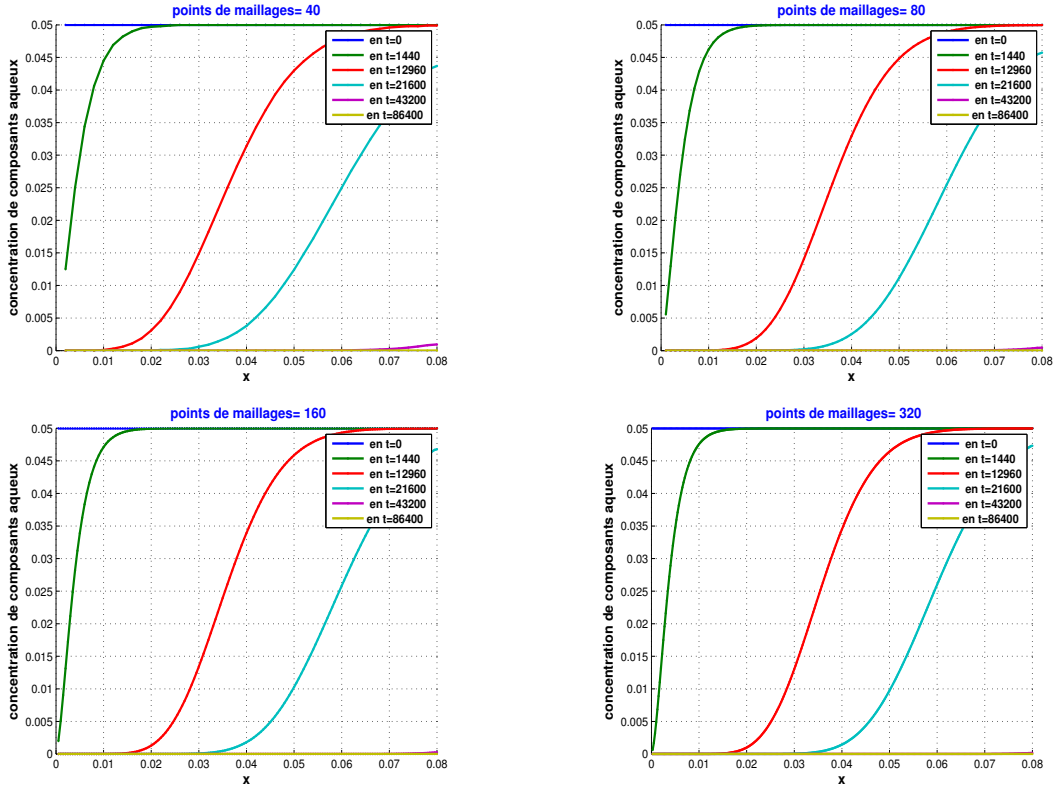


FIGURE 3.4 – Cas test aqueux-adv (1D) : concentrations des composants aqueux calculés avec SIA

On remarque que les résultats sont déjà précis avec le maillage grossier. Les calculs ont été effectués avec SIA1D. Des résultats similaires sont obtenus avec les autres points d'entrée.

3.5.2 ex11-adv (1D)

Ce cas test est défini dans PhreeqC [32]. Dans ce cas, on a quatre composants aqueux et un composant fixé ; toujours dans un domaine 1D de taille 0.08m. Les conditions de transport sont les mêmes que le cas test aqueux-adv (1D).
Les données de chimie et les conditions initiales sont indiquées dans les tableaux 3.2, 3.3 et 3.4.

	c_1	c_2	c_3	c_4	s
β_1	-2	1	0	0	2
β_2	-1	0	1	0	1

TABLE 3.2 – les coefficients stœchiométriques entre les espèces et les composants pour le cas test ex11-adv

	β_1	β_2
logK	8.654	1.6118

TABLE 3.3 – Logarithme de constante de réaction relatif aux espèces secondaires fixées

	c_1	c_2	c_3	c_4	s
T	4.49×10^{-4}	10^{-9}	7.5×10^{-4}	10^{-9}	1.1×10^{-3}

TABLE 3.4 – Concentrations initiales

On présente les résultats calculés avec GDAE1D.

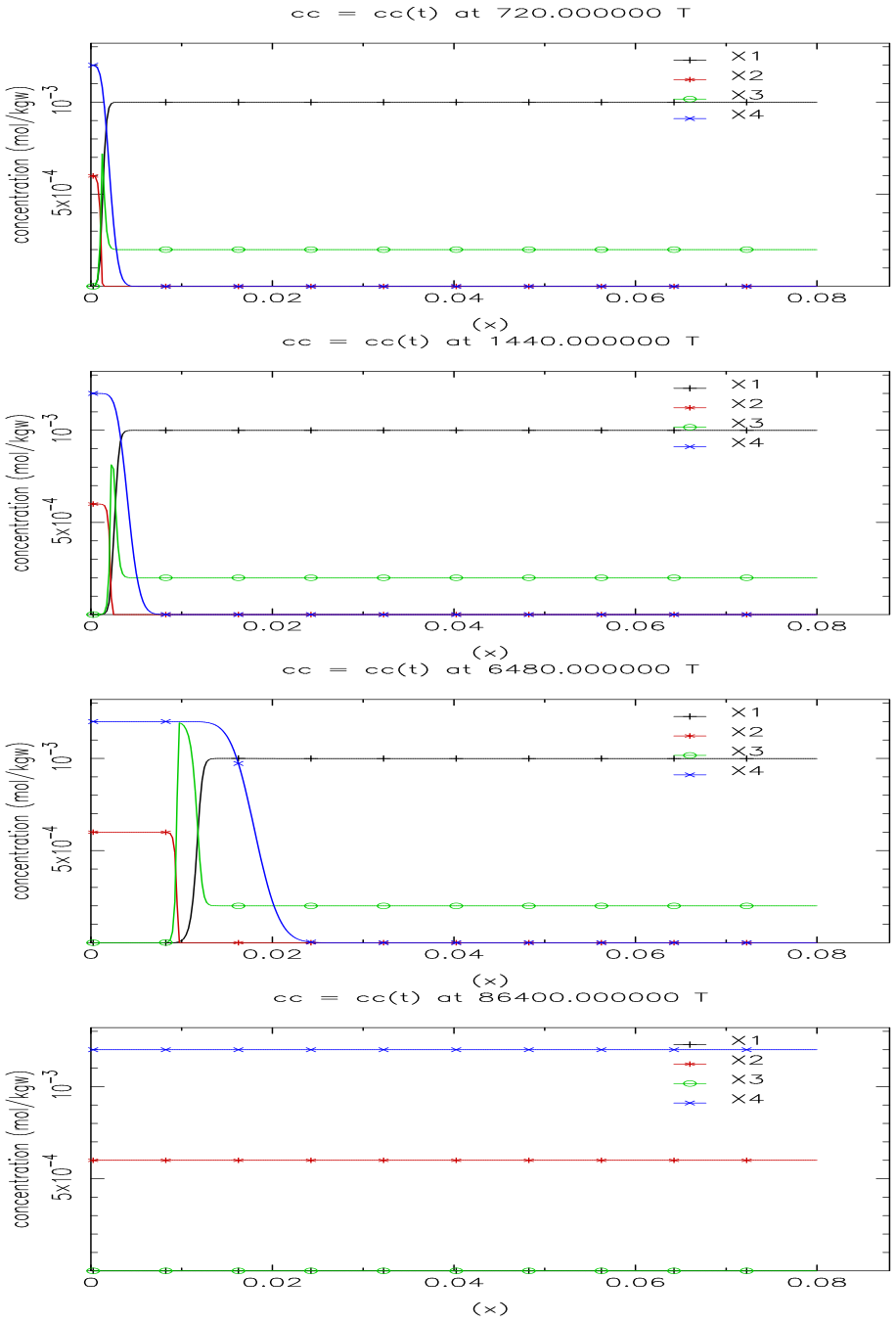


FIGURE 3.5 – Cas test Ex11-adv (1D) : Concentrations de chaque composant aqueux calculées avec GDAE1D

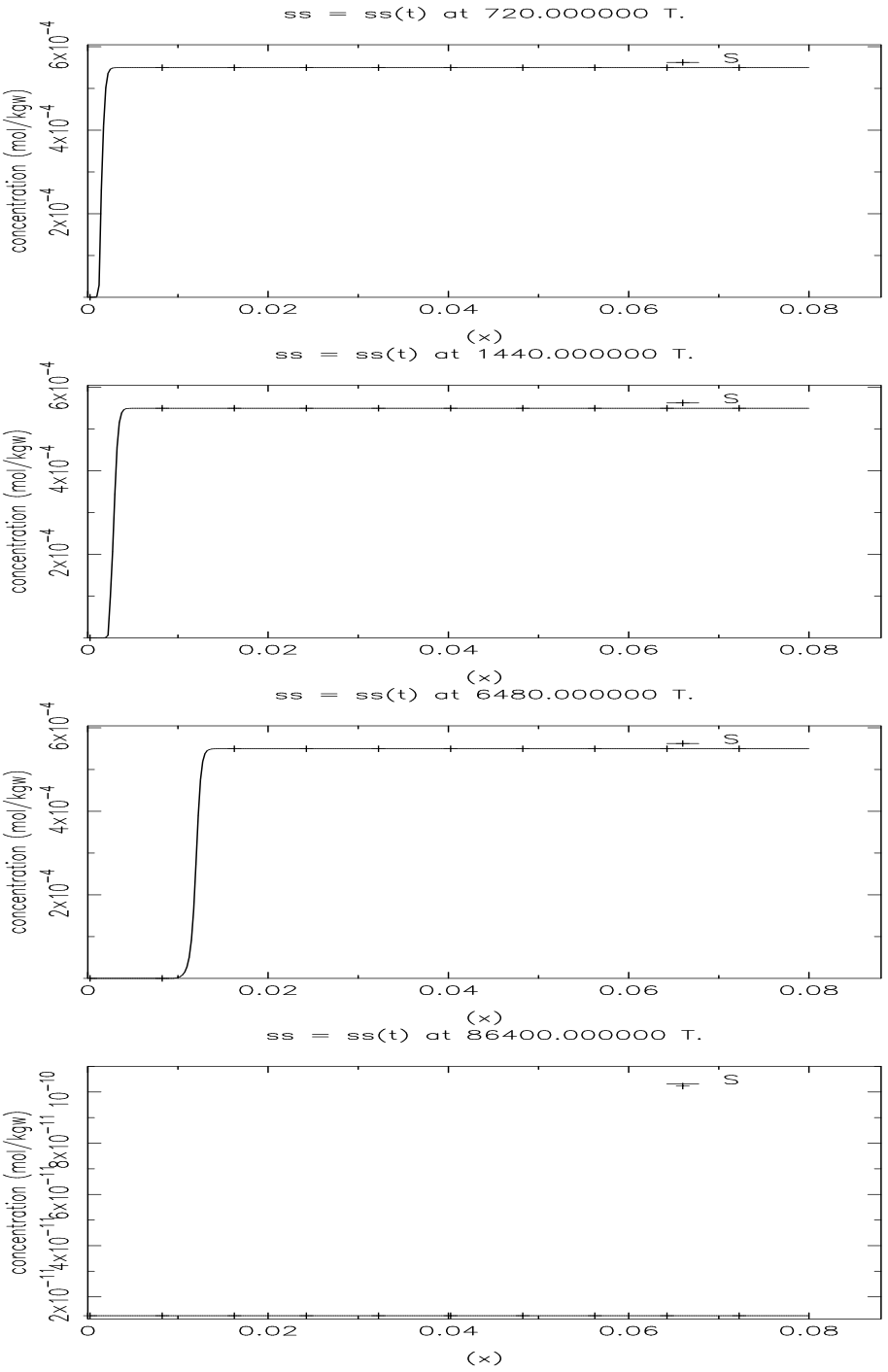


FIGURE 3.6 – Cas test Ex11-adv (1D) : Concentrations de composant fixé calculées avec GDAE1D

3.5.3 Benchmark-adv (1D)

Ce cas test est issu du Benchmark MoMas[4]. Dans ce cas, on a quatre composants aqueux et un composant fixé dans un domaine 1D avec de l'advection et de la dispersion.

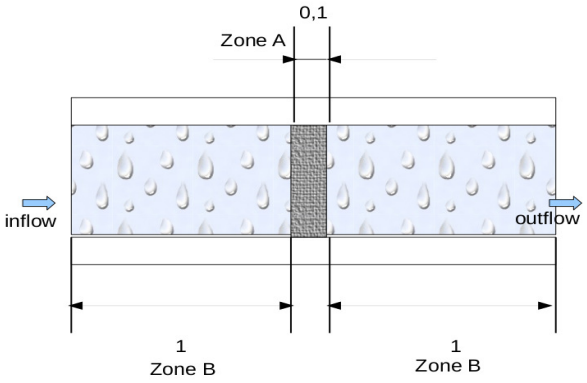


FIGURE 3.7 – Domaine 1D

On considère un domaine 1D, avec deux zones A et B : voir la figure 3.7. Les conditions de transport sont données par les tableaux 3.5 et 3.6. Le temps final des calculs est 6000s.
Le système chimique contient quatre composants aqueux et un composant fixé. Les données chimiques sont indiquées dans les tableaux 3.7 et 3.8.

	porosité	advection	dispersion
zone A	0.5	5.5×10^{-3}	3.3×10^{-4}
zone B	0.25	5.5×10^{-3}	5.5×10^{-5}

TABLE 3.5 – propriétés et caractéristiques de chaque zone

T	c_1	c_2	c_3	c_4	s
zone A	10^{-20}	-2	10^{-20}	2	10
zone B	10^{-20}	-2	10^{-20}	2	1

TABLE 3.6 – Concentrations initiales des zones A et B

	c_1	c_2	c_3	c_4	s
α_1	0	-1	0	0	0
α_2	0	1	1	0	0
α_3	0	-1	0	1	0
α_4	0	-4	1	3	0
α_5	0	4	3	1	0
β_1	0	3	1	0	1
β_2	0	-3	0	1	2

TABLE 3.7 – les coefficients stœchiométriques entre les espèces et les composants dans le test Benchmark-adv

	α_1	α_2	α_3	α_4	α_5	β_1	β_2
logK	-27.63	0	0	-2.3052	80.59048	13.8155	-2.3025

TABLE 3.8 – Logarithme de constante de réaction relatif aux espèces secondaires aqueuses et fixées

On présente dans le tableau 3.9 la différence entre les calculs de GDAE3D et GDAE1D, on remarque que la différence est négligeable.

$$erreur = \frac{\sum_{i=1}^{Nm}(c_{\text{GDAE3D}}(i)) - c_{\text{GDAE1D}}(i))^2}{\sum_{i=1}^{Nm}(c_{\text{GDAE3D}}(i))^2}$$

temps	c_1	c_2	c_3	c_4	s
t=10s	1.850153E-13	0	7.735858E-21	0	0
t=1000s	0	0	0	0	0
t=2000s	0	0	0	0	0
t=6000s	0	0	0	0	0

TABLE 3.9 – L'erreur relative entre les résultats de GDAE3D et GDAE1D pour chaque composant.

On présente les résultats calculés avec GDAE1D et GDAE3D.

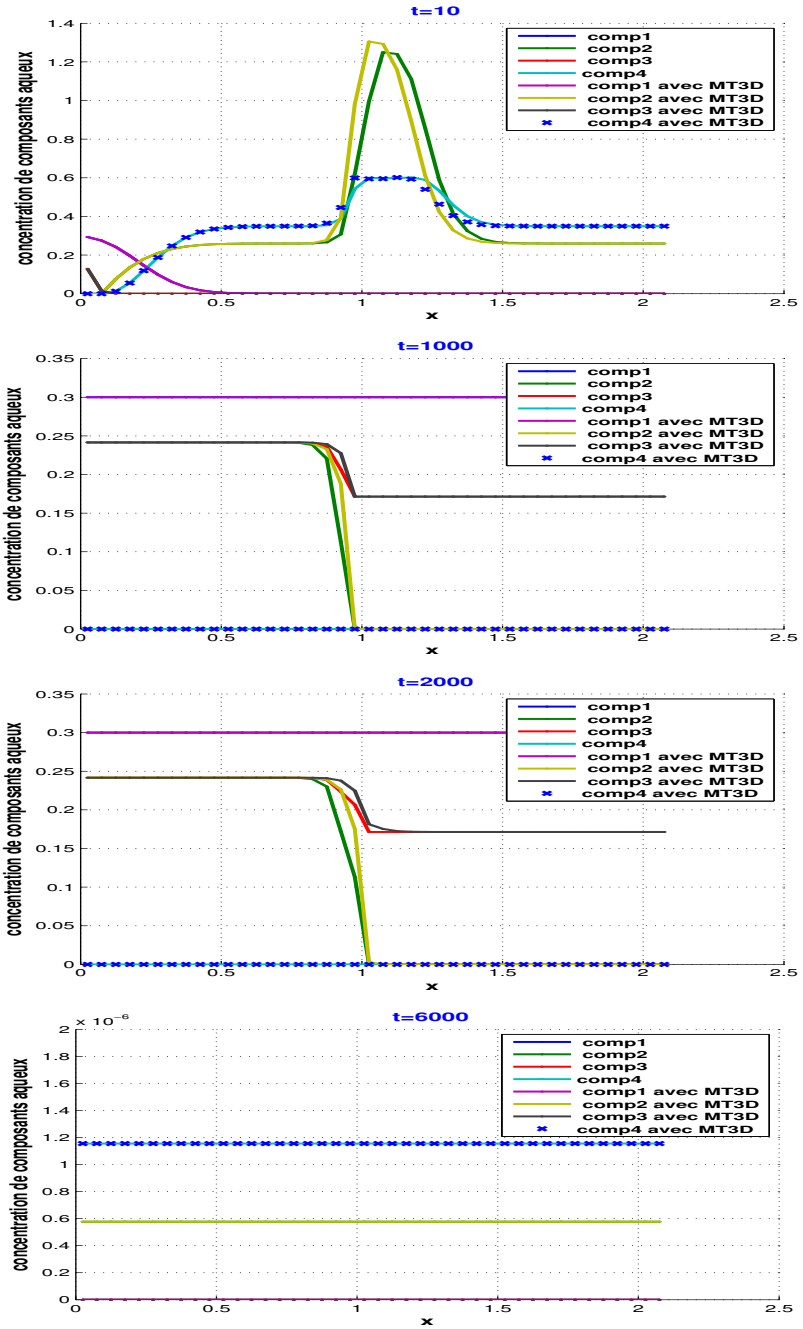


FIGURE 3.8 – Cas Benchmark-adv (1D) : Concentrations de chaque composant aqueux calculées avec GDAE1D et GDAE3D

Au cours du temps, on remarque que GDAE1D et GDAE3D en 1D sont semblables.

3.5.4 Benchmark-disp (1D)

Le domaine et le système chimique sont les mêmes que dans Benchmark-adv (1D). Les conditions de transport sont modifiées, comme indiqué dans le tableau 3.10

	porosité	advection	dispersion
zone A	0.5	5.5e-3	3.3e-1
zone B	0.25	5.5e-3	5.5e-02

TABLE 3.10 – propriétés et caractéristiques de chaque zone dans cas Benchmark-disp

La figure 3.9 présente les concentrations initiales.

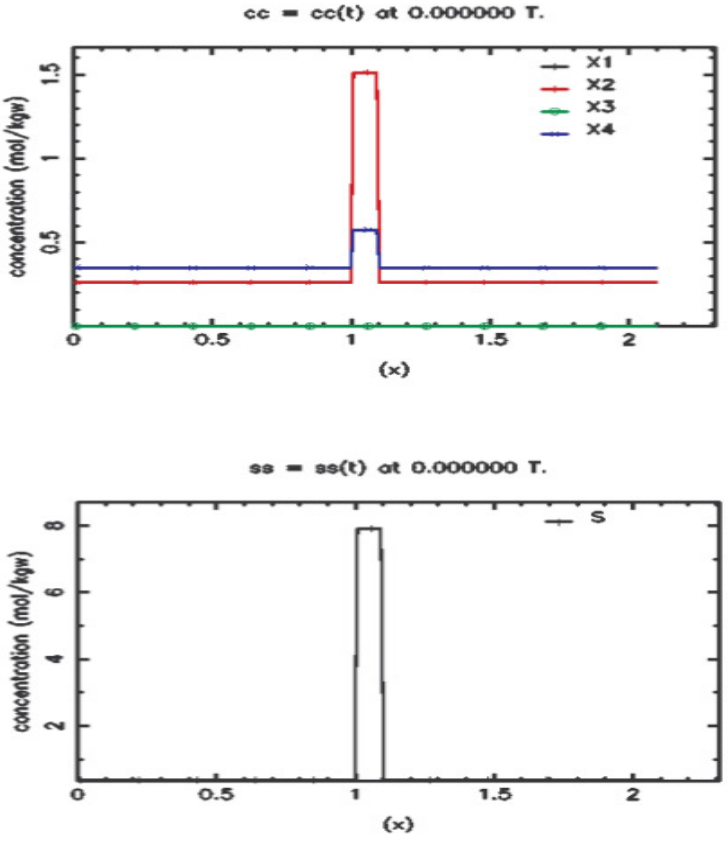


FIGURE 3.9 – Concentrations initiales de chaque composant

Les figures 3.10 et 3.11 présentent les résultats calculés avec GDAE1D pour un maillage de 200 points.

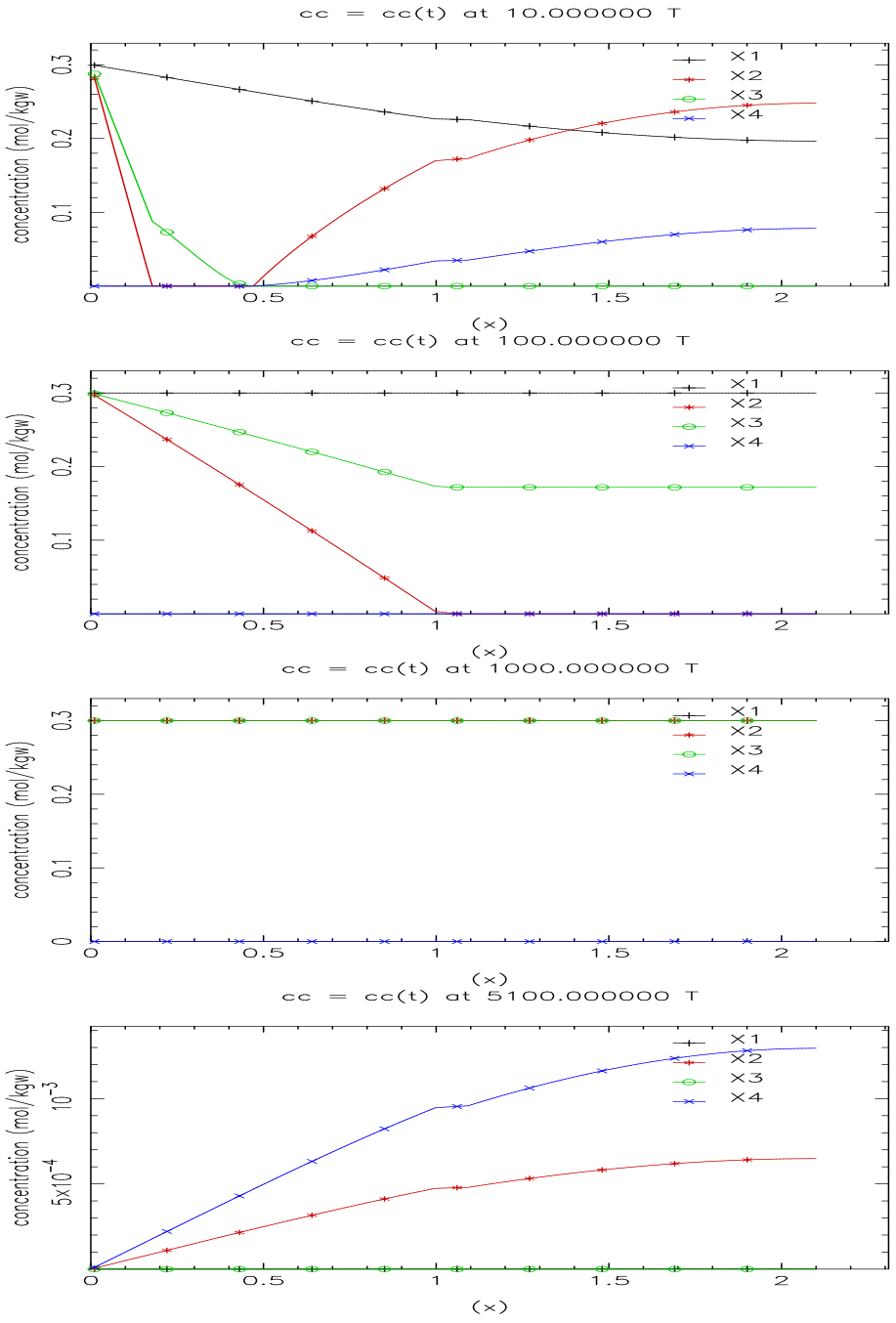


FIGURE 3.10 – Concentrations de chaque composant aqueux calculées avec GDAE1D

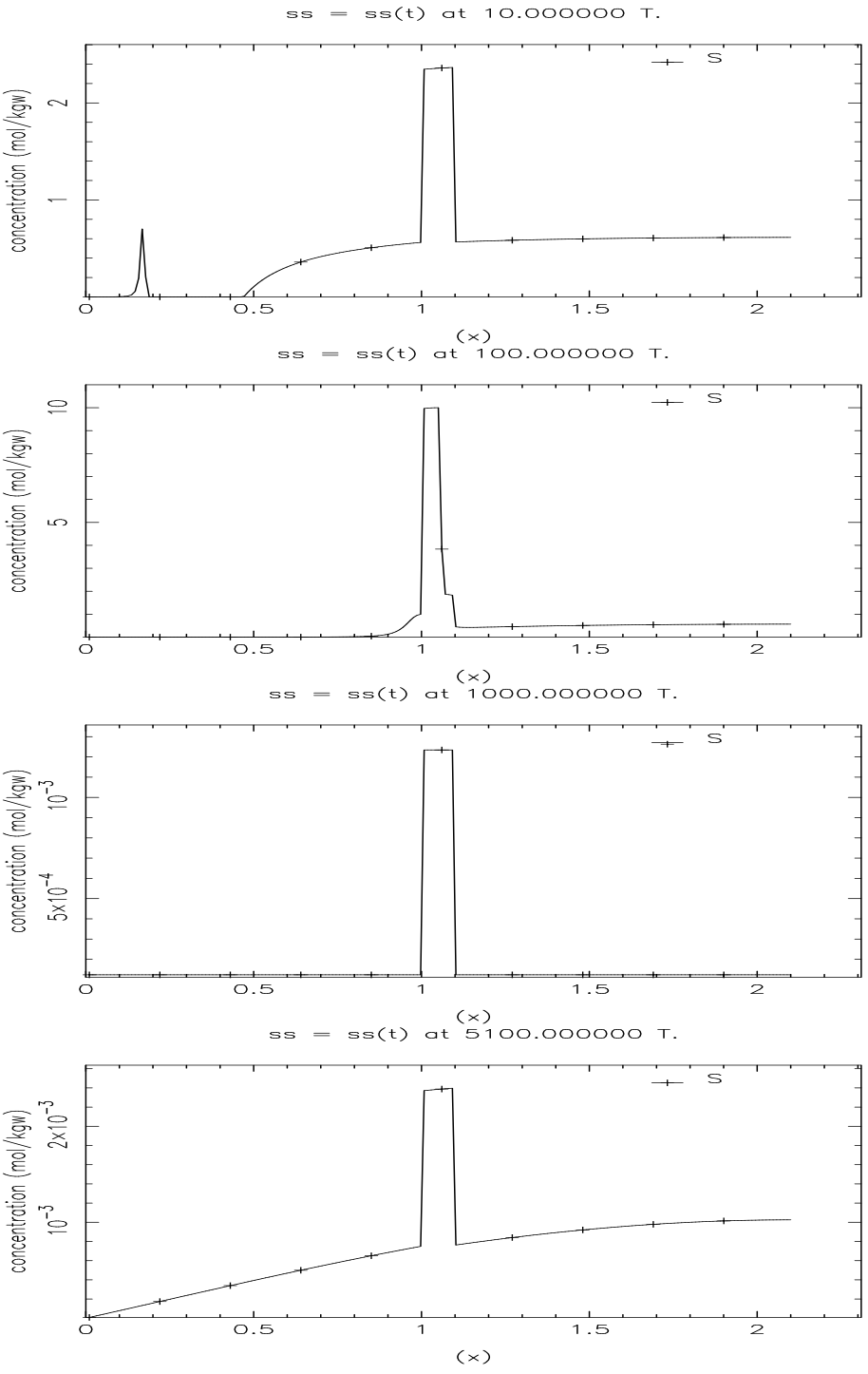


FIGURE 3.11 – Concentrations de chaque composant fixé calculées avec GDAE1D

3.6 Conclusion

Nous avons présenté les modifications faites dans le logiciel GRT3D. Nous avons remarqué dans les cas tests présentés qu'il n'y a pas une grande différence entre les calculs avec SIA ou GDAE1D ou GAE3D.
Nous avons aussi présenté les nouvelles versions de notre logiciel :

- La version GRT3DRL qui réduit la taille de système résolu à chaque pas de temps.
- La version GRT3DRSL qui réduit la taille de système résolu à chaque pas de temps et sans le passage par les logarithmes des concentrations.
- La version SNIA qui implémente la méthode explicite SNIA.

Nous présentons dans les chapitres suivants une comparaison entre ces trois versions de GRT3D et SNIA pour les deux cas tests suivants :

- Cas test ANDRA en 2D.
- Cas Benchmark en 2D.

Chapitre 4

Cas test Andra 2D

Nous avons présenté dans le chapitre précédent les logiciels GRT3D, GRT3DRL, GRT3DRSL et SNIA. Dans ce chapitre, nous validons ces logiciels pour un cas test fourni par l'ANDRA (Agence Nationale pour la gestion des Déchets RAdioactifs). Plus précisément, nous utilisons le second cas test du plan de qualification du projet Alliances (Alliances 6.2) [30, 12, 38]. Il simule l'injection de l'eau alcaline (NaOH) dans un milieu poreux 2D contenant des grains de quartz (SiO2).
Dans ce cas, nous connaissons la solution analytique de ce problème, ce qui permet de mesurer la précision des résultats des quatre logiciels. Nous comparons aussi le temps de calcul et la quantité de mémoire utilisée.

4.1 Description

Dans ce cas test ANDRA, on simule l'injection de l'eau alcaline (NaOH) dans un milieu poreux 2D contenant des grains de quartz (SiO2). Le domaine est rectangulaire, d'une longueur de 5 m et une largeur de 3,5 m et cette injection conduit aux réactions acido-basiques et de dissolution du quartz suivantes :

$$\begin{array}{lcll} H_2O & \leftrightarrow & H^+ & + \quad OH^- \\ H_4SiO_4 & \leftrightarrow & H_3SiO_4^- & + \quad H^+ \\ H_4SiO_4 & \leftrightarrow & SiO_2(s) & + \quad 2H_2O \end{array} \tag{4.1}$$

avec

$$\begin{array}{lcll} K_1 & = & \alpha_{H^+}c_{OH^-} & = \quad 10^{-14} \\ K_2 & = & \dfrac{\alpha_{H_3SiO_4^-}\alpha_{H^+}}{c_{H_4SiO_4}} & = \quad 10^{-9.8} \\ K_3^{-1} & = & c_{H_4SiO_4} & = \quad 10^{-3.6} \end{array} \tag{4.2}$$

où $c_{H_4SiO_4}$ est la concentration de composant aqueux l'acide silicique (H_4SiO_4), c_{OH^-} est la concentration de composant aqueux hydroxyde(OH^-), α_{H^+} est la concentration de l'espèce secondaire aqueuse hydrogène (H^+) et $\alpha_{H_3SiO_4^-}$ est la concentration de l'espèce secondaire aqueuse $H_3SiO_4^-$.
Nous avons aussi la réaction de neutralité

$$NaOH \quad \leftrightarrow \quad Na^+ \quad + \quad OH^-$$

Dans ce système nous avons choisi trois composants aqueux ($N_c = 3$) : Na^+, OH^- et H_4SiO_4, ainsi que deux espèces secondaires aqueuses ($N_\alpha = 2$) :H^+ et $H_3SiO_4^-$ et une espèce précipitée ($N_p = 1$) : le quartz (SiO_2). Ici le sodium Na^+ est un traceur puisqu'il ne réagit pas chimiquement avec les autres espèces.

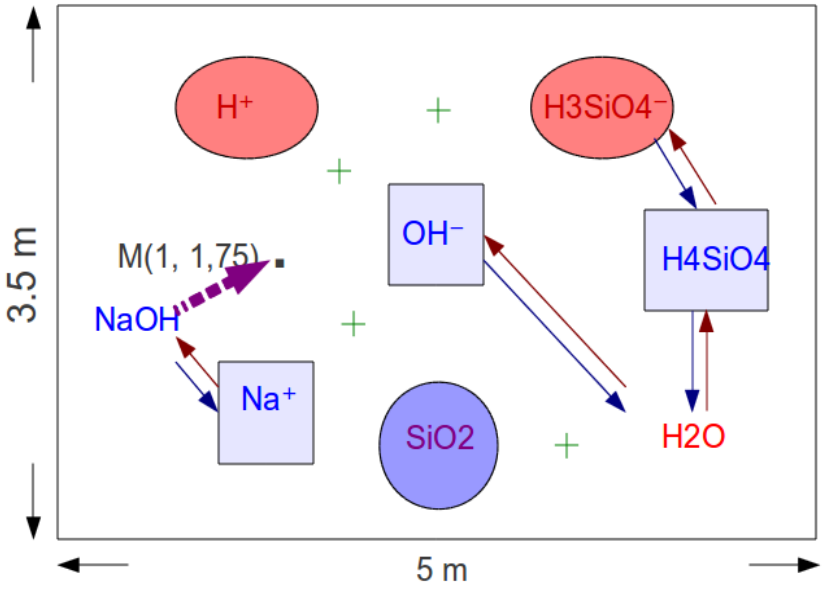


FIGURE 4.1 – Domaine 2D du test ANDRA et système chimique

On suppose que le nombre de moles de quartz reste strictement positif pendant toute la simulation dans chaque maille, de sorte que le seuil de saturation est toujours atteint. la loi d'action de masse associée à l'équation de précipitation est ainsi toujours vérifiée.

On définit les concentrations totales T par les équations suivantes :

$$\begin{aligned} T_{Na^+} &= c_{Na^+} \\ T_{OH^-} &= c_{OH^-} - \alpha_{H^+} + \alpha_{H_3SiO_4^-} \\ T_{H_4SiO_4} &= c_{H_4SiO_4} + \alpha_{H_3SiO_4^-} + p_{SiO_2} \end{aligned} \tag{4.3}$$

avec p_{SiO_2} est la quantité de l'espèce précipitée SiO_2.
Et on a aussi :

$$\begin{aligned} T_{Na^+} &= C_{Na^+} \\ T_{H_4SiO_4} &= C_{H_4SiO_4} + p_{SiO2} \\ T_{OH^-} &= C_{OH^-} \end{aligned} \tag{4.4}$$

C_{Na^+} est la partie aqueuse de la concentration totale T_{Na^+}, C_{OH^-} est la partie aqueuse de la concentration totale T_{OH^-} et $C_{H_4SiO_4}$ est la partie aqueuse de la concentration

totale $T_{H_4SiO_4}$. Puisque $c_{H_4SiO_4} = \dfrac{1}{K_3}$, on peut éliminer l'espèce H_4SiO_4 parce qu'elle est constante à chaque maille et à chaque instant.
En plus, on peut traduire la loi de neutralité électrique sous la forme suivante :

$$C_{Na^+} = C_{OH^-}$$

Ce qui s'écrit, à l'aide des équations (4.3) et (4.4) :

$$c_{Na^+} - c_{OH^-} + \alpha_{H^+} - \alpha_{H_3SiO_4^-} = 0 \tag{4.5}$$

D'autre part :

$$T_{Na^+} = C_{Na^+} = C_{OH^-} = T_{OH^-} = c_{Na^+}$$

Avec les équations (4.2), on obtient

$$\begin{cases} c_{OH^-} & = & \dfrac{K_1}{\alpha_{H^+}} \\ \alpha_{H_3SiO_4^-} & = & \dfrac{K_2}{K_3\alpha_{H^+}} \end{cases} \tag{4.6}$$

En reportant dans l'équation (4.5), on a une équation du second degré en α_{H^+} :

$$\alpha_{H^+}^2 + (C_{OH^-})\alpha_{H^+} - (K_1 + \frac{K_2}{K_3}) = 0$$

Ainsi, dès que la concentration totale C_{OH^-} est connue, on peut résoudre cette équation d'ordre 2. Nous considérons un transport par advection et dispersion, avec une vitesse d'écoulement constante et des coefficients de dispersion constants donnés dans le tableau 4.1. La durée de simulation est de 30 jours avec des conditions aux limites de flux nul (conditions de Neumann homogènes) imposées partout à la frontière.

Les données	valeur
porosité	1.
vitesse du flux longitudinal v_1	$5.7 \times 10^{-7}\ m.s^{-1}$
vitesse du flux transversal v_2	$0\ m.s^{-1}$
le coefficient de diffusion moléculaire d_m	0
le coefficient de dispersion longitudinal α_L	0.2 m
le coefficient de dispersion transversal α_T	0.05 m
la durée de simulation	30 jours
Condition au bord	flux nul

TABLE 4.1 – Les conditions de transport

Le système de transport s'écrit sous la forme suivante :

$$\begin{cases} \dfrac{\partial T_{Na^+}}{\partial t} + v_1 \dfrac{\partial C_{Na^+}}{\partial x} - v_1\alpha_L \dfrac{\partial^2 C_{Na^+}}{\partial x^2} - v_1\alpha_T \dfrac{\partial^2 C_{Na^+}}{\partial y^2} & = & 0 \\ \dfrac{\partial T_{OH^-}}{\partial t} + v_1 \dfrac{\partial C_{OH^-}}{\partial x} - v_1\alpha_L \dfrac{\partial^2 C_{OH^-}}{\partial x^2} - v_1\alpha_T \dfrac{\partial^2 C_{OH^-}}{\partial y^2} & = & 0 \\ \dfrac{\partial T_{H_4SiO_4}}{\partial t} + v_1 \dfrac{\partial C_{H_4SiO_4}}{\partial x} - v_1\alpha_L \dfrac{\partial^2 C_{H_4SiO_4}}{\partial x^2} - v_1\alpha_T \dfrac{\partial^2 C_{H_4SiO_4}}{\partial y^2} & = & 0 \\ + \text{ conditions initiales} \\ + \text{ conditions aux limites} \end{cases} \tag{4.7}$$

4.1.1 Les conditions initiales

On suppose que le domaine contient 10 moles de quartz par litre et ne contient pas de sodium et on injecte de l'eau d'alcaline (NaOH) au point $M(1, 1.75)$.
Le tableau 4.2 résume les conditions initiales suivantes :

Condition	Na^+	OH^-	H_4SiO_4
La valeur initiale de T partout sauf au point M	0	0	10
La valeur initiale de T au point M	10^{-2}	10^{-2}	10

TABLE 4.2 – Les conditions initiales

4.2 La solution analytique

Nous avons vu que :

1. $c_{H_4SiO_4} = \dfrac{1}{K_3}$ donc la concentration du composant aqueux H_4SiO_4 est fixée pour chaque maille et pour chaque instant.
2. α_{H^+} vérifié l'équation suivante :

$$\alpha_{H^+}^2 + (C_{OH^-})\alpha_{H^+} - (K_1 + \frac{K_2}{K_3}) = 0,$$

avec les deux solutions suivantes :

$$\alpha_{H^+} = \begin{cases} -\dfrac{1}{2}(C_{OH^-} + \sqrt{C_{OH^-}^2 + 4(K_1 + \dfrac{K_2}{K_3})}) \\ \text{ou} \\ \dfrac{1}{2}(-C_{OH^-} + \sqrt{C_{OH^-}^2 + 4(K_1 + \dfrac{K_2}{K_3})}) \end{cases} \tag{4.8}$$

La première solution est négative. Par ailleurs, puisque $(K_1 + \dfrac{K_2}{K_3}) \geq 0$, la deuxième solution est positive.

3. Puisque $T_{Na^+} = C_{Na^+} = C_{OH^-} = T_{OH^-} = c_{Na^+}$, il suffit de calculer C_{Na^+}.
4. L'équation appliquée à Na^+ s'écrit sous la forme suivante :

$$\frac{\partial C_{Na^+}}{\partial t} + v_1 \frac{\partial C_{Na^+}}{\partial x} - v_1 \alpha_L \frac{\partial^2 C_{Na^+}}{\partial^2 x} - v_1 \alpha_T \frac{\partial^2 C_{Na^+}}{\partial^2 y} = 0$$

Et la condition initiale pour l'injection au point M à $x_0 = 1$ et $y_0 = 1.75$ peut s'écrire sous cette forme :

$$C_{Na^+}(x, y, t = 0) = C_0 \delta(x - x_0)\delta(y - y_0)$$

avec $C_0 = 10^{-2}$. Si le domaine est infini, il est possible de calculer une solution analytique :

$$C_{Na^+}(x, y, t) = \frac{C_0}{4\pi v_1 \sqrt{\alpha_L \alpha_T}} \exp(-\frac{(x - x_0 - v_1 t)^2}{4 v_1 \alpha_L t} - \frac{(y - y_0)^2}{4 v_1 \alpha_T t})$$

En effet, c'est une solution fondamentale 2D de l'équation de diffusion que l'on translate suivant x avec la vitesse (connue et constante) de l'écoulement (cette solution, valable pour un domaine infini est seulement "approchée" pour notre domaine fini).

4.3 Comparaison des trois versions de GRT3D

Le domaine rectangulaire est maillé avec une maillage régulier de $n_1 \times n_2$ cellules. Nous utilisons six maillages différents, allant d'un maillage grossier de 21×42 cellules à un maillage fin de 322×224 cellules.

Nous utilisons le modèle de transport réactif sans exploiter l'invariance de $c_{H_4SiO_4}$ ni la neutralité électrique, cette approche nous permet de vérifier la précision des résultats.

4.3.1 Précision

Il est facile de comparer la valeur calculée de $c_{H_4SiO_4}$ avec la valeur exacte de $\frac{1}{K_3}$. Pour les trois versions et toutes les tailles de mailles, nous obtenons une erreur de l'ordre de 10^{-15}.

Puis nous comparons les résultats des différentes versions avec la solution analytique pour le composant Na^+, ensuite pour l'espèce secondaire aqueuse H^+ .

Comme nous avons défini les trois versions de GRT3D dans le chapitre 3, les conditions initiales sont les mêmes pour GRT3D et GRT3DRL mais diffèrent par rapport à GRT3DRSL parce que, dans les deux premiers, on remplace la valeur 0 par 10^{-10} à cause de la forme logarithmique.

Pour cette approche globale DAE, les pas temps internes sont choisis dynamiquement par le solveur IDA et dans ce cas $\Delta t_e = 12960s$ pour tous les maillages.

Pour chaque simulation, nous avons un nombre de pas de temps externe N_t pour les sorties.

On note $\hat{c}_{Na^+}(m_j, t_n)$ les concentrations calculées de Na^+ aux différents points $m_j, j = 1, ..., N_m$ et à divers instants $t_n, n = 1, ..., N_t$ pour chaque version de GRT3D, et $c_{Na^+}(m_j, t_n)$ la solution analytique à chaque point m_j et instant t_n.
Pour chaque version de GRT3D sans la chimie (parce que le Na^+ est un traceur), on calcule l'erreur E_{Na^+} définie sous cette forme :

$$E_{Na^+} = \sqrt{\frac{1}{N_m N_t} \sum_{n,j} (\hat{c}_{Na^+}(m_j, t_n) - c_{Na^+}(m_j, t_n))^2}$$

Maillage	GRT3D	GRT3DRL	GRT3DRSL
21×14	2.719521E-04	2.719521E-04	2.719521E-04
41×28	2.084597E-05	2.084597E-05	2.084597E-05
81×56	1.351828E-06	1.351828E-06	1.351829E-06
161×112	2.500897E-07	2.500848E-07	2.500904E-07
322×224	-	1.642168E-08	1.642463E-08

TABLE 4.3 – L'erreur pour le composant Na^+ du test Andra 2D (seulement l'équation de transport)

Dans le tableau 4.3, nous remarquons que pour chaque maillage, l'erreur est la même pour les trois versions de GRT3D. En effet le sodium Na^+ est un traceur, qui ne réagit pas chimiquement avec les autres espèces. Alors, les trois logiciels traitent seulement l'équation de transport, qui est indépendante de l'utilisation de logarithmes ou non.
Pour chaque version de GRT3D, on peut faire le calcul avec le traceur ou sans le traceur.
On note aussi $\hat{\alpha}_{H^+}(m_j, t_n)$ les concentrations calculées de H^+ aux différents points $m_j, j = 1, ..., N_m$ et à divers instants $t_n, n = 1, ..., N_t$ sans traceur.
α_{H^+} est calculée à partir de l'équation (4.8) et la solution analytique c_{Na^+} aux différents points $m_j, j = 1, ..., N_m$ et à divers instants $t_n, n = 1, ..., N_t$.
On calcule l'erreur suivante :

$$E_{H^+} = \sqrt{\frac{1}{N_m N_t} \sum_{n,j} (\hat{\alpha}_{H^+}(m_j, t_n) - \alpha_{H^+}(m_j, t_n))^2}$$

Mesh	GRT3D	GRT3DRL	GRT3DRSL
21×14	4.976852E-08	4.976852E-08	4.976261E-08
41×28	2.111379E-08	2.111379E-08	2.110677E-08
81×56	8.703793E-09	8.703793E-09	8.697009E-09
161×112	3.264638E-09	3.264638E-09	3.257945E-09
322×224	-	9.297617E-10	9.234054E-10

TABLE 4.4 – L'erreur pour l'espèce H^+ dans le cas test Andra 2D

Avec le maillage le plus fin 322×224, les équations de chimie ne peuvent pas être résolues avec le logiciel GRT3D, sans doute parce que la matrice Jacobienne est mal conditionnée.
Dans le tableau 4.4, on remarque qu'il n'y a pas une grande différence entre les résultats des trois version de GRT3D et on remarque aussi la décroissance de l'erreur avec le raffinement du maillage, qui nous confirme la convergence de la méthode GDAE dans les trois versions.

4.3.2 Temps de calcul

Nous donnons le temps CPU pour les trois versions et pour différents maillages.
La table 4.5 présente le temps CPU pour les trois versions avec le traceur, c'est-à-dire que Na^+ est un composant du système chimique :

maillage	GRT3D		GRT3DRL		GRT3DRSL
	taille du système	temps CPU	taille du système	temps CPU	temps CPU
21×14	2940	1 min 50 s	1176	14 s	6 s
41×28	13440	8 min	5376	1 min 25 s	23 s
81×56	45360	56 min	18144	11 min 5 s	2 min 14 s
161×112	180320	3 h	72128	1 h 31 min	20 min 27 s
322×224	721280	-	288512	9 h 3 min	1 h 34 min

TABLE 4.5 – Taille du système et temps CPU pour le test Andra 2D avec le traceur

La table 4.6 présente le temps CPU pour les trois versions sans le traceur, c'est-à-dire que Na^+ n'est pas un composant du système chimique :

maillage	GRT3D		GRT3DRL		GRT3DRSL
	taille du système	temps CPU	taille du système	temps CPU	temps CPU
21×14	2058	18 s	882	4s	2s
41×28	9408	1 min 36 s	4032	21 s	8s
81×56	31752	6 min 33 s	13608	1 min 53 s	50 s
161×112	126224	32 min 43 s	54096	16 min 30 s	4 min 32 s
322×224	504896	-	216384	1 h 52 min	37 min 38 s

TABLE 4.6 – Taille du système et temps CPU pour le test Andra 2D sans le traceur

Nous constatons trois sources de réduction du temps de calcul :
- enlever le traceur.
- éliminer les inconnues T et C du système linéaire.

— ne pas utiliser les logarithmes.

Enlever le traceur réduit la taille du système, donc le temps de calcul.
Éliminer des inconnues réduit aussi la taille du système, donc le temps de calcul.
La taille de système est réduite pour les deux logiciels GRT3DRL et GRT3DRSL parce qu'on a réduit la taille de la matrice Jacobienne à chaque itération de Newton.
En effet, la taille de système est $(3N_c + N_s + N_p) \times N_m$ dans GRT3D et est réduite à $(N_c + N_s + N_p) \times N_m$. On a $N_c = 3$ dans le cas avec traceur, $N_c = 2$ dans le cas sans traceur, $N_s = 0$, $N_p = 1$.
L'accroissement des performances est aussi dûe, dans les trois techniques, à un meilleur conditionnement du Jacobien.
Les figures 4.2 et 4.3 montrent les conditionnements, avec ou sans traceur, à différents instants :

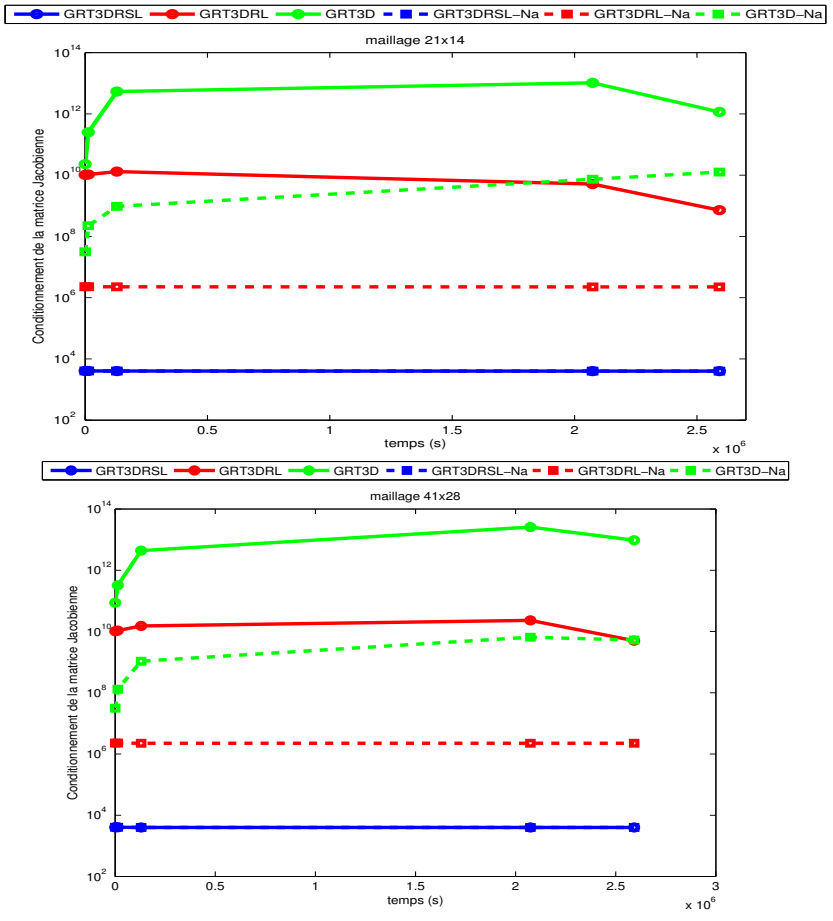


FIGURE 4.2 – Conditionnement de la Jacobienne pour les maillages 21×14, 41×28 pour chaque logiciel à plusieurs instants

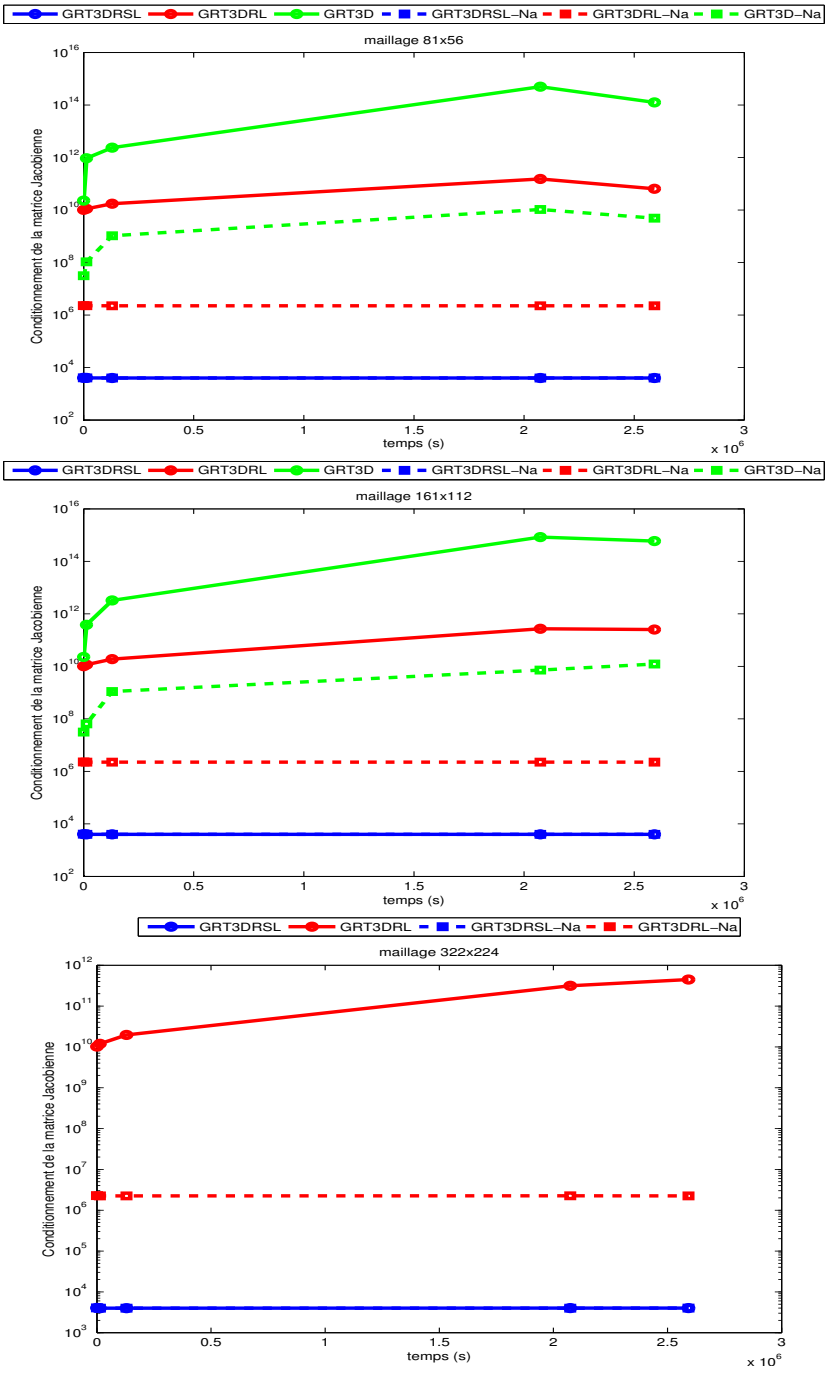


FIGURE 4.3 – Conditionnement de la Jacobienne pour les maillages 81×56, 161×112 et 322×224 pour chaque logiciel à plusieurs instants

On constate qu'enlever le traceur réduit nettement le conditionnement dans GRT3D et GRT3DRL. Par contre, cela n'a pas d'impact sur le conditionnement dans GRT3DRSL. Ce résultat est conforme à l'analyse du chapitre 1, où nous montrons qu'avec un traceur, le Jacobien est singulier avec les logarithmes mais a priori bien conditionné sans les logarithmes.
Ainsi, le gain en enlevant le traceur est double : réduction de la taille du système et meilleur conditionnement.
On constate aussi qu'éliminer les inconnues T et C réduit fortement le conditionnement. Là aussi, le gain est double.
Enfin, ne pas utiliser les logarithmes réduit encore le conditionnement. Ce résultat est conforme à l'analyse du chapitre 1, où nous montrons que de faibles concentrations produisent un Jacobien proche de la singularité avec les logarithmes. Ici, la concentration c_{OH^-} est faible dans une grande partie du domaine.
Cette fois le gain est dû uniquement à cette amélioration, puisque la taille du système est la même avec ou sans logarithme.

4.4 La solution

La figure 4.4 montre les concentrations à $t = 0$ avec le maillage le plus fin de 322×224 et avec GRT3DRSL,

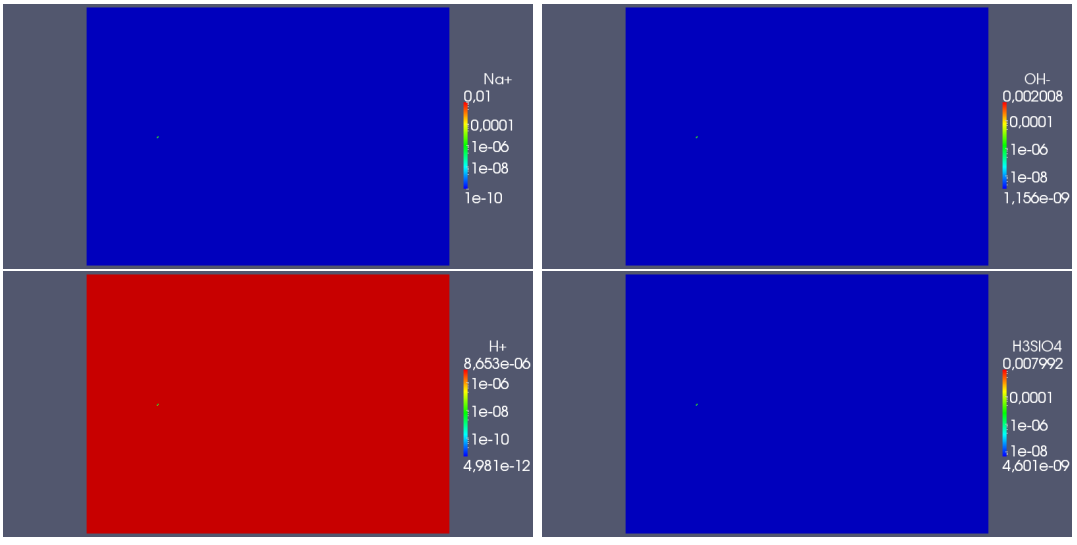


FIGURE 4.4 – Les concentrations à l'état initial pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$

Nous pouvons calculer le pH de la solution : par définition, $pH = -\log_{10}(\alpha_{H^+})$. Au point d'injection M, $pH = 11.3$ donc la solution est basique et dans le reste de domaine $pH = 5.06$ donc la solution est acide.

Les figures 4.5 à 5.10 montrent les concentrations après l'injection à différents instants.

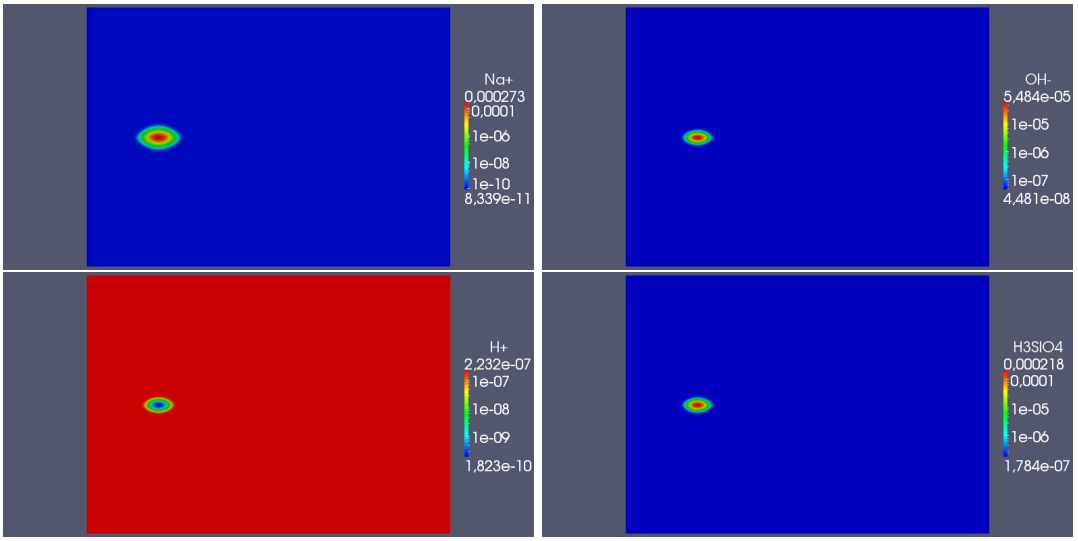


FIGURE 4.5 – Les concentrations à l'instant t=3h 36m pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$

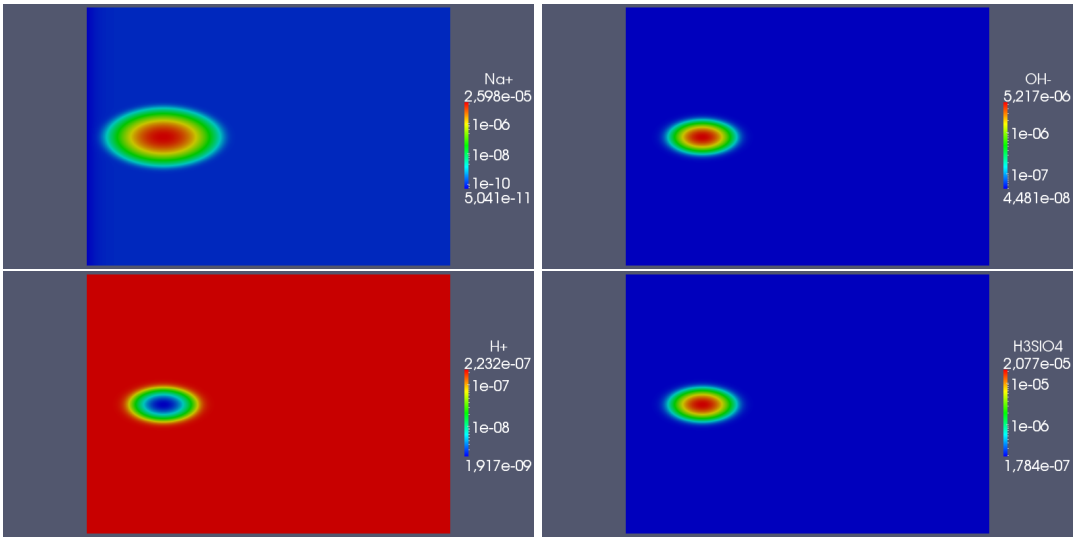


FIGURE 4.6 – Les concentrations à l'instant t=36h pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$

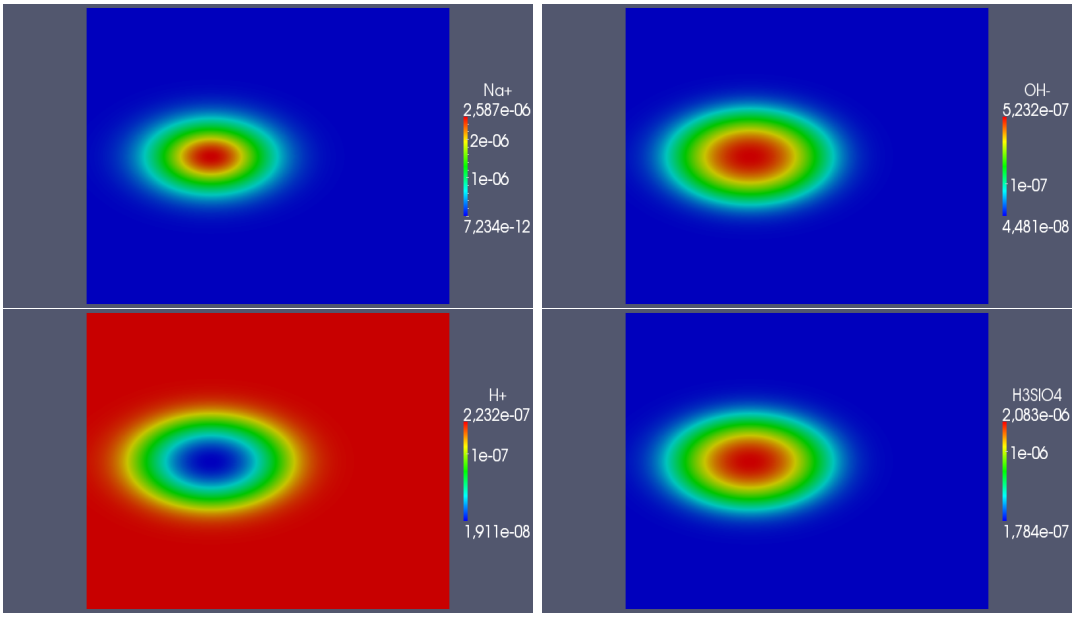


FIGURE 4.7 – Les concentrations à l'instant t=15 jours pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$

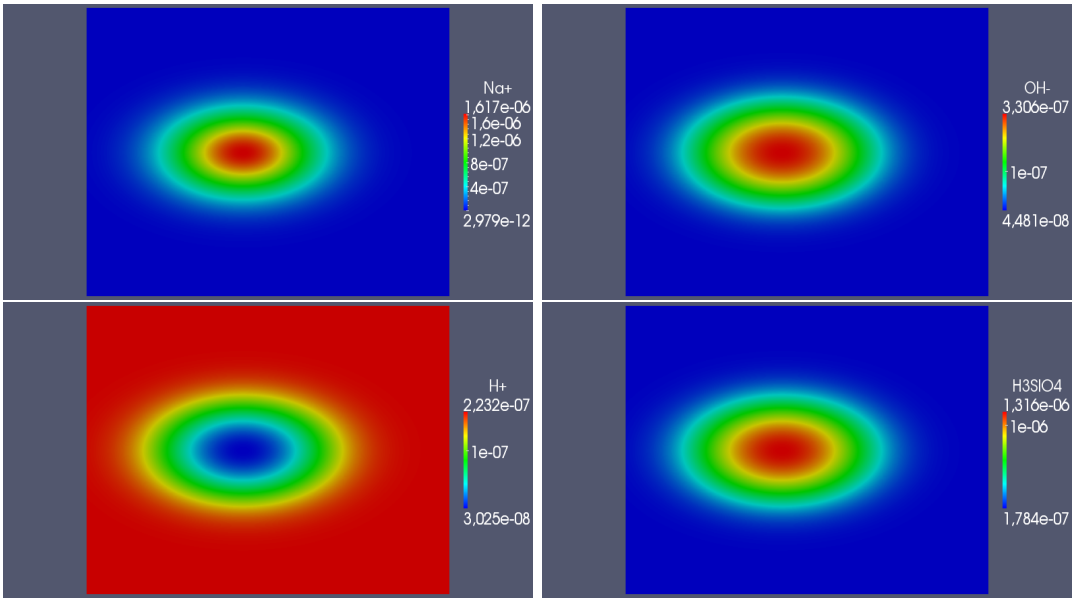


FIGURE 4.8 – Les concentrations à l'instant t=24 jours pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$

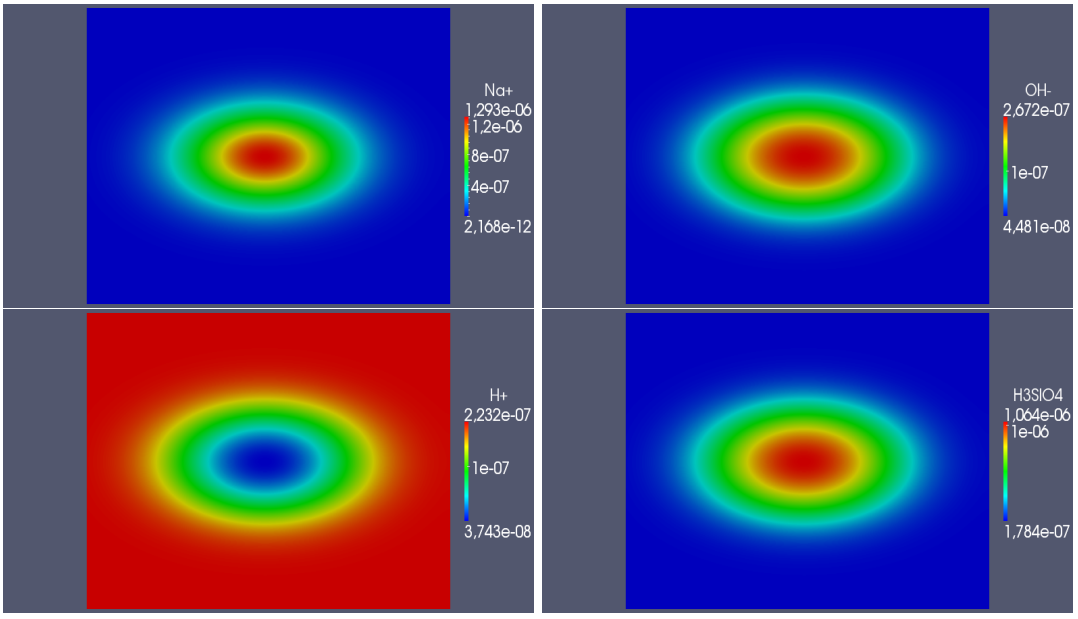


FIGURE 4.9 – Les concentrations à l'instant t=30 jours pour les deux composants et les deux espèces

Nous observons l'effet du couplage entre le transport et les réactions chimiques. À cause de la vitesse transversale nulle et de la dispersion, un panache d'ions Na^+ et OH^- se déplace vers la droite en grandissant. Le panache d'ions OH^- est plus aplati à cause des réactions chimiques. Le pH devient de plus en plus basique dans le domaine.

4.5 Comparaison entre GRT3DRSL et SNIA

Nous comparons dans cette section l'approche Globale DAE sans logarithmes (GRT3DRSL) et la méthode séquentielle non itérative (SNIA). Puisque la méthode SNIA est une méthode explicite, il est nécessaire de choisir les pas temps pour assurer la stabilité de cette méthode. Nous avons alors les pas temps externes suivants pour chaque maillage :

maillage	21×14	41×28	81×56	161×112
Δt	3600	1296	432	120

Nous allons maintenant comparer les deux méthodes GDAE et SNIA. Il n'est pas aisé de comparer deux méthodes différentes. Il en est ainsi de certains produits de la vie courante que nous sommes désormais habitués à comparer en terme de rapports qualité/poids ou encore qualité/prix.
Concernant les deux méthodes à comparer, les grandeurs d'intérêt sont les suivantes :

1. L'erreur, qui traduit l'inverse de la "précision" (c'est en quelque sorte la "qualité" du produit)
2. Le temps CPU, qui est le "coût" (ce sera le "prix" du produit)
3. La RAM nécessaire à sa bonne exécution, qui traduit l'occupation mémoire (l'équivalent pourrait être "l'encombrement" ou le "poids")

Nous avons donc trois grandeurs et par conséquent il suffit de deux rapports pour comparer les méthodes, sachant que le troisième rapport peut se déduire des deux autres. Ici, les deux rapports qui nous semblent pertinents sont d'une part le rapport précision/temps CPU (équivalent d'un rapport qualité/prix), d'autre part le rapport précision/RAM (équivalent d'un rapport qualité/poids).
Dans la Figure 4.10, nous traçons l'erreur E_{Na^+} en fonction du temps de calcul pour les deux méthodes GDAE et SNIA. Le diagramme log-log donne accès, de manière indirecte, au rapport précision/coût.
Nous remarquons que la courbe de méthode globale GDAE est sous la courbe de SNIA. L'approche globale DAE est beaucoup plus rapide que la méthode SNIA, pour une précision ainsi donnée. De plus, l'écart se creuse lorsque l'erreur diminue.
En effet, SNIA utilise une méthode explicite qui demande un pas temps adapté à chaque maillage pour assurer la convergence. Les pas de temps deviennent donc de plus en plus petits, alors que la méthode GDAE, qui utilise une méthode implicite stable, peut garder des pas de temps assez grands en raffinant le maillage.

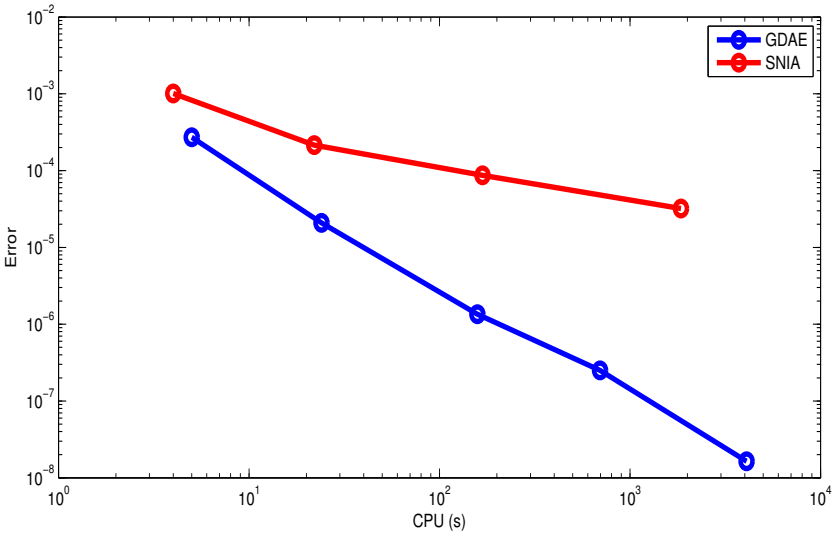


FIGURE 4.10 – Erreur E_{Na^+} pour GDAE (bleu) et SNIA (en rouge) pour les différents maillages par rapport au temps CPU

Dans la Figure 4.11, nous traçons l'erreur E_{Na^+} par rapport à la mémoire utilisée par les deux logiciels GRT3DRSL et SNIA et pour chaque maillage. Ce diagramme log-log donne accès, de manière indirecte, au rapport précision/occupation mémoire.

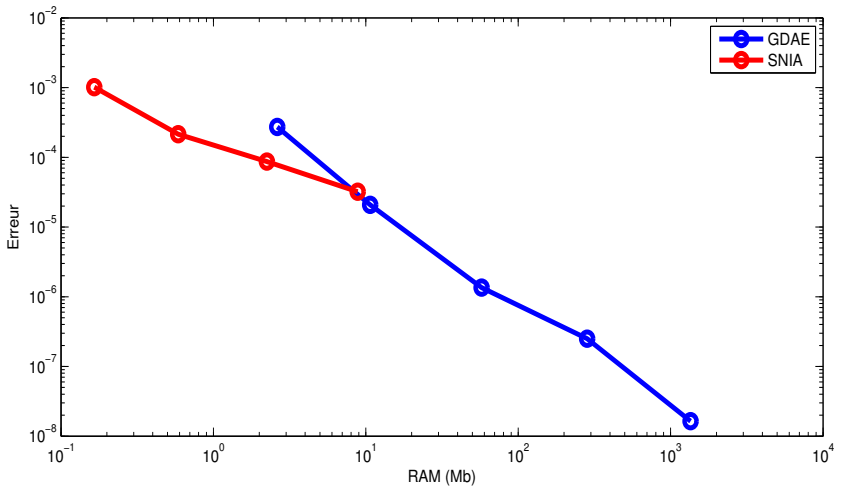


FIGURE 4.11 – Erreur E_{Na^+} pour GDAE (bleu) et SNIA (en rouge) pour les différents maillages par rapport au mémoire RAM

En ce qui concerne ce rapport précision/poids, la méthode SNIA est plus compétitive que GDAE pour de faibles précisions. Mais, pour une erreur de l'ordre 10^{-5}, les deux méthodes nécessitent environ la même taille de mémoire.
De plus, GDAE descend à une erreur de l'ordre 10^{-8} pour un poids mémoire de l'ordre du Giga-octets.
En éliminant la variable C dans le système algébro-différentiel, il est possible de réduire le poids de la mémoire (de $N_c \times N_m$).
En éliminant T dans le système non linéaire et en stockant judicieusement le Jacobien réduit, il est possible de réduire encore plus le poids de la mémoire.

4.6 Conclusion

Dans ce chapitre, nous avons montré les résultats numériques des deux méthodes globale approchée DAE et séquentielle non itérative SNIA, et aussi une comparaison entre les trois versions de GRT3D. Nous avons remarqué que la réduction dans le système résolu à chaque pas temps de Newton améliore le conditionnement de la matrice et aussi la convergence de la méthode. L'utilisation des concentrations directement sans les logarithmes, réduit aussi le conditionnement. La version sans logarithme sans traceur avec réduction est ainsi la plus rapide. Elle est aussi nettement plus rapide que la méthode séquentielle non itérative SNIA. Par contre, elle prend plus de mémoire pour une faible précision sans toutefois dépasser 1GB.

Chapitre 5

Cas test Benchmark 2D

5.1 Description

Dans ce chapitre on présente le cas test Benchmark, créé par le groupe MoMaS (Modélisations Mathématiques et Simulations Numériques liées aux problèmes de gestion des déchets nucléaires)[4]. Il y a trois modèles du cas test Benchmark, nous étudions dans ce travail le cas test " Easy test case " en 2D.
Le domaine de calcul est un rectangle qui contient deux milieux, notés A et B. Le milieu A est très perméable, avec une faible porosité et une faible réactivité et le milieu B est à faible perméabilité avec une porosité élevée et une forte réactivité. Toutes les dimensions sont normalisées : L est l'unité de longueur et T est l'unité de temps.

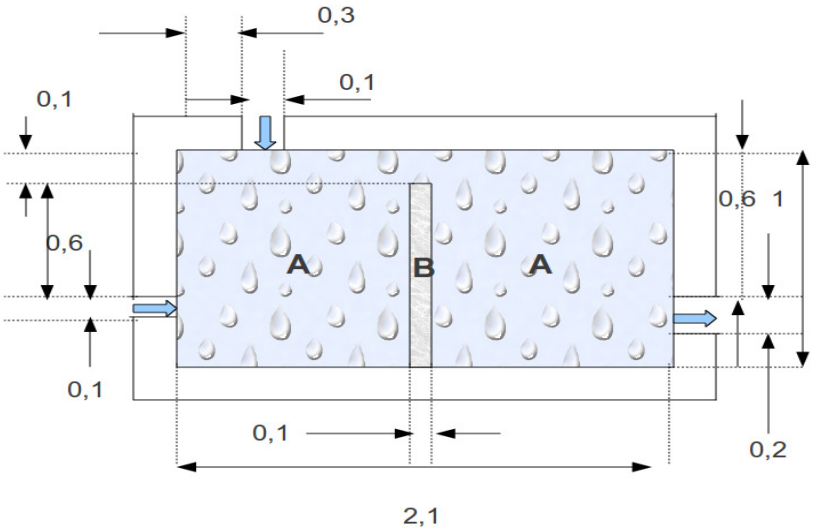


FIGURE 5.1 – Le domaine 2D du cas test Benchmark (Les dimensions sont indiquées en utilisant la longueur de référence L). Il y a deux entrées et une sortie.

5.1.1 Les caractéristiques physiques

Nous présentons dans le tableau 5.1 les données du transport advectif :

	Milieu A	Milieu B
Porosité ε	0.25	0.5
Perméabilité K (L.T^{-1})	10^{-2}	10^{-5}
Coefficient de diffusion moléculaire d_m (L^2T^{-1})	0	0
Dispersion longitudinale $\alpha_L(L)$	10^{-2}	6×10^{-2}
Dispersion transversale $\alpha_T(L)$	10^{-3}	6×10^{-3}
Temps initial (T)	0	0
Temps final (T)	6000	6000

TABLE 5.1 – Caractéristiques physiques du cas test Benchmark MoMaS

Dans les deux entrées du domaine, la vitesse d'écoulement $q = 2.25 \times 10^{-2} L.T^{-1}$ est imposée et le flux est nul sur le reste des bords.
Les figures 5.2 et 5.3 présentent respectivement le champ de vitesse et la charge obtenus numériquement avec MODFLOW, pour un maillage moyen 40×84.

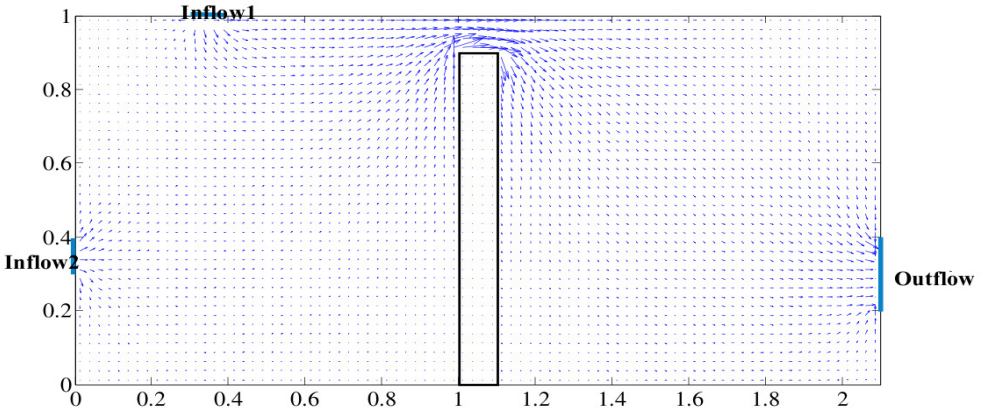


FIGURE 5.2 – Champ de vitesse dans le domaine 2D du cas test Benchmark MoMaS pour le maillage 40×84

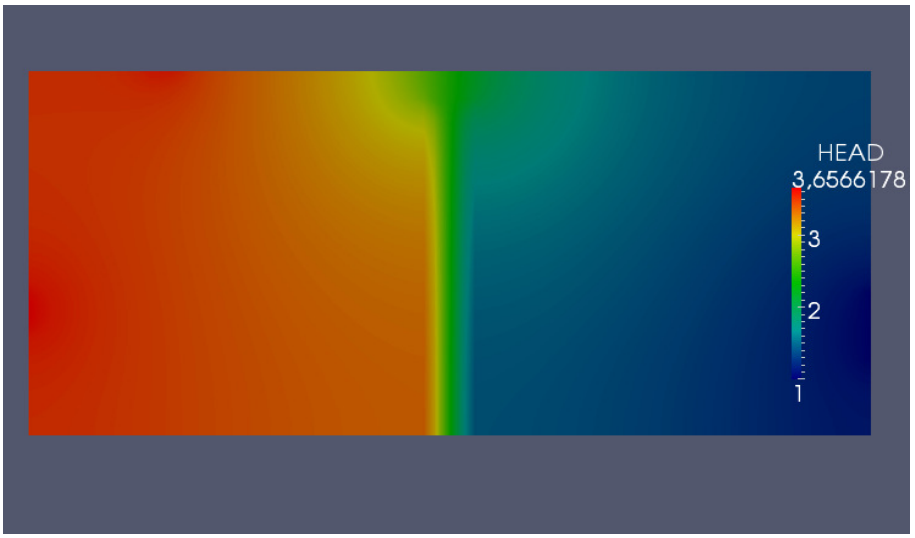


FIGURE 5.3 – La charge sur le domaine 2D du cas test Benchmark MoMaS pour le maillage 40×84

5.1.2 Le système chimique du cas test Benchmark

Le tableau de Morel [28] de ce cas est donné par le tableau 5.2 avec $N_c = 4$, $N_s = 1$, $N_p = 0$, $N_\alpha = 5$ et $N_\beta = 2$:

	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 5.2 – Tableau de Morel du cas test Benchmark MoMaS 2D " easy test case"

Dans nos simulations, nous avons remplacé la valeur $K_{\alpha_5} = 10^{+35}$ par $K_{\alpha_5} = 10^{+6}$ pour que le problème non linéaire à $t = 0$ soit assez bien conditionné. Nous remarquons à partir de ces coefficients que c_1 ne réagit pas avec les autres composants : c_1 est un traceur.

5.1.3 Les conditions initiales et les conditions aux limites

Dans le tableau 5.3, nous donnons les conditions initiales et les conditions aux limites pour les quatre composants aqueux (c_1, c_2, c_3, c_4) et le composant fixé s (sorbé). Pour les deux versions de GRT3D avec logarithme, nous remplaçons les valeurs des concentrations nulles par une très faible valeur (10^{-10}) pour éviter la singularité des valeurs nulles.

	T_1	T_2	T_3	T_4	W
	valeurs	initiales de la	concentration	totale	
Milieu A	0	-2	0	2	1
Milieu B	0	-2	0	2	10
	Concentration	totale imposée	sur la frontière	entrante	
injection $t \in [0, 5000]$	0.3	0.3	0.3	0	0
Relâchement $t \in [5000, 6000]$	0	-2	0	2	0

TABLE 5.3 – Les conditions initiales et les conditions aux limites du cas test Benchmark MoMaS 2D " easy test case"

La première période de temps, de $t = 0$ à $t = 5000$, est une période d'injection aux frontières entrantes. Puis, de $t = 5000$ à $t = 6000$, la deuxième période de temps est une période de relâchement.

5.2 Expériences numériques

Le domaine est maillé avec un maillage régulier de $n_1 \times n_2$ cellules. Nous avons défini quatre maillages différents, allant d'un maillage grossier de 20×42 cellules à un maillage fin de 80×168 cellules.
Dans le tableau 5.4, nous indiquons les principaux paramètres de sundials utilisés.

Ftol	10^{-8}	Nombre maximum d'itérations de Newton	4
Stol	10^{-10}	L'ordre maximum de la méthode BDF	5
Atol	10^{-11}	Nombre maximum de l'échec du test d'erreur	60
Rtol	10^{-7}	Nombre maximum de l'échec de convergence	60
Nombre maximum de pas de temps internes	1000000	coefficient du test de convergence non linéaire	0.33

TABLE 5.4 – Paramètres de Sundials

Les calculs sont effectués en séquentiel, sur une machine Intel Xeon, 3.47 GHZ, la RAM est 24 Go et le cache : 12 Mo.

5.3 Comparaison des trois versions de GRT3D

Pour chaque simulation, nous utilisons N_t pas de temps externes pour les sorties. Pour un maillage donné, on note $\hat{c}_i(m_j, t_n)$ les concentrations calculées de c_i aux différents points $m_j, j = 1, ..., N_m$ et à divers instants $t_n, n = 1, ..., N_t$ pour chaque version de GRT3D avec logarithmes (**GRT3D** et **GRT3DRL**).
On note $c_i(m_j, t_n)$ les concentrations calculées avec la version sans logarithme réduite **GRT3DRSL** sans traceur (référence du calcul). Pour chaque version de GRT3D avec logarithmes, on calcule l'erreur E_{c_i} par rapport à **GRT3DRSL** définie sous cette forme :

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

Les tables 5.5 à 5.9 indiquent les erreurs calculées respectivement pour les composants aqueux c_1, c_2, c_3, c_4, et le composant fixé s :

Maillage	GRT3D	GRT3DRL	GRT3D (sans traceur)	GRT3DRL (sans traceur)
20x42	1.696328E-07	3.930183E-08	2.269643E-08	3.893719E-09
32x56	1.618450E-07	5.025178E-08	5.556718E-08	4.782764E-09
40x84	1.696328E-07	3.927450E-08	6.663325E-08	3.474018E-09
80x168	1.690063E-07	3.798127E-08	1.221142E-08	1.628034E-09

TABLE 5.5 – L'erreur E_{c_1} pour le composant aqueux c_1

Maillage	GRT3D	GRT3DRL	GRT3D (sans traceur)	GRT3DRL (sans traceur)
20x42	6.402528E-08	3.112242E-09	3.227338E-08	2.568312E-09
32x56	6.950711E-08	1.622862E-09	1.287118E-08	1.509531E-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 5.6 – L'erreur E_{c_2} pour le composant aqueux c_2

Maillage	GRT3D	GRT3DRL	GRT3D (sans traceur)	GRT3DRL (sans traceur)
20x42	7.691391E-09	7.584722E-09	6.710077E-09	6.580256E-09
32x56	6.501412E-09	6.463589E-09	5.115954E-09	5.083994E-09
40x84	1.148123E-09	1.071758E-09	2.763441E-09	2.732291E-09
80x168	1.293119E-09	1.249099E-09	1.548455E-09	1.552428E-09

TABLE 5.7 – L'erreur E_{c_3} pour le composant aqueux c_3

Maillage	GRT3D	GRT3DRL	GRT3D (sans traceur)	GRT3DRL (sans traceur)
20x42	3.177351E-08	1.612077E-09	2.057822E-08	1.409932E-09
32x56	3.201613E-08	4.090839E-09	2.066802E-08	3.358463E-09
40x84	3.182991E-08	2.485368E-09	1.467422E-08	1.466819E-09
80x168	2.302817E-08	2.221754E-09	2.002241E-08	1.958571E-09

TABLE 5.8 – L'erreur E_{c_4} pour le composant aqueux c_4

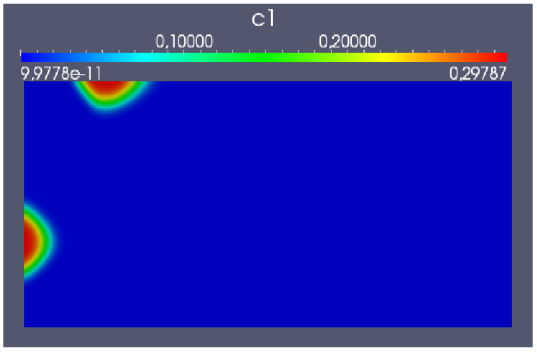
Maillage	GRT3D	GRT3DRL	GRT3D (sans traceur)	GRT3DRL (sans traceur)
20x42	2.000629E-07	3.334816E-08	9.442679E-07	3.010784E-08
32x56	2.542402E-07	1.390674E-08	2.522896E-07	1.193361E-08
40x84	2.004018E-07	1.029330E-08	1.006046E-07	1.004838E-08
80x168	4.808055E-07	4.529755E-08	7.342318E-07	7.247864E-08

TABLE 5.9 – L'erreur E_s pour le composant fixé s

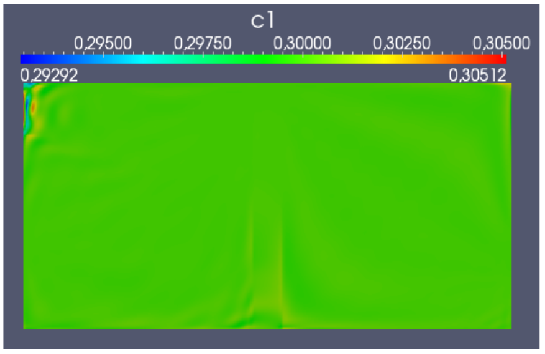
Nous remarquons que pour l'ensemble des maillages et des composants, l'erreur est du même ordre pour les deux versions de GRT3D. Les résultats sont semblables pour toutes les versions.

5.4 Résultats

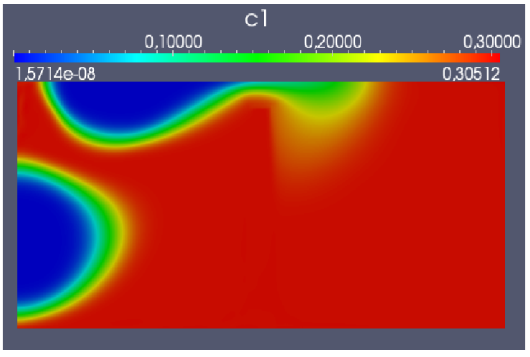
La figure 5.4 montre les concentrations de composant c_1 avec le maillage le plus fin de 80×168 et avec GRT3DRSL, à différents instants. On voit ici l'effet du transport et des conditions aux limites.



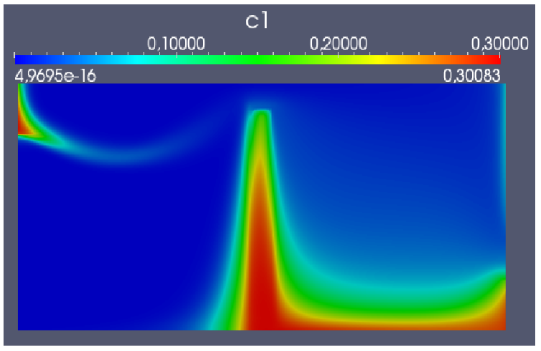
(a) Les concentrations du composant aqueux c_1 à $t = 1T$



(b) Les concentrations du composant aqueux c_1 à $t = 4000T$



(c) Les concentrations du composant aqueux c_1 à $t = 5010T$



(d) Les concentrations du composant aqueux c_1 à $t = 5800T$

FIGURE 5.4 – Les concentrations du composant aqueux c_1 à différents instants (avec des échelles différentes)

Les figures 5.5 à 5.10 montrent les concentrations des composants aqueux c_2, c_3, c_4 et composant fixé s à différents instants. Nous observons l'évolution des concentrations en composants sur les deux périodes.

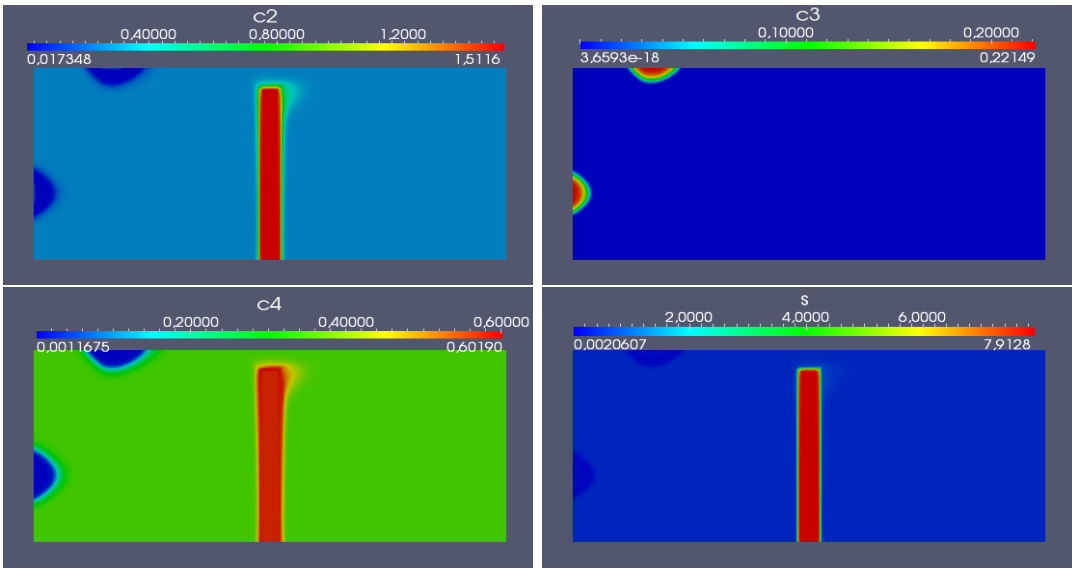


FIGURE 5.5 – Les concentrations des composants aqueux et composant fixé à l'instant $t = 10T$

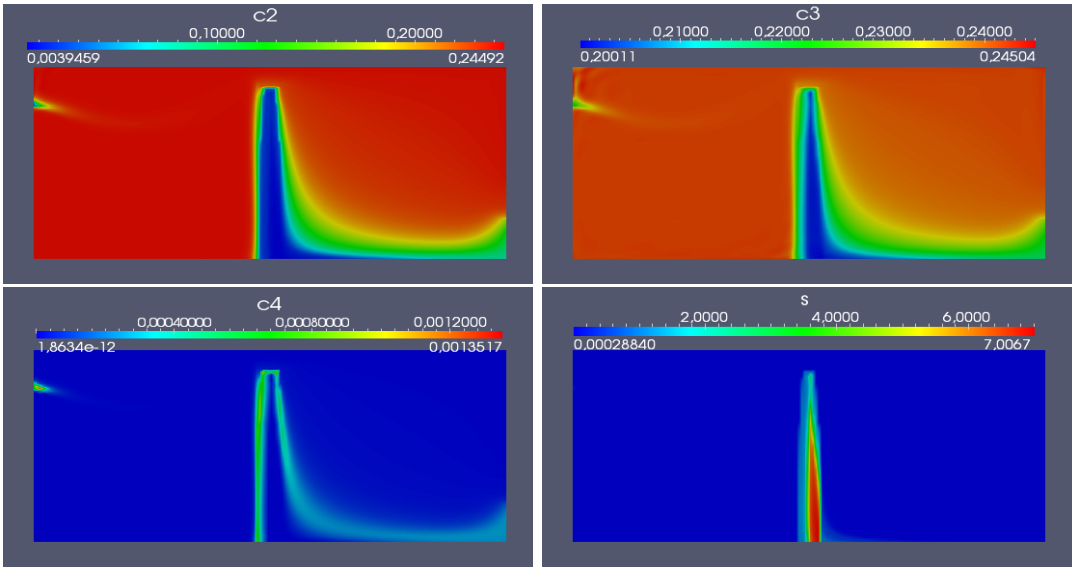


FIGURE 5.6 – Les concentrations des composants aqueux et composant fixé à l'instant $t = 2000T$

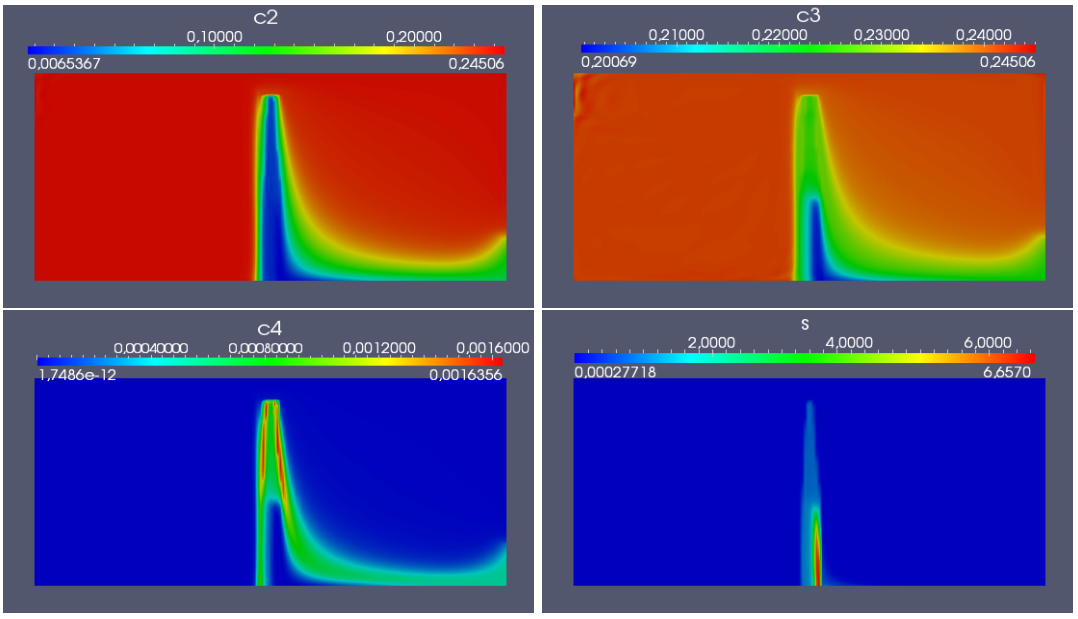


FIGURE 5.7 – Les concentrations des composants aqueux et composant fixé à l'instant $t = 4000T$

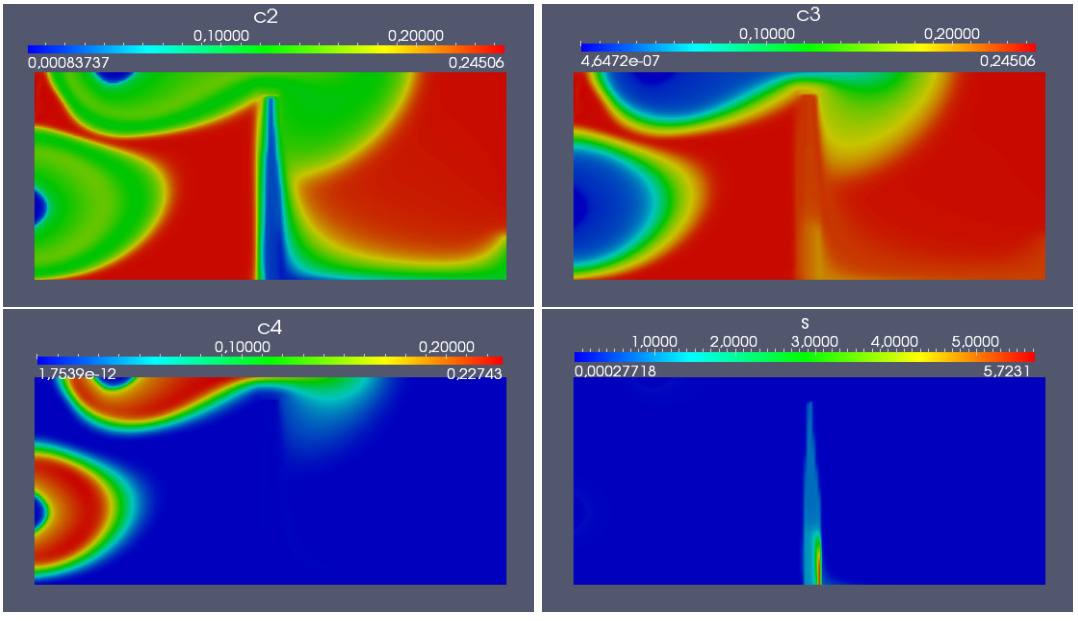


FIGURE 5.8 – Les concentrations des composants aqueux et composant fixé à l'instant $t = 5010T$

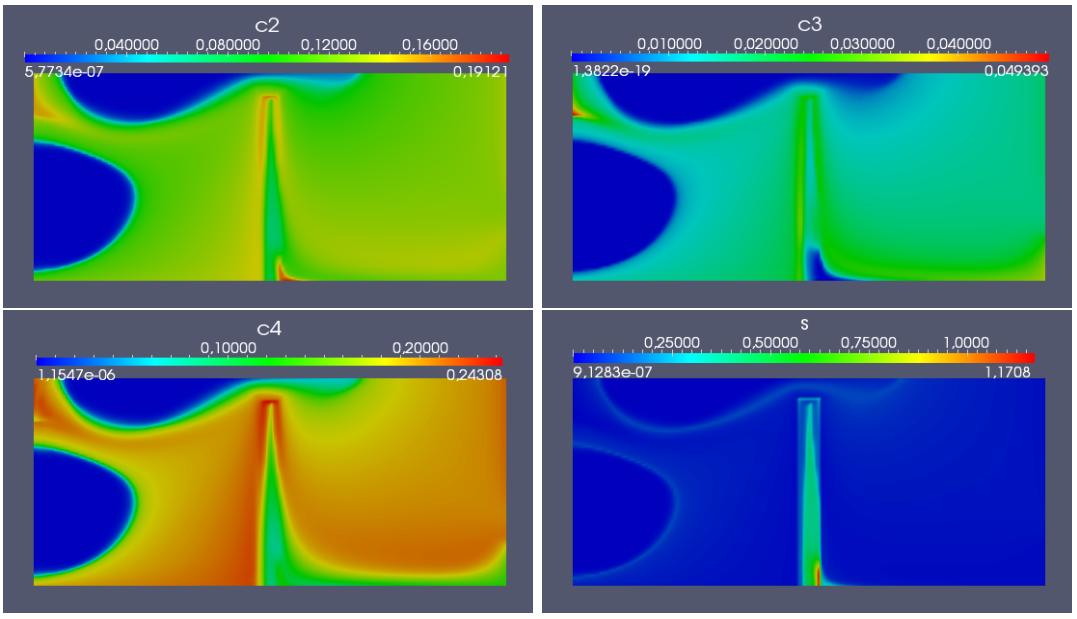


FIGURE 5.9 – Les concentrations des composants aqueux et composant fixé à l'instant $t = 5200T$

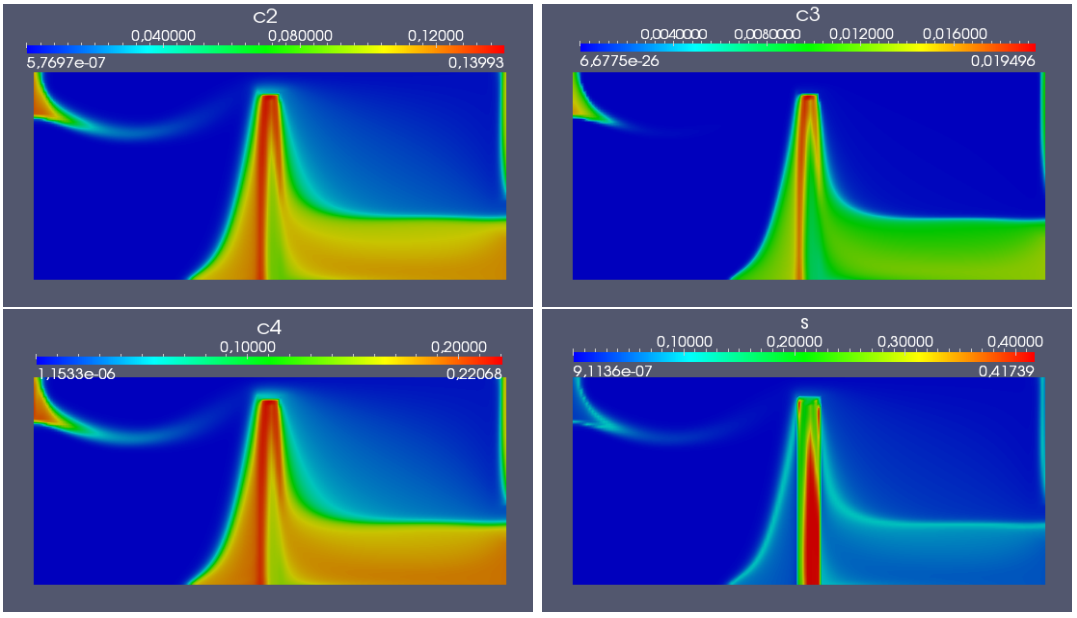


FIGURE 5.10 – Les concentrations des composants aqueux et composant fixé à l'instant $t = 5800T$

5.5 Analyse du temps de calcul

5.5.1 Temps total

Dans les tableaux 5.10 et 5.11, nous donnons le temps CPU total pour les trois versions et pour différents maillages.
Le tableau 5.10 présente le temps CPU pour les trois versions avec traceur :

maillage	GRT3D		GRT3DRL		GRT3DRSL
	tailles de système	CPU time	tailles de système	CPU time	CPU time
20x42	10920	52 m 52 s	4200	27 m 16 s	19 m 6 s
32x56	23296	3 h 28 m 1 s	8960	1 h 50 m 33 s	1 h 14 m 13 s
40x84	43680	9 h 52 m 24 s	16800	4 h 44 m 40 s	3 h 9 m 8 s
80x168	174720	5 j 15 h 12 m 15 s	67200	3 j 11 h 24 m 23 s	2 j 2 h 12 m 5 s

TABLE 5.10 – Temps CPU pour le test Benchmark MoMaS 2D avec traceur

Le tableau 5.11 présente le temps CPU pour les trois versions sans traceur :

maillage	GRT3D		GRT3DRL		GRT3DRSL
	tailles de système	CPU time	tailles de système	CPU time	CPU time
20x42	8400	40 m 20 s	3360	19 m 1 s	13 m 49 s
32x56	17920	2 h 32 m 47 s	7168	1 h 19 m 59 s	55 m 17 s
40x84	33600	7 h 20 m 22 s	13440	4 h 4 m 1 s	2 h 12 m 45 s
80x168	134400	2 j 18 h 22 m 46 s	53760	2 j 6 h 56 m 42 s	1 j 8 h 20 m 46 s

TABLE 5.11 – Temps CPU pour le test Benchmark MoMaS 2D sans traceur

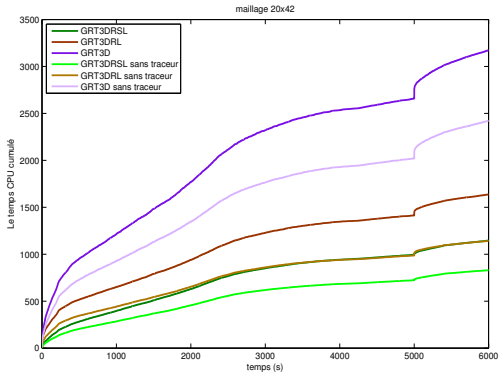
Nous observons trois sources de réduction du temps de calcul :
- enlever le traceur.
- réduire le système linéaire.
- garder les valeurs sans logarithme.

Enlever le traceur dans GRT3D réduit le temps de calcul de $5j\ 15h$ à $2j\ 18h$, d'un facteur deux, notamment parce que la taille du système est réduite.
Après diminution de la taille du système linéaire, le temps de calcul est réduit de $5j\ 15h$ à $3j\ 11h$ dans le cas avec traceur et de $2j\ 18h$ à $2j\ 6h$ dans le cas sans traceur, notamment à cause de la réduction de la taille du système.
En effet, la taille de système dans GRT3D avec traceur est $(3N_c + N_s + N_p) \times N_m = 13N_m$; elle est réduite dans GRT3DRL à $(N_c+N_s+N_p)\times N_m = 5N_m$. Dans les versions sans traceur, la taille passe de $10N_m$ dans GRT3D à $4N_m$ dans GRT3DRL.
Garder les valeurs des concentrations sans logarithmes réduit le temps de calcul d'un facteur deux, par exemple sans traceur de $2j\ 6h$ avec GRT3DRL à $1j\ 8h$ avec GTR3DRSL. Ici, la taille du système est la même.

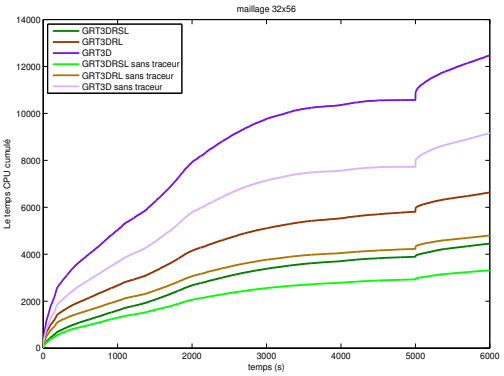
L'accroissement des performances peut aussi être dû, dans les trois techniques, à un meilleur conditionnement du Jacobien ou à une meilleure convergence de Newton ou à des pas de temps plus grands. Pour déterminer les différentes raisons, on va analyser le conditionnement de la matrice Jacobienne des trois versions, avec et sans traceur, et les sorties de sundials.

5.5.2 Temps de calcul cumulé

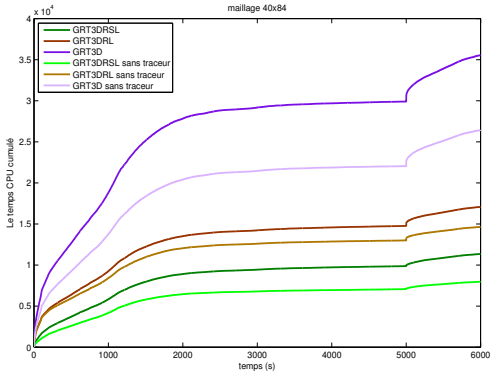
Pour chaque pas de temps externe, on a accès au temps de calcul (le temps CPU). Dans la figure 5.11, nous présentons le temps de calcul cumulé à chaque itération externe (sinon les courbes seraient trop oscillantes) pour chaque maillage et chaque logiciel :



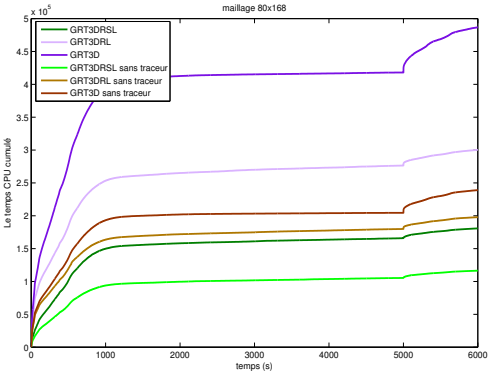
(a) Le temps de calcul cumulé à chaque itération externe pour le maillage 20×42



(b) Le temps de calcul cumulé à chaque itération externe pour le maillage 32×56



(c) Le temps de calcul cumulé à chaque itération externe pour le maillage 40×84



(d) Le temps de calcul cumulé à chaque itération externe avec pour le maillage 80×168

FIGURE 5.11 – Le temps de calcul cumulé à chaque itération externe avec et sans traceur pour les quatre maillages.

Nous remarquons dans les figures (a), (b), (c) et (d) l'augmentation de temps de calcul à chaque itération externe avec le raffinement de maillage pour les trois versions. Enlever le traceur réduit le temps de calcul pour les différents maillages, et la réduction du système linéaire réduit aussi le temps de calcul à chaque itération externe.
Enfin, l'utilisation des concentrations sans logarithmes diminue le temps de calcul à chaque itération externe.

5.6 Analyse du conditionnement de la Jacobienne

Dans cette section, nous analysons le conditionnement de la Jacobienne pour les trois logiciels (**GRT3D** avec logarithme, la version avec logarithme réduite **GRT3DRL** et la version sans logarithme réduite **GRT3DRSL**), avec traceur et sans traceur.
Les figures 5.12 à 5.15 montrent les conditionnements pour les différents maillages.

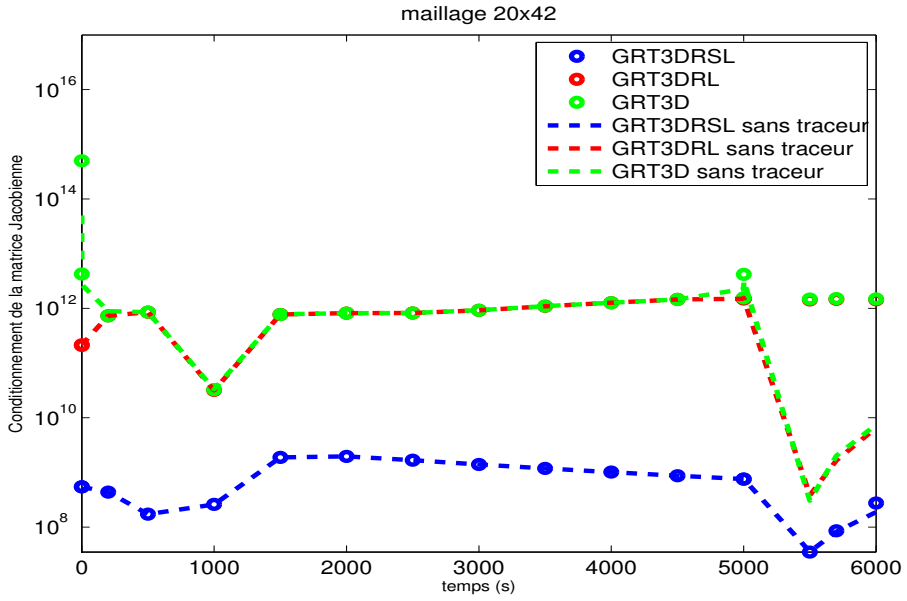


FIGURE 5.12 – Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 20×42

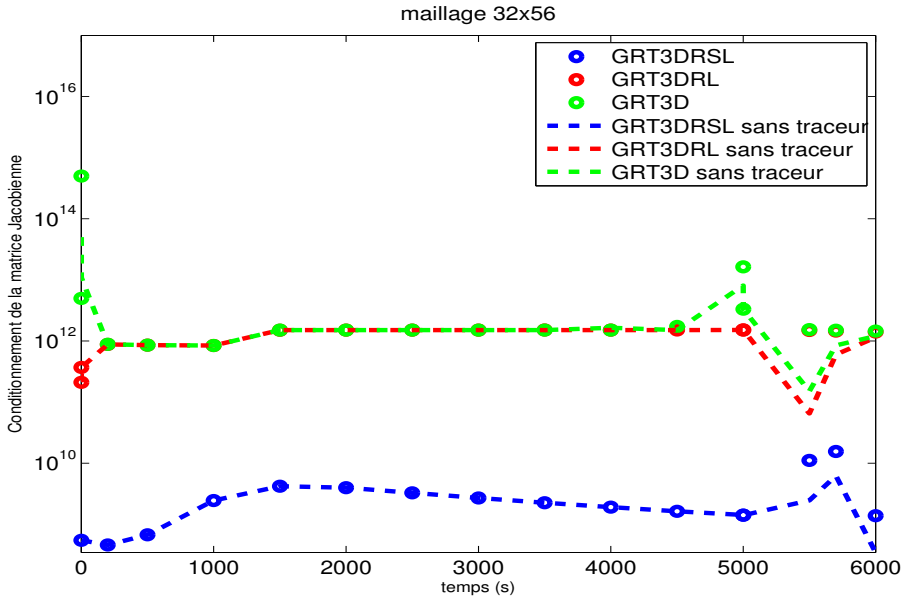


FIGURE 5.13 – Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 32×56

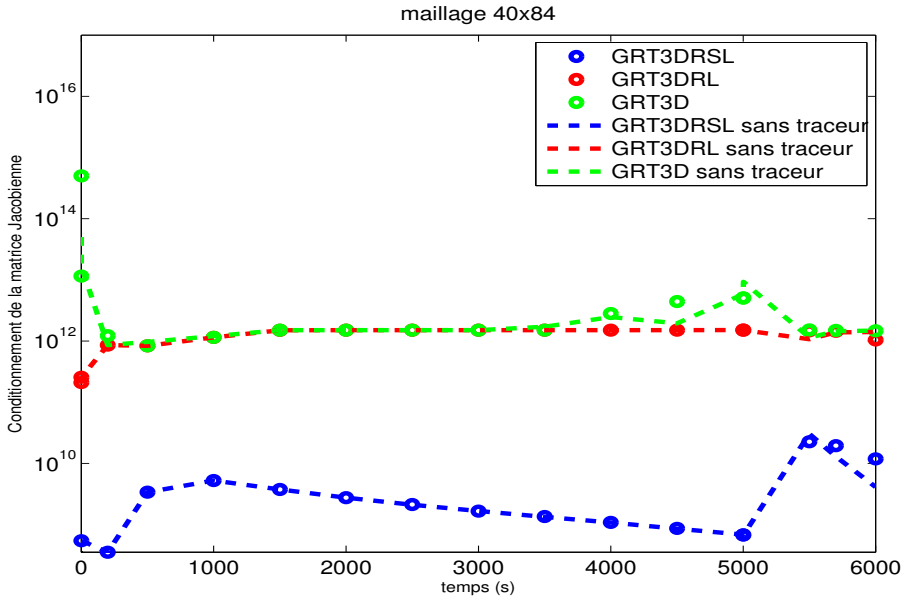


FIGURE 5.14 – Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 40×84

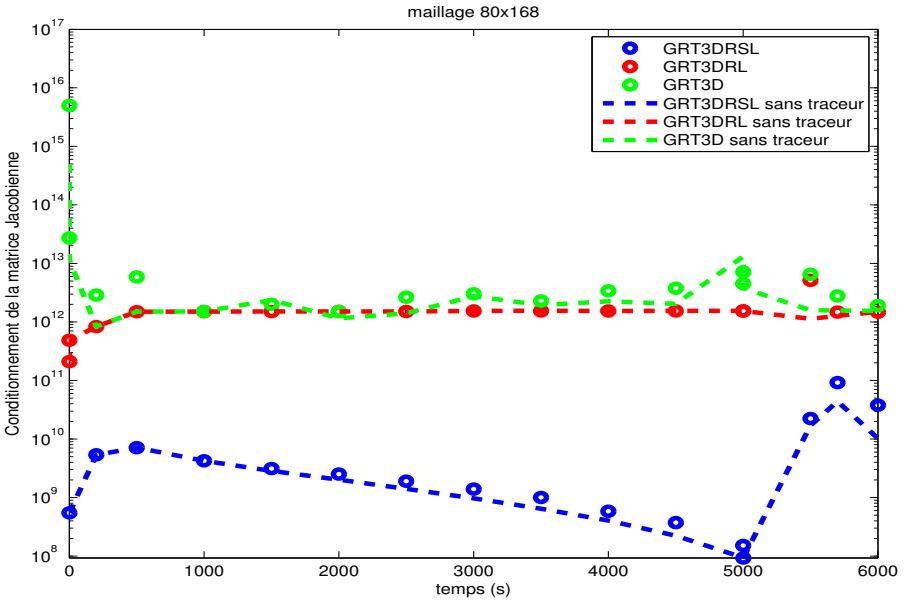


FIGURE 5.15 – Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 80×168

Nous observons que nous avons nettement amélioré le conditionnement de la matrice Jacobienne quand nous avons utilisé les concentrations sans le passage par les logarithmes (GRT3DRSL). Par conséquent, la réduction du conditionnement peut expliquer au moins en partie le gain de temps entre GRT3DRL et GRT3DRSL.
Dans GRT3DRSL, il n'y a pas de différence de conditionnement par rapport au calcul avec traceur et sans traceur parce que la matrice Jacobienne est bien conditionnée (voir le chapitre 1).
Quand nous avons réduit la taille du système (GRT3DRL) avec l'utilisation des logarithmes, nous avons amélioré un peu le conditionnement de la matrice Jacobienne au début de chaque période mais pas pour le reste des périodes. Le calcul sans traceur améliore le conditionnement de la matrice pour les deux logiciels avec logarithmes à $t = 0$ et $t = 5000$, parce que dans ces cas la matrice jacobienne est presque singulière (voir le chapitre 1).

5.7 Analyse des sorties de Sundials

Pour détecter d'autres sources de performances, nous analysons les sorties de sundials (fichier global.out, voir chapitre 3) :

5.7.1 Nombre de pas de temps internes

Pour chaque pas de temps externe, on a accès au nombre de pas de temps. Dans les figures 5.16 à 5.19, nous présentons le nombre cumulé de ces pas de temps (sinon les courbes seraient trop oscillantes) pour chaque maillage et chaque logiciel :

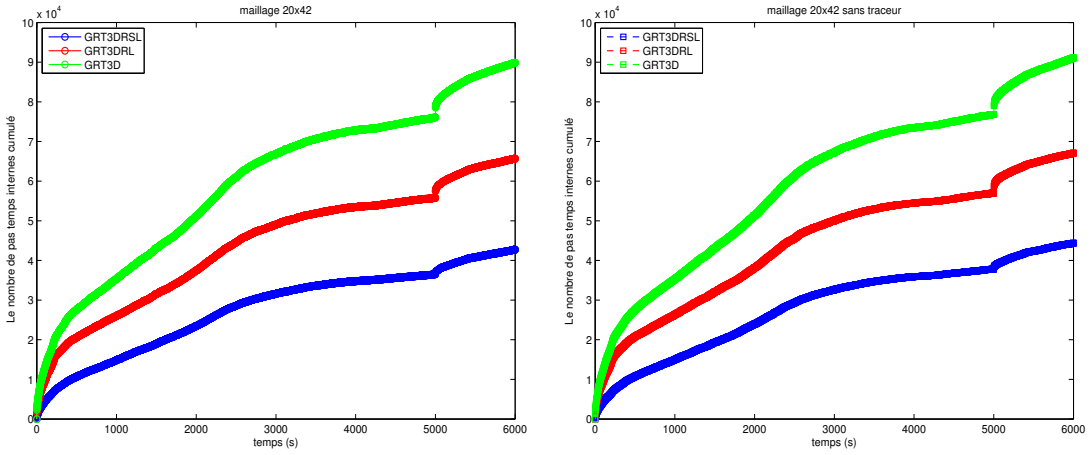


FIGURE 5.16 – Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 20×42 (À gauche : avec traceur, à droite : sans traceur)

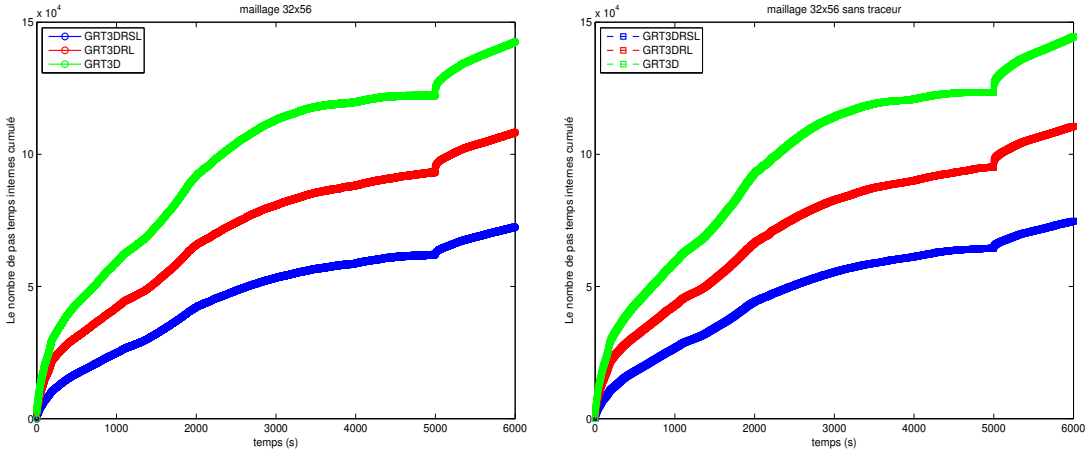


FIGURE 5.17 – Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 32×56 (À gauche : avec traceur, à droite : sans traceur)

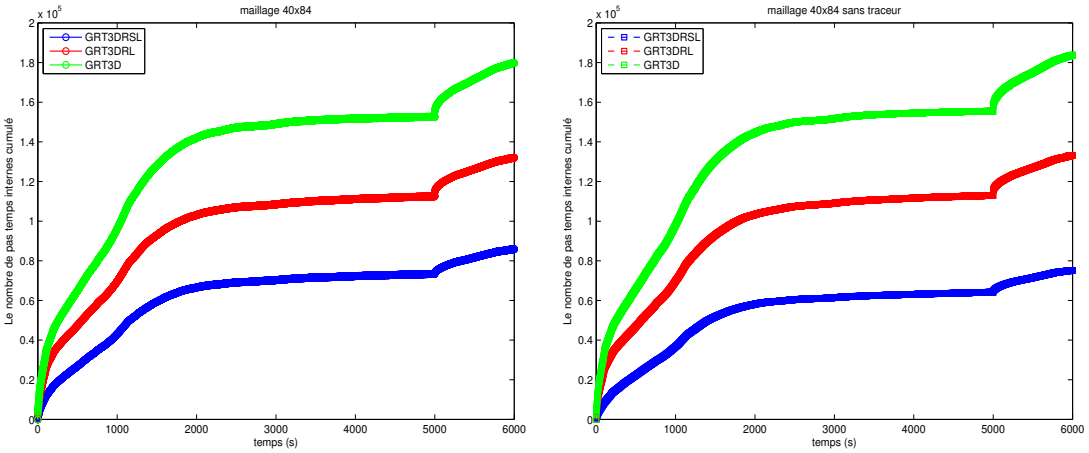


FIGURE 5.18 – Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 40×84 (À gauche : avec traceur, à droite : sans traceur)

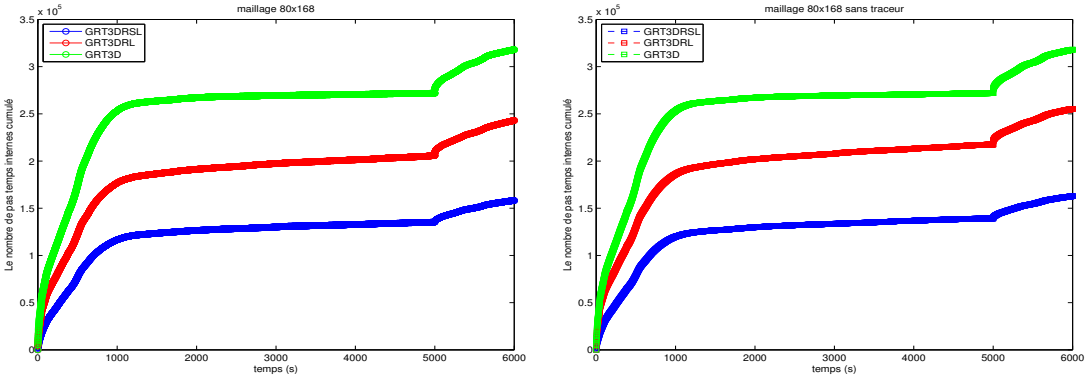


FIGURE 5.19 – Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 80×168 (À gauche : avec traceur, à droite : sans traceur)

Nous remarquons que le nombre de pas de temps internes augmente à chaque pas de temps externe de la même façon pour les trois logiciels : l'augmentation est forte de $t = 0$ à un instant t_1, dont la valeur dépend du maillage, puis est très faible jusqu'à $t = 5000$, où elle est à nouveau forte. On voit l'effet des conditions aux limites. Nous observons aussi que ce nombre est le même avec ou sans traceur et qu'il augmente avec le raffinement de maillage.
Le nombre de pas de temps internes avec GRT3D est plus grand qu'avec GRT3DRL, où la taille du système linéaire est réduite. Ce dernier est plus grand qu'avec GRT3DRSL, où la matrice Jacobienne est mieux conditionnée à $t = 0$ et $t = 5000$. Le nombre de pas de temps internes varie donc beaucoup au cours du temps. La stratégie du pas de temps adaptatif est efficace pour accroître les performances tout en garantissant une bonne précision.

5.7.2 Nombre d'échecs du test d'erreur

Pour chaque pas de temps externe, nous avons également accès au nombre d'échecs du test d'erreur . Dans les figures 5.20 à 5.23 , nous présentons ce nombre cumulé pour chaque maillage et chaque logiciel :

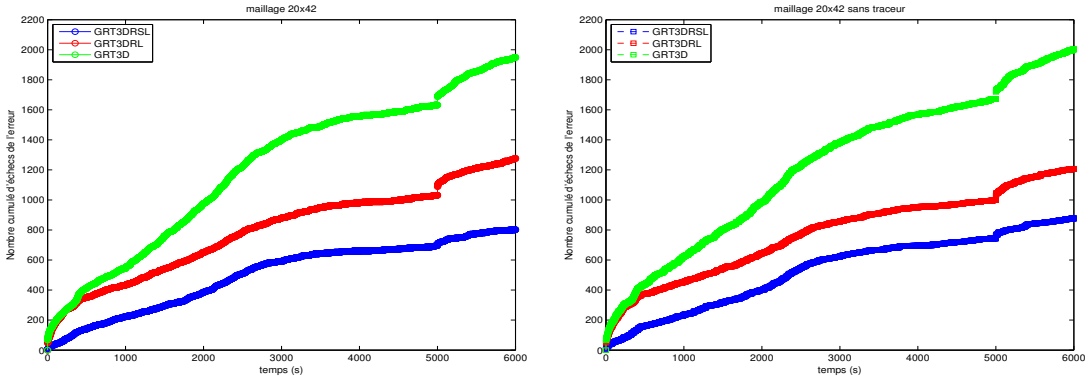


FIGURE 5.20 – Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 20×42 (à gauche : avec traceur, à droite : sans traceur)

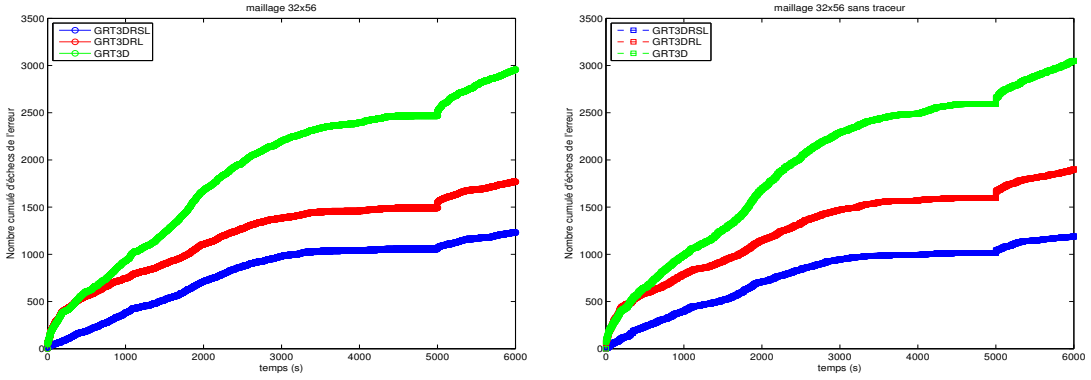


FIGURE 5.21 – Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 32×56 (à gauche : avec traceur, à droite : sans traceur)

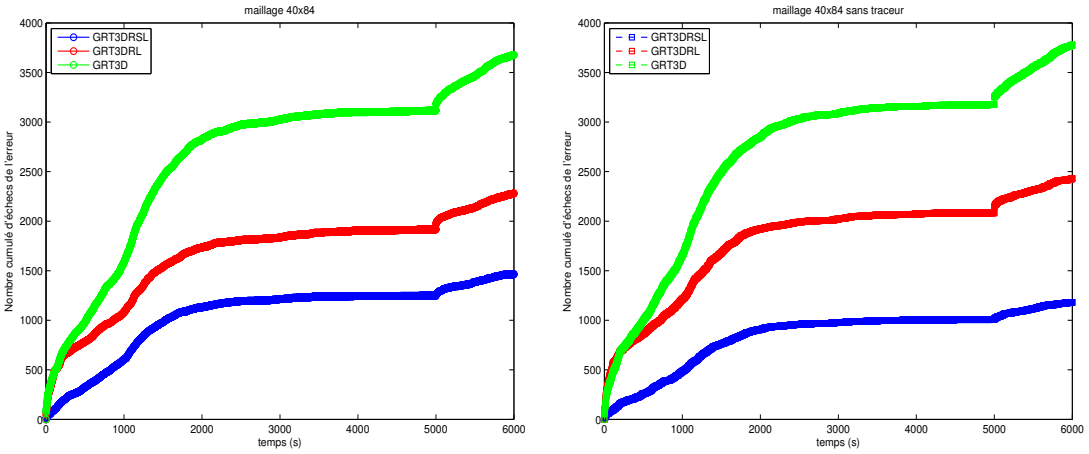


FIGURE 5.22 – Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 40×84 (à gauche : avec traceur, à droite : sans traceur)

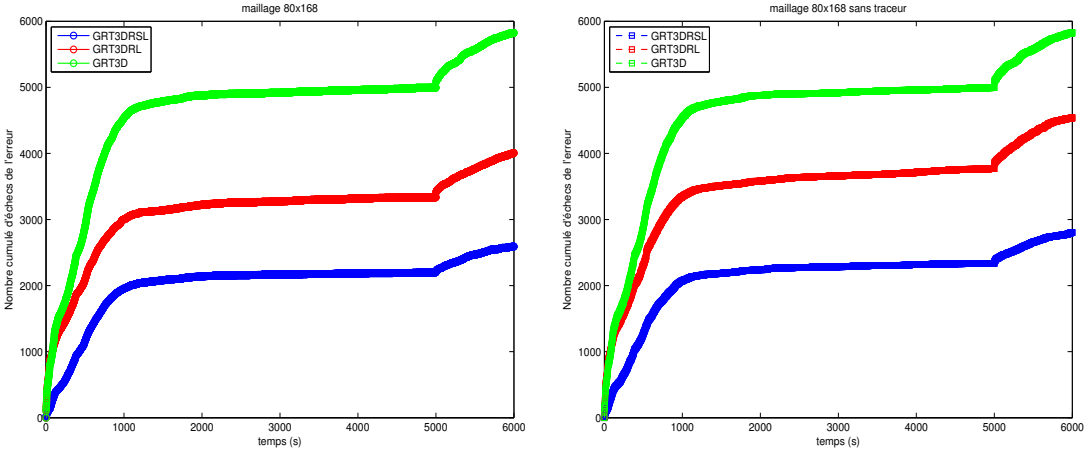


FIGURE 5.23 – Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 80×168 (à gauche : avec traceur, à droite : sans traceur)

Le comportement dynamique du nombre d'échecs est similaire à celui du nombre de pas de temps internes et celui du temps cumulé de calculs. Il y a donc un lien clair entre les trois quantités.
Le gain en temps CPU est dû essentiellement à une diminution du nombre d'échecs du test d'erreur .

5.7.3 Nombre d'échecs de la Jacobienne

Maintenant nous montrons dans les figures 5.24 à 5.26 le nombre d'échecs de la Jacobienne pour chaque logiciel :

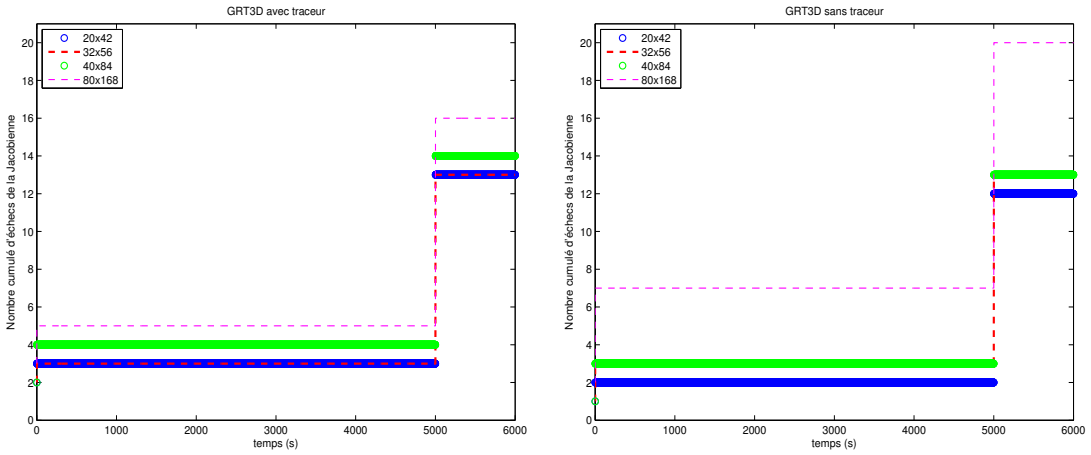


FIGURE 5.24 – Nombre cumulé d'échecs de la Jacobienne du cas test Benchmark MoMaS pour GRT3D (À gauche : avec traceur, à droite : sans traceur)

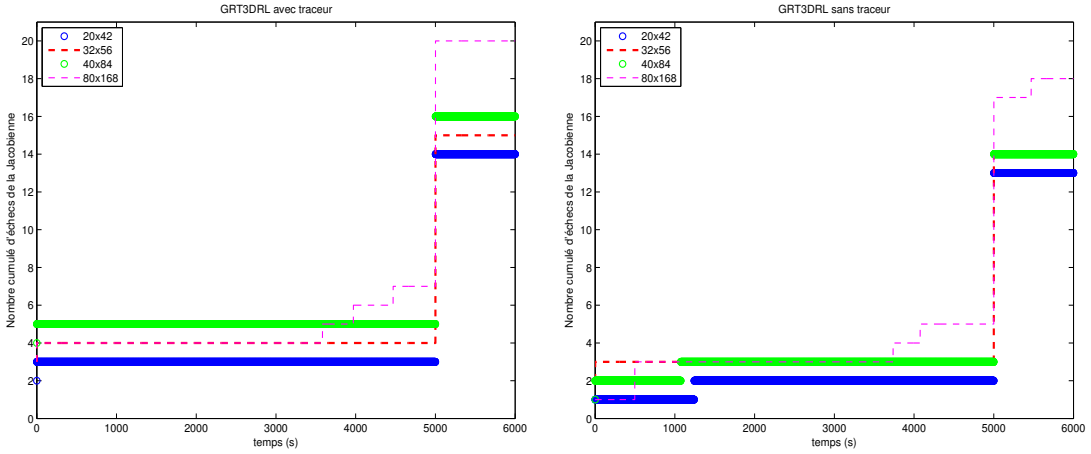


FIGURE 5.25 – Nombre cumulé d'échecs de la Jacobienne du cas test Benchmark MoMaS pour GRT3DRL (À gauche : avec traceur, à droite : sans traceur)

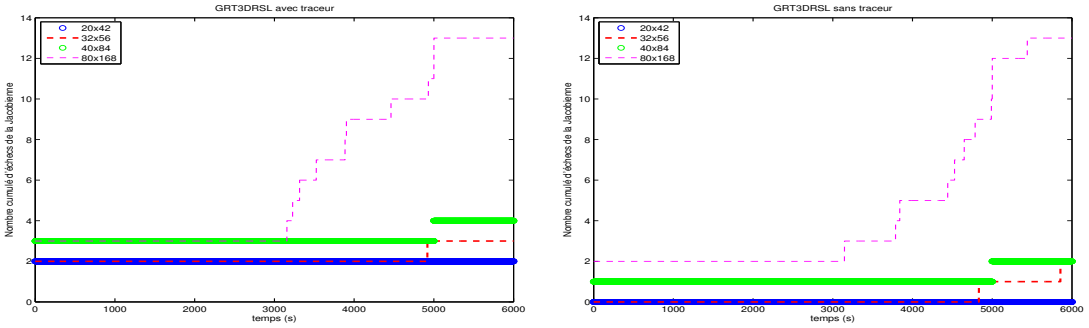


FIGURE 5.26 – Nombre cumulé d'échecs de la Jacobienne du cas test Benchmark MoMaS pour GRT3DRSL (À gauche : avec traceur, à droite : sans traceur)

Nous observons que le nombre d'échecs de la Jacobienne avec GRT3D est plus grand que celui de GRTRDRL, et ce dernier est plus grand que celui de GRT3DRSL. En général, il y a un peu moins d'échecs sans traceur qu'avec.
Les échecs ont lieu principalement à $t = 0$ et à $t = 5000$, ce qui s'explique par le conditionnement du Jacobien. Ces échecs à $t = 0$ et à $t = 5000$ ont un impact sur le nombre d'échec en temps. Globalement, le nombre d'échecs du Jacobien est faible, et moins d'influence sur le temps de calcul que le nombre d'échecs en temps.

5.8 Analyse des étapes linéaires

5.8.1 Nombre de résolutions et de factorisations

Le nombre de pas de temps a un impact direct sur le nombre d'itérations de Newton autrement dit le nombre de résolutions de système linéaire et le nombre de factorisations. Nous montrons ici les résultats avec et sans traceur, pour les quatre maillages et les trois versions.

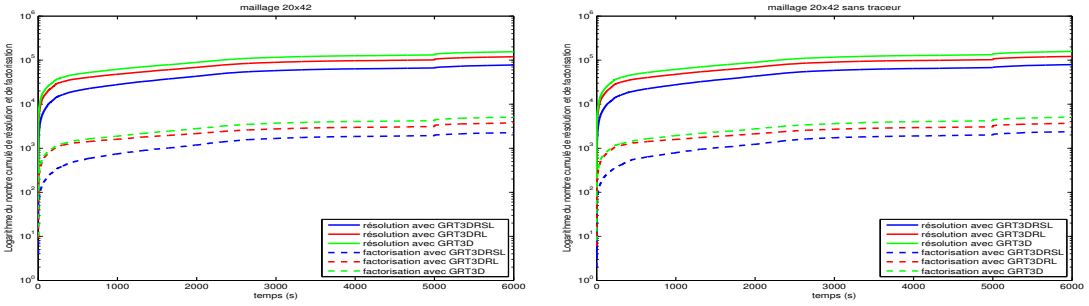


FIGURE 5.27 – Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 20×42 (À gauche : avec traceur, à droite : sans traceur)

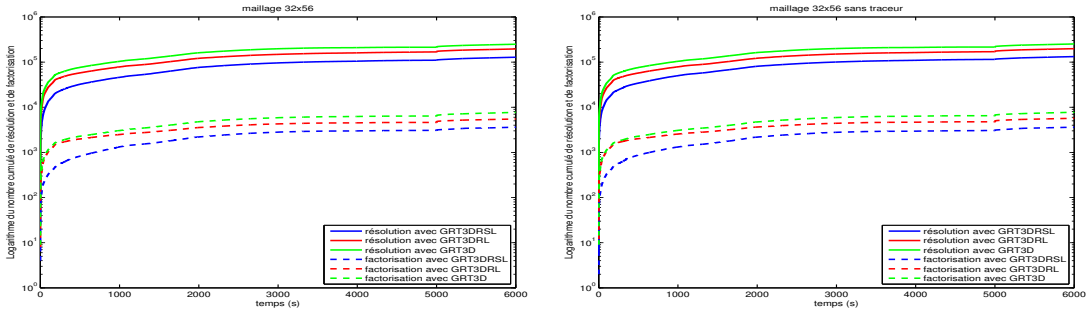


FIGURE 5.28 – Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 32×56 (À gauche : avec traceur, à droite : sans traceur)

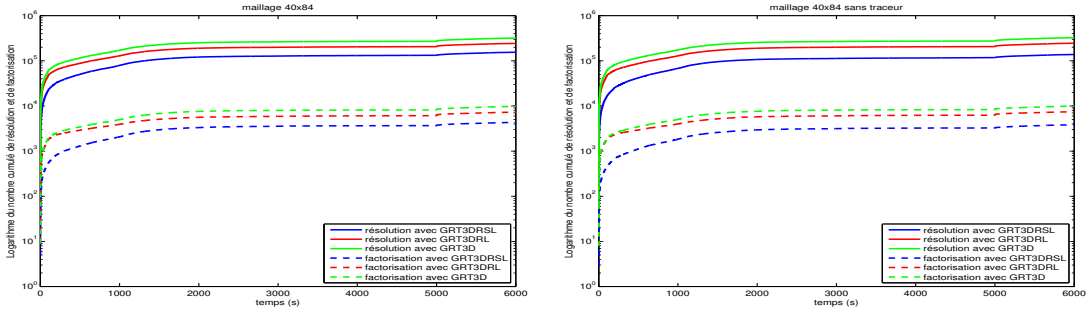


FIGURE 5.29 – Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 40×84 (À gauche : avec traceur, à droite : sans traceur)

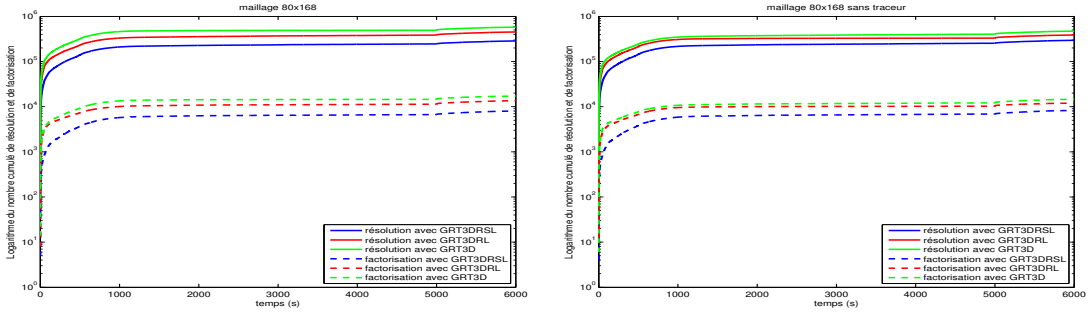


FIGURE 5.30 – Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 80×168 (À gauche : avec traceur, à droite : sans traceur)

Nous remarquons que les nombres de résolutions et de factorisations ont le même comportement que le nombre de pas de temps. Il y a aussi une forte augmentation de ces nombres au début de chaque période. La réduction linéaire diminue le nombre de résolutions et le nombre de factorisations par rapport à la version GRT3D. L'utilisation des concentrations sans logarithmes donne un gain fort par rapport aux autres versions. D'autre part, le nombre de factorisations est nettement plus faible que le nombre de résolutions. En effet, le Jacobien est gelé dans les itérations de Newton et entre les pas de temps. Cette technique permet de réduire le temps de calcul des factorisations.

5.8.2 Temps de calcul des résolutions et factorisations

Dans les figures 5.31 à 5.34, nous montrons les temps de calcul des résolutions et des factorisations à chaque pas de temps externe pour chaque logiciel et pour chaque maillage :

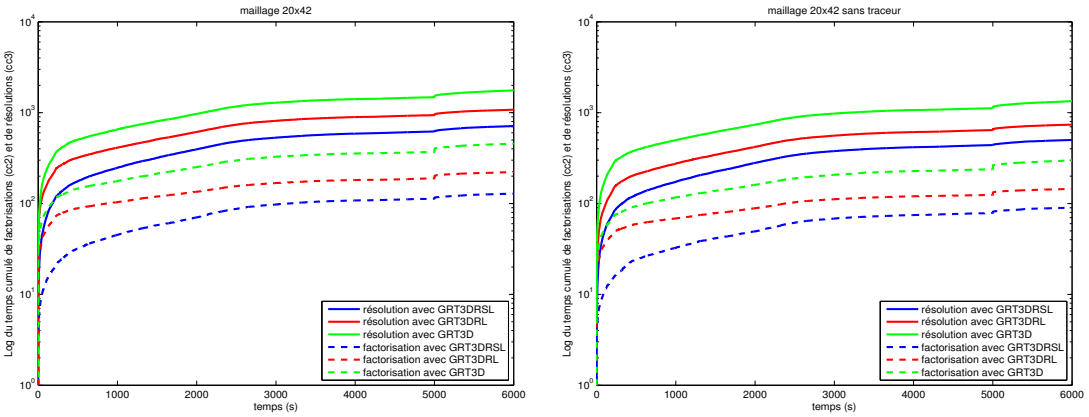


FIGURE 5.31 – Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 20×42 (À gauche : avec traceur, à droite : sans traceur)

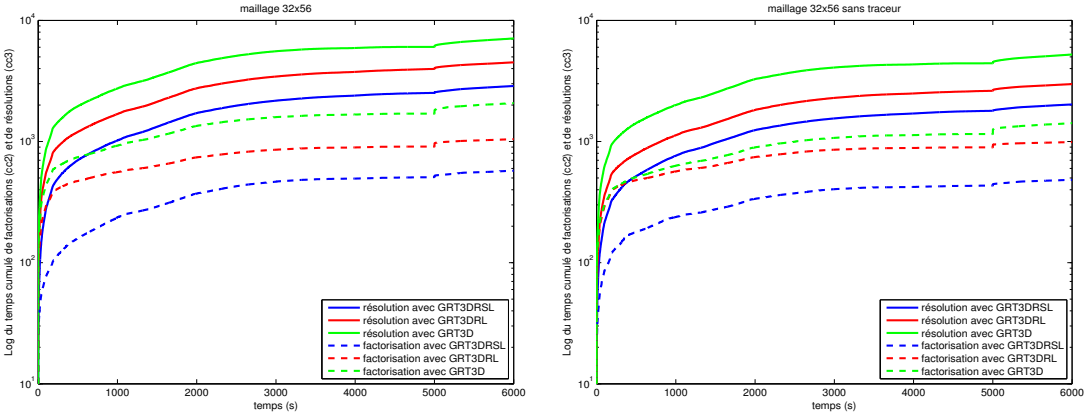


FIGURE 5.32 – Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 32×56 (À gauche : avec traceur, à droite : sans traceur)

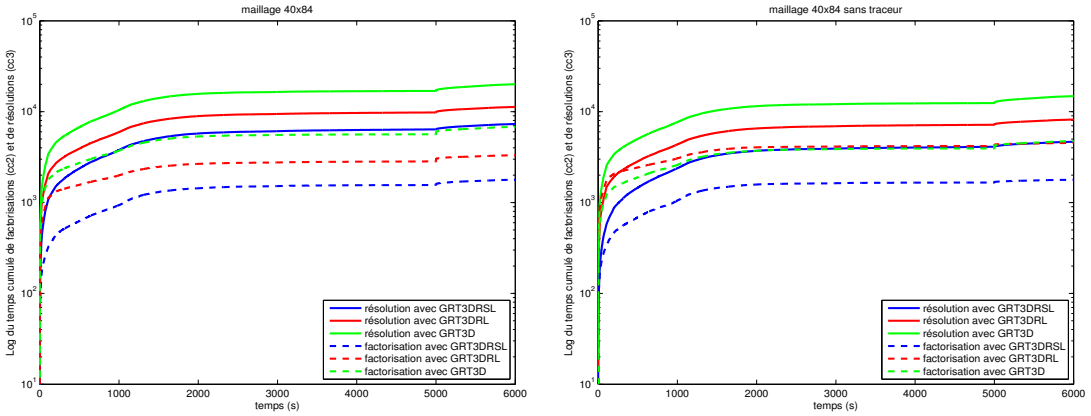


FIGURE 5.33 – Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 40×84 (À gauche : avec traceur, à droite : sans traceur)

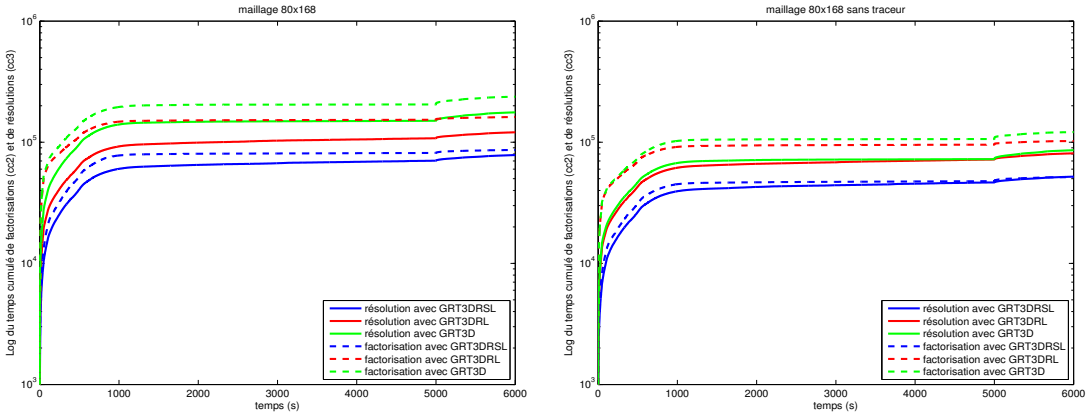
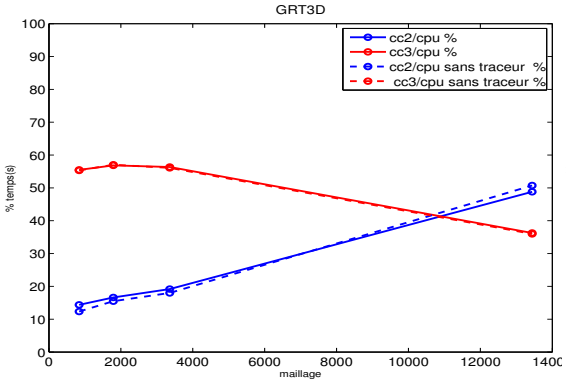


FIGURE 5.34 – Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 80×168 (À gauche : avec traceur, à droite : sans traceur)

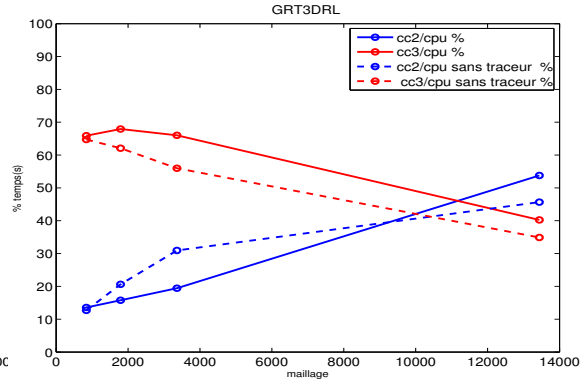
On remarque que le temps de résolution est plus grand que le temps de factorisation pour les trois logiciels avec et sans traceur mais on note le contraire dès que le maillage est assez fin.

5.8.3 Pourcentages des temps de calcul des étapes linéaires

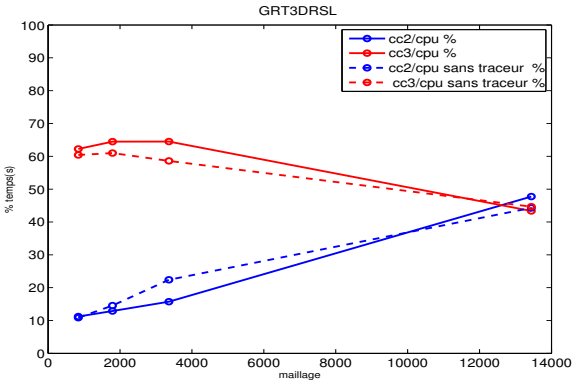
Dans la Figure 5.35, nous montrons la proportion du temps de résolution linéaire (cc3) et du temps de factorisation (cc2) sur le temps total pour chaque logiciel et pour chaque maillage. Ces temps sont mesurés à la fin des simulations.



(a) Le pourcentage du cc3 et du cc2 dans GRT3D



(b) Le pourcentage du cc3 et du cc2 dans GRT3DRL



(c) Le pourcentage du cc3 et du cc2 dans GRT3DRSL

FIGURE 5.35 – Le pourcentage du temps de résolution final et du temps de factorisation final par rapport au temps total

On remarque que, relativement au temps total, la part du temps de résolution diminue en raffinant le maillage, tandis que le pourcentage du temps de factorisation augmente. Les courbes se croisent dès que le maillage est assez fin.
En effet le temps de factorisation a une complexité plus élevée que le temps de résolution et augmente donc plus vite avec le nombre de mailles. On note aussi l'intérêt de geler le Jacobien pour réduire le nombre des factorisations.
Le temps de calcul total est dû principalement aux étapes linéaires : environ 70% pour le maillage grossier (dans GRT3D et avec traceur) et environ 90% avec le maillage le plus fin (dans GRT3DRSL et avec traceur).

5.9 Conclusion

Dans ce chapitre, nous avons montré les résultats numériques du cas test Benchmark MoMaS 2D concernant les trois versions de GRT3D. Nous avons comparé le conditionnement de la matrice jacobienne pour chaque logiciel et pour chaque maillage (avec traceur et sans traceur). Nous avons remarqué que l'élimination des logarithmes améliore le conditionnement de la matrice.

Nous observons l'efficacité du pas de temps adaptatif et de la mise à jour adaptative du Jacobien, mis en œuvre dans IDA. Ces techniques, combinées à la réduction de la taille du système et à la réduction du conditionnement, permettent d'accroître les performances. Ainsi, la nouvelle version GRT3DRSL est nettement plus rapide que la version d'origine GRT3D.
D'autre part, l'analyse des pourcentages de temps montrent que les étapes linéaires de factorisation et résolution constituent la majorité des calculs. La factorisation devient prépondérante pour le maillage fin.

Conclusion

Ce travail de thèse concerne le modèle chimique lié au transport réactif en milieux poreux. Ce modèle concerne les réactions à l'équilibre (sans les réactions cinétiques).

Dans le chapitre 1, nous avons présenté deux modèles chimiques : le modèle général où les minéraux peuvent être dissous complètement et le modèle simplifié où les minéraux sont toujours présents. Le modèle chimique général conduit à un système algébrique d'équations non linéaires avec des contraintes d'inégalité. Nous avons montré que la reformulation comme problème de complémentarité permet de supprimer les inégalités dans le modèle général et que la méthode semi-smooth est appropriée pour résoudre ce nouveau type de problème.
Ensuite, nous avons présenté le modèle chimique avec l'utilisation des logarithmes des concentrations pour assurer la positivité des concentrations et le modèle sans le passage par les logarithmes des concentrations avec des contraintes de positivité. Après avoir défini les deux modèles, nous avons écrit la matrice Jacobienne du système et nous avons étudié la singularité de cette matrice dans des cas particuliers.

Dans le chapitre 2, les équations d'écoulement et de transport, basées sur la loi de Darcy, la conservation de la masse, l'advection et la dispersion, ont été établies.
Les équations du transport sont couplées avec les équations du système chimique. Une discrétisation spatiale conduit à un système d'équations différentielles algébriques.
D'autre part, nous avons présenté les deux approches utilisées pendant ce travail : la méthode séquentielle non itérative (SNIA) et l'approche globale DAE (GDAE). Puisque la méthode GDAE est basée sur la méthode de Newton, nous avons donc écrit la matrice Jacobienne du système couplé.

Dans cette partie, nous avons proposé plusieurs réductions du système :
— la réduction de taille du système linéaire à chaque pas de temps interne de $(3N_c + N_s + N_p) \times N_m$ à $(N_c + N_s + N_p) \times N_m$, cette réduction est implémentée dans le logiciel GRT3DRL.
— la réduction de taille du système algébro-différentiel avec élimination d'inconnues C de $(3N_c + N_s + N_p) \times N_m$ à $(2N_c + N_s + N_p) \times N_m$.
— la réduction de taille du système non linéaire avec élimination d'inconnues C et T de $(3N_c + N_s + N_p) \times N_m$ à $(N_c + N_s + N_p) \times N_m$.

Nous avons présenté les logiciels suivants dans le chapitre 3 :
— la version SIA qui implémente la méthode implicite SIA.
— la version GDAE1D qui implémente la méthode GDAE en 1D.
— la version GDAE3D (GRT3D) qui implémente la méthode GDAE en 1D, 2D et 3D avec l'utilisation de MT3D et MODFLOW.
— la version GRT3DRL qui réduit la taille du système résolu à chaque pas de temps.
— la version GRT3DRSL qui réduit la taille du système résolu à chaque pas de temps et sans le passage par les logarithmes des concentrations.
— la version SNIA qui implémente la méthode explicite SNIA.
— la version GRT3DRSLC qui implémente la méthode GDAE sans le logarithme des concentrations avec le modèle général (problème de complémentarité).

Pour chaque version, nous avons présenté l'installation, les fichiers d'entrée, les fichiers de sortie et les logiciels libres utilisés. Ensuite, nous avons validé les versions SIA, GDAE1D et GDAE3D pour des cas simples et nous avons remarqué qu'il n'y avait pas de grande différence entre les résultats de GDAE1D et ceux de GDAE3D.

Nous avons présenté dans le chapitre 4 un cas test proposé par l'Andra dans un domaine 2D avec advection et dispersion données. Le système contient dans ce cas un traceur (composant qui ne réagit pas chimiquement avec les autres espèces) ; dans ce cas, il est possible de calculer la solution analytique et de la comparer aux résultats de GRT3D et SNIA. Nous avons comparé aussi la convergence et la mémoire RAM des deux versions et nous avons conclu que la méthode globale GDAE est plus rapide et précise que la méthode séquentielle non itérative SNIA. En revanche, elle prend plus de mémoire pour une faible précision, sans toutefois dépasser 1 Go.

À la fin de ce travail, au chapitre 5, nous avons étudié un deuxième cas test, plus compliqué que le premier cas, proposé par le groupe MoMas : le cas test "Easy test case" en 2D avec deux milieux et deux périodes de temps. Nous avons comparé les résultats des trois logiciels GRT3D, GRT3DRL et GRT3DRSL avec traceur et sans traceur qui montrent que les trois logiciels donnent les mêmes résultats. Ensuite, nous avons présenté les temps de calculs de chaque version et nous avons conclu que GRT3RSL est plus rapide que GRT3RL, qui est plus rapide que GRT3D. Enlever le traceur accélère aussi les calculs. Les causes des meilleures performances sont multiples : la taille du système, le conditionnement de la matrice Jacobienne, le nombre d'échecs du test d'erreur. Geler le Jacobien est très efficace car il réduit fortement le nombre de factorisations. Ainsi le temps de résolution est relativement plus important que le temps de factorisation pour des maillages assez grossiers. La tendance s'inverse pour des maillages fins.

Comme perspective, dans le but d'améliorer la méthode GDAE, on peut implémenter les deux réductions mentionnées dans le chapitre 2 :
— réduction du système différentiel avec l'élimination de C
— réduction du système non linéaire avec l'élimination de T

Dans le chapitre 1, nous avons développé le problème de complémentarité et la méthode semi-smooth et nous avons implémenté cette méthode dans le logiciel GRT3DRSLC. Nous devons maintenant ajouter les contraintes de positivité de la deuxième variable dans la fonction min. Il serait intéressant de tester cette version avec des cas tests à plusieurs espèces précipitées.

Un des enjeux est maintenant de simuler un transport réactif dans des milieux 3D. En exploitant la structure tensorielle du Jacobien, nous prévoyons de développer des algorithmes parallèles, pour gagner du temps de calculs et résoudre du plus gros problèmes.

Annexes

A Installation de GRT3D

A.1 Contrôle de version

Subversion est un système de gestion de versions qui permet de stocker des fichiers en conservant toutes les modifications qui ont été faites. Nous avons placé l'ensemble des fichiers sources sous contrôle de version (SVN) avec un test pour chaque branche.

1. SIA1D
2. GDAE1D
3. GDAE3D
4. GRT3DRL
5. GRT3DRSL
6. TEST
7. SNIA

A.2 Variables d'environnement

Définir les variables d'environnements : LIB_EXT, MPI_ROOT, PATH et PGPLOT_DIR .

— LIB_EXT contient le chemin du répertoire d'installation.
— PATH contient les répertoires dans desquels le shell cherche la commande qu'on écrit au clavier, on ajoute le chemin vers l'exécutable MPI.
— MPI_ROOT est le chemin de répertoire qui contient les fichiers include et les bibliothèques de MPI.
— PGPLOT_DIR est le chemin de répertoire qui contient les librairies de pgplot.

A.3 Sundials-2.3.0 [1] et Suitesparse

On télécharge sundials-2.3.0 et SuiteSparse et on fait l'installation sans modification.

A.4 Sundials_sparse-2.3.0 et Sundials_sparse_reduction-2.3.0

1. entrer dans $LIB_EXT/../build/

2. copier les librairies Sundials_sparse-2.3.0 et Sundials_sparse_reduction-2.3.0 avec la commande :

   ```
   svn checkout svn+ssh://login@scm.gforge.inria.fr/svn/grt3d/lib_ext/* .
   ```

   On remplace login et on adapte le makefile de chaque sundials (CC, CFLAGS, les répertoires de chaque librairies)

3. On se place dans Sundials_sparse-2.3.0 (ensuite Sundials_sparse_reduction-2.3.0) et on fait :

   ```
   make
   make install
   ```

A.5 PGPLOT

Télécharger PGPLOT et faire l'installation

— En 64 bits, on change integer par integer*8 ligne 81 du fichier gidriv.f dans /pgplot/drivers/

— Il faut remplacer XW_DEF_COLORS 100 par XW_DEF_COLORS 256, ligne 52 du fichier pgxwin_server.c dans /pgplot/drivers/

A.6 SIA1D, GDAE1D, GDAE3D, GRT3DRL, GRT3DRSL ...

```
svn checkout svn+ssh://login@scm.gforge.inria.fr/svn/grt3d/tags/version-1.0/* .
```

On remplace login et on adapte le makefile (CC, CFLAGS, les répertoires de chaque librairie)

```
make
```

Remarque 2 Il est possible d'utiliser un script qui fait l'installation automatique et qui remplace les points A.3 à A.5 mais il reste à faire des make dans les répertoires SIA1D, GDAE1D, GDAE3D .

A.7 MODFLOW-96 et MT3D-5.2.0

Télécharger MODFLOW et MT3D et faire les modifications indiquées dans la section 3.3.

B Les fichiers d'entrée

Pour utiliser le code, on a besoin de fichiers d'entrée (pour lire les paramètres de chimie et de transport) avec un ordre spécial.

B.1 Fichiers d'entrée de GRT3D

1. **chemistry.dat** (Pour toutes les versions)
   (a) Partie commune
   — nc, ns, np, mx, my (nombre des composants aqueux, fixés, précipités, les espèces secondaires aqueuses et fixées)
   — Scoef (de taille mx, nc) : matrice des coefficients stœchiométriques entre espèces secondaires aqueuses et composants aqueux.
   — Acoef (de taille my, nc) : matrice des coefficients stœchiométriques entre espèces secondaires fixées et composants aqueux.
   — Bcoef (de taille my, ns) : matrice des coefficients stœchiométriques entre espèces secondaires fixées et composants fixés.
   — Dcoef (de taille np,nc) : matrice des coefficients stœchiométriques entre espèces précipitées et composants aqueux.

   (b) Partie supplémentaire

   Pour GDAE1D,GDAE3D, SIA1D et GRT3DRL
   — logKx (de taille mx) : vecteur logarithme de constante de réaction relatif à l'espèce secondaire aqueuse
   — logKy (de taille my) : vecteur logarithme de constante de réaction relatif à l'espèce secondaire fixée
   — logKp (de taille np) : vecteur logarithme de constante de réaction relatif à l'espèce précipitée

   (c) Partie supplémentaire

   Pour GRT3DRSL, SNIA et GRT3DRSLC
   — Kx (de taille mx) : vecteur de constante de réaction relatif à l'espèce secondaire aqueuse
   — Ky (de taille my) : vecteur de constante de réaction relatif à l'espèce secondaire fixée
   — Kp (de taille np) : vecteur de constante de réaction relatif à l'espèce précipitée

2. **save.dat** (Pour GDAE3D, GRT3DRL, GRT3DRSL, SNIA et GRT3DRSLC)
   — nodes_graph : numéro des mailles de sauvegarde en temps. Si nodes_graph $= -1$, la sauvegarde est faite pour tous les points de maillage.
   — nt_print : nombre des pas de temps externes. Si nt_print< 0, la sauvegarde est effectuée à chaque pas de temps externe.
   — n_sauv : nombre des pas temps sauvegardés

- pour chaque sauvegarde i (1 à n_sauv) :
  - t_sauv[i] (temps auquel a lieu la sauvegarde)

3. **parameters.dat**
   (a) Cas SIA1D et GDAE1D
   - RTOL tolérance relative pour IDA
   - ATOL tolérance absolue pour IDA
   - FTOL tolérance pour KINSOL
   - STEPTOL tolérance pour KINSOL

   (b) Cas SNIA
   - condition initiale sur les concentrations totales à chaque composant
   - FTOL tolérance pour KINSOL
   - STOL tolérance pour KINSOL

   (c) Cas GDAE3D, GRT3DRL, GRT3DRSL et GRT3DRSLC
   - FTOL tolérance pour KINSOL
   - STOL tolérance pour KINSOL
   - condition initiale sur les concentrations totales à chaque composant
   - RTOL tolérance relative pour IDA
   - ATOL tolérance absolue pour IDA
   - maxcor : nombre maximum d'itérations de Newton
   - maxord : ordre maximum de la méthode BDF
   - maxnef : nombre maximum de erreur test failure
   - maxncf : nombre maximum de convergence failure
   - mxsteps : nombre maximum de pas de temps interne
   - epsnewt : coefficient du test de convergence non linéaire

4. **time.dat** (Pour SIA1D et GDAE1D)
   - dt : pas de temps
   - nperiod : nombre des périodes
   - pour chaque période i :
     - t_bound[i] : temps de chaque période
     - C_bound_t_t[i] (nc éléments pour chaque zone) : condition initiale de chaque période
     - type[i] : type de condition

5. **space.dat** (Pour SIA1D et GDAE1D)
   - c, s, p (de taille nc+ns+np)
   - nzone : nombre des zones
   - pour chaque zone i :
     - $xl[i]$: début de la zone, $xr[i]$: la fin de la zone, poros[i] : la porosité, $a[i]$: vitesse de Darcy, $D[i]$: tenseur de dispersion
     - $T[i]$: condition initiale sur les concentrations totales en composant aqueux (de taille nc)
     - $W[i]$: condition initiale sur les concentrations totales en composant fixé (de taille ns)

- dx : pas d'espace
- gamma : coefficient du schéma en espace.

6. **graphic.dat** (Pour SIA1D et GDAE3D)
   - noms des légendes.
   - nodes_graph.
   - nt_print.
   - n_sauv.
   - pour chaque sauvegarde i (1 à n_sauv) :
     - t_sauv[i] (temps auquel a lieu la sauvegarde).
7. **A_RHS.dat (ou A_RHS.bin)** (pour toutes les versions sauf SIA1D et GDAE1D)
   Ils sont créés par MT3D.
   - nperiod (nombre périodes),ncomp (nombre des composants total(nc+np+ns)), mcomp (nombre des composants aqueux), nstp_tot (nombre des pas de temps).
   - nlay (nombre des points sur l'axe des abscisses), nrow (nombre des points sur l'axe des ordonnées), ncol (nombre des points sur l'axe des hauteurs), mt3d_nodes (nombre des points de maillages).
   - nstp nombre des pas temps à chaque période .
   - dt les pas de temps .
   - nz_row : nombre des éléments non nuls dans la matrice.
   - $A[i, j]$: matrice de transport (pour chaque point de maillage de maillage i, j nombre de colonne non nulle de la matrice de transport).
   - $L[j]$: matrice des indices des éléments non nuls de la matrice de transport.
   - icbund[i] : condition au bord à chaque point de maillage i.
   - $rhs[i]$: second membre (pour chaque période et chaque composant et chaque point de maillage i).
   - $cold[i]$: concentration (chaque composant et chaque point de maillage i).
   - $poros[i]$: porosité (pour chaque point de maillage i).
8. **DXDYDZ.dat (ou DXDYDZ.bin)** (pour toutes les versions sauf SIA1D et GDAE1D)
   Ils sont créés par MT3D.
   - pt_nlay, pt_nrow, pt_ncol, pt_nodes (définis ci-dessus).
   - $i, xx(i), yy(i), zz(i)$ (pour chaque point).

B.2 Fichiers d'entrée de MODFLOW

1. bas : (packages Basic)
   - nom de cas test
   - nlay × nrow
   - Nlay Nrow Ncol Nper ITMUNI
     avec ITMUNI : indique l'unité de temps
     0 - non défini 3 - heure
     1 - seconde 4 - jour
     2 - minutes 5 - année

— Option
  * XSECTION : indique que la dimension du problème.
  * CHTOCH : indique que le flux entre les cellules adjacentes à charge constante devrait être calculé.
  * FREE : indique que le format utilisé pour les paramètres d'entrée dans le fichier Basic est libre.

— IAPART ISTRT
  IAPART : indique si le tableau BUFF est séparé de RHS.
  * Si IAPART = 0, tableau BUFF et RHS occupent même espace.
  * Si IAPART $\neq$ 0, tableau BUFF et RHS occupent un espace différent.

  ISTRT : indique les charges initiales doivent être gardées en mémoire tout au long de la simulation.
  * Si ISTRT = 0, les charges initiales ne sont pas conservées.
  * Si ISTRT $\neq$ 0, les charges initiales sont conservées.

— IBOUND(NCOL, NROW) ou (NCOL, NLAY)est la variable de la frontière
  * Si IBOUND $(J, I, K) < 0$, la cellule J, I, K a une charge constante.
  * Si IBOUND $(J, I, K) = 0$, la cellule J, I, K n'a pas de flux.
  * Si IBOUND $(J, I, K) > 0$, la cellule J, I, K est de charge variable.

— HNOFLO : valeur de charge qui est attribuée à toutes les cellules sans écoulement.

— Shead (NCOL, NROW) ou (NCOL, NLAY) : champ initial.

— PERLEN NSTP TSMULT
  PERLEN : longueur d'une période.
  NSTP : nombre de pas de temps dans une période.
  TSMULT : multiplicateur de la longueur de pas de temps successifs.
  la longueur du premier pas de temps est :

$$\Delta_1 = PERLEN * (TSMULT - 1)/(TSMULT^{NSTP} - 1)$$

2. OC (Packages de Contrôle Option)

— IHEDFM, IDDNFM, IHEDUN, IDDNUN
  Pour chaque pas de temps :

— INCODE, IHDDFL, IBUDFL, ICBCFL

— HDPR, DDPR, HDSV, DDSV

avec :

- IHEDFM : : code pour le format dans lequel les champs seront imprimés.

0 - (10G11.4)	10 - (20F5.3)
1 - (11G10.3)	11 - (20F5.4)
2 - (9G13.6)	12 - (10G11.4)
3 - (15F7.1)	13 - (10F6.0)
4 - (15F7.2)	14 - (10F6.1)
5 - (15F7.3)	15 - (10F6.2)
6 - (15F7.4)	16 - (10F6.3)
7 - (20F5.0)	17 - (10F6.4)

8 - (20F5.1) 18 - (10F6.5)
9 - (20F5.2)

- IDDNFM : code pour le format dans lequel tirages seront imprimés.
- IHEDUN : numéro d'unité sur lesquels les champs seront sauvegardés.
- IDDNUN : code pour le format dans lequel tirages seront sauvegardés.
- INCODE : code pour la lecture des éléments de ligne 3.
  * si INCODE < 0 alors les éléments de ligne 3 n'est pas lu.
  * si INCODE=0 les éléments de ligne 3 sera composé d'un enregistrement.
  * si INCODE > 0 les éléments de ligne 3 sera composé d'un enregistrement pour chaque couche.
- IHDDFL : indicateur de sortie de charge et de tirage.
  * Si IHDDFL $= 0$, pas de charge ou les prélèvements seront imprimées ou sauvegardées indépendamment des éléments de ligne 3.
  * Si IHDDFL $\neq 0$, les charges et les tirages seront imprimés ou sauvegardés selon l'élément 3
- IBUDFL : indicateur d'impression du budget.
  * Si IBUDFL $= 0$, le budget global volumétrique ne sera pas imprimé.
  * Si IBUDFL $\neq 0$, le budget global volumétrique sera imprimé.
- ICBCFL : cellule par cellule à flux terme flag.
  * Si ICBCFL $= 0$, les termes de flux ne sont pas enregistrés ou imprimés cellule par cellule.
  — Si ICBCFL $\neq 0$, les termes de flux sont imprimés ou enregistrés sur le disque en fonction de flag cellule par cellule.
- Hdpr : flag de sortie pour l'impression de charge.
  * Si Hdpr $= 0$, charge n'est pas imprimé pour la couche correspondante.
  * Si Hdpr $\neq 0$, charge est imprimé pour la couche correspondante.
- Ddpr : flag de sortie pour l'impression de tirage.
  * Si Ddpr $= 0$, tirage n'est pas imprimé pour la couche correspondante.
  * Si Ddpr $\neq 0$, tirage est imprimée pour la couche correspondante.
- Hdsv : flag de sortie pour sauver charge.
  * Si Hdsv $= 0$, charge n'est pas sauvé pour la couche correspondante.
  * Si Hdsv $\neq 0$, charge est sauvée pour la couche correspondante.
- DDSV : flag de sortie pour sauver rabattement.
  * Si DDSV $= 0$, tirage n'est pas sauvé pour la couche correspondante.
  * Si DDSV $\neq 0$, tirage est enregistrée pour la couche correspondante.

3. PCG : (Packages de gradient conjugué)
   — MXITER, ITER1, NPCOND
   — HCLOSE, RCLOSE, RELAX, NBPOL, IPRPCG, MUTPCG, DAMPPCG
     - MXITER : nombre maximum d'itérations extérieures, si problème linéaire MXITER est 1
     - NPCOND : flag utilisé pour sélectionner la méthode de conditionnement de matrice :
       * 1 : Cholesky incomplète de modification.
       * 2 : polynomiale.

- ITER1 : nombre d'itérations intérieure. Pour les problèmes non linéaires, ITER1 varie habituellement de 10 à 30
- HCLOSE : Critère de changement charge pour la convergence.
- RCLOSE : Critère de changement résiduel pour la convergence.
- RELAXE : Facteur de relaxation (uniquement utilisé si INPCOND 1).
- NBPOL : utilisé quand NPCOND=2, pour indiquer si l'estimation de la borne supérieure de la valeur propre maximale est de 2,0.
- IPRPCG : l'intervalle d'impression pour le PCG, si il est égale zéro sera changé à 999.
- MUTPCG : un indicateur qui contrôle l'impression d'informations à partir de convergence du solveur :
  0 - est pour l'impression des tableaux de changement de charge maximale et résiduelle chaque itération.
  1 - c'est l'impression que le nombre total d'itérations.
  2 - est pour l'impression sans...
  3 - est pour l'impression que si la convergence échoue.
- DAMPPCG : facteur d'amortissement.

4. BCF : (Block Centered Flow)
   — ISS IBCFCB HDRY IWDFLG WETFCT IWETIT IHDWET
   - ISS : état stable.
     * si ISS> 0 la simulation est l'état d'équilibre
     * si ISS=0 la simulation est transitoire
   - IBCFCB : un numéro d'unité.
     * Si IBCFCB> 0, c'est le nombre unité à laquelle la cellule par cellule termes de flux seront écrites quand "SAVE BUDGET".
     * Si IBCFCB = 0,cellule par cellule les termes de flux ne seront pas écrites
     * Si IBCFCB $<$ 0,cellule par cellule les termes de flux qui sont enregistrés sont le stockage, à charge constante du débit et entre les cellules adjacentes ne seront pas écrits dans n'importe quel fichier.
   - HDRY : charge qui est attribué à des cellules qui sont convertis pendant une simulation.
   - IWDFLG : un indicateur qui détermine si la capacité de mouillage est actif.
     * Si IWDFLG = 0, la capacité de mouillage est inactif.
     * Si IWDFLG $\neq$ 0, la capacité de mouillage est actif.
   - WETFCT : facteur qui est inclus dans le calcul de charge qui est initialement établi à une cellule où il est converti à sec à humide.
   - IWETIT : intervalle d'itération pour tenter de cellules humides.
   — Ltype (NLAY) : code combiné spécifiant le type de couche (LAYCON) et la méthode de calcul de la conductance inter blocs
   0 ou blanc - moyenne harmonique (la méthode utilisée dans MODFLOW-88).
   1 - la moyenne arithmétique
   2 - moyenne logarithmique

3 - la moyenne arithmétique de l'épaisseur saturée et la transmissivité moyenne logarithmique

— TRPY (NLAY) : tableau à une dimension contenant un facteur d'anisotropie pour chaque couche
— DELR (NCOL) : largeur de la cellule le long des lignes. Lire une valeur pour chacune des colonnes ncol
— DELC (NROW) : largeur de la cellule le long des colonnes. Lire une valeur pour chacune des lignes nrow.
— Sf1 (NCOL, NROW) : coefficient de stockage primaire (si ISS=0)
— Tran (NCOL, NROW) : transmissivité le long des rangées (si LAYCON est 0 ou 2).
— HY (NCOL, NROW) : conductivité hydraulique le long des rangées(si LAYCON est 1 ou 3).
— BOT (NCOL, NROW) : élévation du fond aquifère (si LAYCON est 1 ou 3).
— Vcont (NCOL, NROW) : conductivité hydraulique verticale divisée par l'épaisseur d'une couche à la couche inférieure (Si ce n'est pas la couche de fond).
— Sf2 (NCOL, NROW) : coefficient de stockage secondaire (si ISS=0 et LAYCON est 2 ou 3)
— TOP (NCOL, NROW) : altitude du sommet des aquifères (si LAYCON est 2 ou 3).
— WETDRY (NCOL, NROW) : combinaison du seuil de mouillage et d'un flag pour indiquer où les cellules voisines peuvent provoquer une cellule de devenir humide (si IWDFLG n'est pas 0 et LAYCON est 1 ou 3).

5. nam :

LIST	6	xx.lst (fichier sortie)
BAS	1	xx.bas
BCF	11	xx.bcf
PCG	19	xx.pcg
OC	20	xx.oc
LMT	22	xx.ftl (fichier sortie)

B.3 Fichiers d'entrée de MT3D

1. adv
   — MIXELM, PERCEL, MXPART, NADVFD
   — ITRACK, WD
   — INTERP, NLSINK, NPSINK
   — DCHMOC
   - MIXELM est l'option de la solution d'advection
     * si IMXELM=0 méthode standard des différences finies.
     * si IMXELM=1 méthode de caractéristique (MOC).
     * si IMXELM=2 méthode de caractéristique modifiée (MMOC).
     * si IMXELM=3 méthode hybride de caractéristique (HMOC).
     * si IMXELM=-1 méthode variation totale diminue (TVD).

- PERCEL nombre de Courant i.e le nombre de cellules advection sera autorisé dans toutes les directions en une seule étape de transport.
  * pour la méthode différence finis implicite ou particulière tracking based schemes PERCEL> 1.
  * pour la méthode différence finis explicite ou la méthode variation totale diminue (TVD) PERCEL est également une contrainte de stabilité alors, PERCEL< 1.
- MXPART nombre total maximum de particules en mouvement, si MIXELM = 1 ou 3.
- NADVFD entier indiquant quel système de pondération doit être utilisé
  * NADVFD=0 ou 1 si méthode de upstream weighting.
  * NADVFD=2 si centrale dans l'espace de weighting
- ITRACK indique l'algorithme de tracking particules est sélectionné pour les méthodes Eulérienne-lagrangiennes.
  * ITRACK = 1, l'algorithme de premier ordre d'Euler est utilisée ;
  * ITRACK = 2, le quatrième ordre de Runge-Kutta algorithme est utilisé.
  * ITRACK = 4, le 1 er hybride et algorithme de 4 ème ordre est utilisé.
- WD facteur de weighting de concentration entre 0,5 et 1.
- INTERP indique la méthode d'interpolation de concentration pour les utiliser dans le schéma MMOC.
- NLSINK indique si le modèle aléatoire ou fixe est sélectionné dans le schéma de MMOC.
- NPSINK le nombre de particules utilisé pour les cellules couler approximative dans le schéma de MMOC.
- DCHMOC gradient de concentration critique relative pour contrôler l'utilisation sélective de deux GPM ou MMOC dans le schéma de solution HMOC.

2. dsp :
   — AL (NCOL, NROW) le coefficient de dispersion longitudinale (alpha L)
   — TRPT (NLAY) tableau 1D réal définir ratio de la dispersion transversale horizontale.
   — TRPV (NLAY) tableau 1D réal définir ratio de la dispersion transversale vectorielle.
   — DMCOEF (NLAY) coefficient de diffusion effectif moléculaires

3. gcg :
   — MXITER, ITER1, ISOLVE, NCRS
   — ACCL, CCLOSE, IPRGCG
   - MXITER nombre maximum d'itérations extérieures.
   - ITER1 nombre maximum d'itérations intérieures.
   - ISOLVE type de pré-conditionneurs pour être utilisé avec le système d'accélération de Lanczos / ORTHOMIN
     * si ISOLVE= 1 : Jacobi
     * si ISOLVE = 2 : SSOR
     * si ISOLVE = 3 : modification de Cholesky incomplète (MIC)

- NCRS flag entier pour le traitement des termes de dispersion tenseur
  * NCRS=0 regrouper tous les termes croisés de dispersion de la droite-côté.
  * NCRS=1 tenseur de dispersion incluent complet.
- ACCL facteur de relaxation pour l'option SSOR.
- CCLOSE critère de convergence en termes de concentration relative.
- IPRGCG intervalle pour l'impression des changements de concentration maximale de chaque itération.

4. ssm :
   — FWEL, FDRN, FRCH, FEVT, FRIV, FGHB
   — MXSS
   Pour chaque période :
   — INCRCH
   Pour chaque espèce :
   — CRCH (NCOL, NROW) NSS
   — si NSS> 0 : KSS, ISS, JSS, CSS, ITYPE
   - FWEL : flag logique pour l'option bien ;
   - FDRN : flag logique pour l'option de vidange ;
   - FRCH : flag logique pour l'option de recharge ;
   - FEVT : flag logique pour l'option évapo-transpiration ;
   - Friv : flag logique pour l'option de la rivière ;
   - FGHB : flag logique pour l'option générale Boundary-Head-dépendant ;
   - FNEW : flagx logiques réservés pour sink supplémentaires / options de source.
   - MXSS : nombre maximum de tous les points sink et sources et inclus dans le modèle d'écoulement.
   - INCRH : flag indiquant si un tableau contenant la concentration des flux
   - CRCH : concentration de flux de recharge pour une espèce particulière.
   - NSS : nombre des points de sources dont les concentrations doivent être spécifiés.
   - KSS, ISS, JSS : indices de cellules (couche, ligne, colonne) des points de sources dont les concentrations doivent être spécifiés.
   - CSS : concentration source spécifiée ou la masse de chargement de taux
   - ITYPE : entier indiquant le type de la source ponctuelle comme indiqué ci-dessous :
     * ITYPE = 1, à charge constante de cellule ;
     * ITYPE = 2, bien ;
     * ITYPE = 3, fuite
     * ITYPE = 4, d'une rivière (ou flux) ;
     * ITYPE = 5, la cellule limites générales charge-dépendant ;
     * ITYPE = 15, masse chargement à la source ;
     * ITYPE = -1, Constante concentration cellulaire.

5. btn :
   — HEADNG(1)

— HEADNG(2)
— NLAY, NROW, NCOL, NPER, NCOMP, MCOMP
— TUNIT, LUNIT, MUNIT
— ADV DSP SSM RCT GCG
— LAYCON (NLAY)
— DELR (NCOL)
— DELC (NROW)
— HTOP (NCOL, NROW)
— DZ (NCOL, NROW)
— PRSITY (NCOL, NROW)
— ICBUND (NCOL, NROW)
For each species
- SCONC (NCOL, NROW)
— CINACT, THKMIN
— FMTCN, IFMTNP, IFMTRF, IFMTDP, SAVUCN
— NPRS
— NOBS, NPROBS
— CHKMAS, NPRMAS
FOR EACH STRESS PERIOD :
- PERLEN, NSTP, TSMULT
- DT0, MXSTRN, TTSMULT, TTSMAX

avec
- HEADNG(1) : première ligne de tout titre de la simulation.
- HEADNG(2) : deuxième ligne indique le numéro de maillage.
- NLAY : nombre total de couches.
- NROW : nombre total de lignes.
- NCOL : nombre total de colonnes.
- NPER : nombre des périodes.
- NCOMP : nombre total d'espèces chimiques.
- MCOMP : nombre total d'espèces chimiques mobiles. (MCOMP $\leq$ NCOMP).
- TUNIT : nom de l'unité de temps.
- LUNIT : nom de l'unité de longueurs.
- MUNIT : nom de l'unité de Masses.
- ADV : advection si il y a on met T sinon F.
- DSP : dispersion si il y a on met T sinon F.
- SSM : sink et source mélange si il y a on met T sinon F.
- RCT : réaction chimique si il y a on met T sinon F.
- GCG : généralisée gradient conjugué si il y a on met T sinon F.
- LAYCON (NLAY) : entier 1D indiquant le type de couches du modèle.
  * si LAYCON=0, a couche du modèle est confiné.
  * si LAYCON$\neq$ 0, a couche du modèle est ouverte.
- DELR (NCOL) : tableau 1D réel représentant $\Delta(x)$pour tous les NCOL.
- DELC (NROW) : tableau 1D réel représentant $\Delta(y)$pour tous les NROW.

- HTOP (NCOL, NROW) : tableau 2D définissant l'élévation supérieure de toutes les cellules de la couche
- DZ (NCOL, NROW) : tableau 1D réel représentant $\Delta(z)$pour tous les (NCOL, NROW).
- PRSITY (NCOL, NROW) : porosité
- ICBUND (NCOL, NROW) : condition au bord.
- SCONC (NCOL, NROW) : concentration initial.
- CINACT : valeur pour indiquer que la concentration inactive.
- THKMIN : épaisseur minimale saturés dans une cellule, par défaut est 0.01.
- FMTCN : flag indiquant si la concentration calculée doit être imprimée dans le fichier texte de sortie standard.
- IFMTNP : flag indiquant si le nombre de particules dans chaque cellule (entiers) doit être imprimée
- IFMTRF : flag indiquant si le modèle calcule facteur de retard devraient être imprimés.
- IFMTDP : flag indiquant si le modèle calculé, coefficient de dispersion à distance pondérée doit être imprimée.
- SAVUCN : flag logique indiquant si la solution de concentration doit être enregistré dans un défaut non formaté (binaire) fichier nommé MT3Dnnn.UCN
- NPRS : flag indiquant la fréquence de la sortie
- NOBS : nombre de points d'observation au laquelle la concentration de chaque espèce seront sauvés à la fréquence spécifiée dans le MT3Dnnn.OBS
- NPROBS : entier indiquant quelle fréquence la concentration aux points d'observation spécifiées doivent être enregistrées dans le fichier d'observation MT3Dnnn.OBS
- CHKMAS : flag logique indiquant si un sommaire d'une ligne d'information du bilan de masse doit être imprimée,dans le MT3Dnnn.MAS
- NPRMAS : entier indiquant quelle fréquence l'information sur le budget de masse doivent être enregistrés dans le bilan MT3Dnnn.MAS
- PERLEN : longueur de la période.
- NSTP : nombre de pas de temps de la période.
- TSMULT : multiplicateur de la longueur de pas de temps
- DT0 : pas de discrétisation transports spécifié par l'utilisateur à chaque étape de temps
- MXSTRN : nombre maximum d'étapes de transport autorisé dans un pas de temps
- TTSMULT : multiplicateur pour les étapes successives de transport.
- TTSMAX : maximum pas de discrétisation transport.

6. nam

LIST	0	cas2.list
BTN	0	cas2.btn
ADV	0	cas2.adv
DSP	0	cas2.dsp

GCG	0	cas2.gcg
SSM	0	cas2.ssm
FTL	0	flow/cas2.ftl

C Exécution (en ligne de commande)

C.1 Exécution de MODFLOW et MT3D

1. On se place dans le répertoire concernant les fichiers d'entrées de modflow
2. l'emplacement de l'exécutable de modflow/mf96.exe
3. On écrit xx.nam où xx est le nom de l'exemple
4. On se place dans le répertoire concernant les fichiers d'entrées de mt3d
5. l'emplacement de l'exécutable de mt3d/mt3d
6. On écrit xxx.nam où xxx est le nom de l'exemple.
   On obtient A_RHS.dat ou (.bin) et DXDYDZ.bin ou (.dat) fichiers de sorties de mt3d.

C.2 Exécution de SIA1D

On peut exécuter le programme SIA1D de deux manières :

— Si on travaille dans le répertoire du code, alors on a besoin des fichiers avec un ordre spécial.

1. argv[1]=chemistry.dat
2. argv[2]= save.dat
3. argv[3]=parameters.dat
4. argv[4]=time.dat
5. argv[5]= space.dat
6. argv[6]= graphic.dat

Par exemple :

```
soufre [12:08]% SIA \
 ~/couplage/TEST-1D/cas_qualif/domaine_L=0.4/new_conc-400/chemistry.dat \
 ~/couplage/TEST-1D/cas_qualif/domaine_L=0.4/new_conc-400/save.dat \
 ~/couplage/TEST-1D/cas_qualif/domaine_L=0.4/new_conc-400/parameters.dat\
 ~/couplage/TEST-1D/cas_qualif/domaine_L=0.4/new_conc-400/time.dat \
 ~/couplage/TEST-1D/cas_qualif/domaine_L=0.4/new_conc-400/space.dat \
 ~/couplage/TEST-1D/cas_qualif/domaine_L=0.4/new_conc-400/graphic.dat
```

— Si on travaille dans le répertoire des fichiers d'entrée, alors

```
~/couplage/SIAbranch/SIA
( où on  trouve le code et SIA est le nom de l'exécutable)
Par exemple:
```

```
soufre:~/couplage/TEST-1D/cas_qualif/domaine\_L=0.4/new_conc-400/$ \
~/couplage/SIA/SIA
```

C.3 Exécution de GDAE1D

C'est comme SIA mais on remplace le nom de l'exécutable par coupling.
Par exemple :

—
```
soufre[12:29]% coupling \
 ~/couplage/TEST-1D/1aqueux/adv/input_files/chemistry.dat \
 ~/couplage/TEST-1D/1aqueux/adv/input_files/save.dat \
 ~/couplage/TEST-1D/1aqueux/adv/input_files/parameters.dat\
 ~/couplage/TEST-1D/1aqueux/adv/input_files/time.dat \
 ~/couplage/TEST-1D/1aqueux/adv/input_files/space.dat \
~/couplage/TEST-1D/1aqueux/adv/input_files/graphic.dat
```
—
```
soufre:~TEST-1D/1aqueux/adv/input_files$~/couplage/Global1D/coupling
```

C.4 Exécution de GDAE3D

Il y a deux manières d'exécuter le programme GDAE3D :

— Si on travaille dans le répertoire du code, alors on a besoin des fichiers avec un ordre spécial.

1. argv[1]=chemistry.dat
2. argv[2]=save.dat
3. argv[3]=parameters.dat
4. argv[4]=A_RHS.dat (ou.bin)
5. argv[5]=DXDYDZ.dat (ou.bin)
6. dans le cas binaire argv[6]= --binary

Par exemple : binaire

```
soufre[12:51]% Coupling2D \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/chemistry.dat \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/save.dat \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/parameters.dat \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/A_RHS.bin \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/DXDYDZ.bin --binary
```

Ascii :

```
soufre[12:51]% Coupling2D \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/chemistry.dat \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/save.dat \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/parameters.dat \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/A_RHS.dat \
 ~/couplage/TEST-MT3D/cas_qualif/21x14/DXDYDZ.dat
```

— Si on travaille dans le répertoire des fichiers d'entrée, alors
```
~/couplage/Global1D2D3D/Global2D --binary (pour les binaires)
~/couplage/Global1D2D3D/Global2D  (pour les Ascii)
Par exemple:
soufre:~/couplage/TEST-MT3D/cas_qualif/21x14/$ \
~/couplage/Global1D2D3D/Global2D --binary
ou
soufre:~/couplage/TEST-MT3D/cas_qualif/21x14/$ \
~/couplage/Global1D2D3D/Global2D
```

Pour les autres versions GRT3DRL, GRT3DRSL, SNIA et GRT3DRSLC, on change le nom de l'exécutable. Par exemple :

```
~/couplage/Global1D2D3DRL/Coupling2DRL (pour la versions GRT3DRL)\\
~/couplage/Global1D2D3DRSL/Coupling2DRSL (pour la versions GRT3DRSL)\\
~/couplage/Global1D2D3DRSLC/Coupling2DRSLC (pour la versions GRT3DRSLC)\\
~/couplage/SNIA-SL/SNIA (pour la versions SNIA)\\
```

D Fichiers de sortie

Après l'exécution d'un code, les résultats sont stockés dans des fichiers sortie. Il existe des fichiers communs aux points d'entrée et d'autres spécifiques

1. **parameters.out** (pour toutes les versions)
   (a) Cas SIA1D et GDAE1D
   for the global system
   — ATOL.
   — RTOL.
   for the equilibrate system
   — FTOL.
   — STOL.
   (b) Cas SNIA
   — On affiche les conditions initiales pour chaque composant.
   — ATOL.
   — RTOL.
   (c) Cas GDAE3D et les autres versions
   for the equilibrate system
   — FTOL.
   — STOL.
   for the global system
   — RTOL.
   — ATOL.
   — lecteur du nombre maximum d'itérations de Newton :maxcor.
   — lecteur de l'ordre maximum de la méthode BDF : maxord.
   — nombre maximum de erreur test failure : maxnef.

- lecteur du nombre maximum de convergence failure : maxncf.
- lecteur du nombre maximum de pas de temps interne : mxsteps.
- lecteur du coefficient . du test de convergence non linéaire : epsnewt.

2. **chemistry.out** (pour toutes les versions)
   - nombre de composants aqueux = nc.
   - nombre de composants fixés = ns.
   - nombre de précipités = np.
   - nombre d'espèces secondaires aqueuses = mx.
   - nombre d'espèces secondaires fixées = my.
     - valeur de Scoef (i j) :matrice de coefficients stœchiométriques entre espèces secondaires aqueuses et composants aqueux.
     - Scoef (i ; j)= Scoef[i][j] pour chaque espèce secondaire et composant aqueux .
     - valeur de Acoef (i j) : matrice de coefficients stœchiométriques entre espèces secondaire fixées et composants aqueux.
     - Acoef (i ; j)= Acoef[i][j] pour chaque espèce secondaire fixée et composant aqueux.
     - valeur de Bcoef (i j) : matrice de coefficients stœchiométriques entre espèces secondaires fixées et composants fixés.
     - Bcoef (i ; j) = Bcoef[i][j] pour chaque espèce secondaire fixée et composant fixé.
     - valeur de Dcoef (i j) : matrice de coefficients stœchiométriques entre espèces précipitées et composants aqueux.
     - Dcoef (i ;j)= Dcoef[i][j] pour chaque précipité et composant aqueux.
       (a) Cas SIA1D, GDAE1D,GDAE3D et GRT3DRL
         - valeur de logKx : logarithme de constante de réaction relatif à l'espèce secondaire aqueuse.
         - logKx (i) pour chaque espèce secondaire aqueuse.
         - valeur de logKy : logarithme de constante de réaction relatif à l'espèce secondaire fixées.
         - logKy (i) pour chaque espèce secondaire fixée.
         - valeur de logKp : logarithme de constante de réaction relatif à l'espèce précipité.
         - logKp (i) pour chaque précipité.

       (b) Cas GRT3DRSL, SNIA et GRT3DRSLC
         - valeur de Kx : la constante de réaction relatif à l'espèce secondaire aqueuse.
         - Kx (i) pour chaque espèce secondaire aqueuse.
         - valeur de Ky : la constante de réaction relatif à l'espèce secondaire fixées.
         - Ky (i) pour chaque espèce secondaire fixée.
         - valeur de Kp : la constante de réaction relatif à l'espèce précipité.
         - Kp (i) pour chaque précipité.

3. **global.out**
   - nb (nombre des points de sauvegarde).
   - nst (nombre de mesures prises par l'IDA).
   - nre (nombre d'appels à la fonction de résolution de l'utilisateur).
   - nsetups (nombre d'appels fonction de configuration).
   - netf (nombre d'échecs critère de l'erreur).
   - nniters (nombre d'itérations non linéaires préformées).
   - nncfails (nombre d'échecs de convergence non linéaire).
   - nje (nombre d'appels à la fonction jacobienne de l'utilisateur).
   - dt time.
   - nn1 (nombre d'appel de la fonction sparseInit).
   - nn2 (nombre d'appel de la fonction sparseFactor).
   - nn3 (nombre d'appel de la fonction sparseSolve).
   - cc1 (temps de calcul (fonction clock()) d'initialisation de la fonction sparseInit).
   - cc2 (temps de calcul (fonction clock()) de factorisation de la fonction sparseFactor).
   - cc3 (temps de calcul (fonction clock()) de résolution de la fonction sparseSolve).
   - CPU_time.
4. **save.out**
   - numéro de la maille auquel a lieu la sauvegarde (nodes_graph)= nodes_graph (sur nodes).
   - pas de temps d'affichage (nt_print)= nt_print.
   - nombre de sauvegarde (n_sauv)= n_sauv.
   - temps auxquels ont lieu les sauvegardes t_sauv= t_sauv.
5. **time.out** (pour SIA1D et GDAE1D)
   - pas de temps (dt) = dt.
   - nombre de période de temps (nperiod) = nperiod.
   - période i.
   - début de la période = t_bound (pour chaque période).
   - concentrations analytiques totales C à la limite gauche du domaine.
   - (nc= j) : C_bound_t.
   - type_bc.
   - fin de période = t_bound.
   - concentrations analytiques totales C à la limite gauche du domaine.
   - (nc= j) : C_bound_t[nperiod][j].
   - type_bc[nperiod].
   - numéro du pas de temps aux changements de zone $j-1/j$ nt[j] (pour chaque période).
   - valeur du temps :
     - au pas de temps i time[i].

6. **space.out** (pour SIA1D et GDAE1D)
   — nombre des zones= nzone.
   — zone i :
      — valeur de xl= $xl[i]$.
      — valeur de xr= $xr[i]$.
      — valeur de la porosité = poros[i].
      — valeur de a = $a[i]$.
      — valeur de D = $D[i]$.
      — valeur de T T[i]= $T[i]$ à chaque composant aqueux (de taille nc).
      — valeur de W W[i]= $W[i]$ à chaque composant fixé (de taille ns).
   — pas d'espace = dx.
   — gamma = gamma.
   — nombre de mailles *nodes*.
7. **global_outputcc001.dat** (pour SIA1D et GDAE1D) ou à la place de cc on met (ss, xx, yy, pp, CC)
   # point auxquels a lieu la sauvegarde : point
   t_coord, cc_t à chaque composant (ss_t, xx_t, yy_t, pp_t, CC_t)
8. **outputcc001.dat** (pour SIA1D et GDAE1D) ou à la place de cc on mets (ss, xx, yy, pp, CC) et avec i de 0 à nz_row
   # point auxquels a lieu la sauvegarde : point
   x_coord, cc_x à chaque composant (ss_x, xx_x, yy_x, pp_x, CC_x)
9. **sia.out** (pour SIA1D)
   nb, nst, nfe, kl, nniters, nbcfails, nbacktr, nje, n_chem, dt, t, cpu_time (définir avant).
10. **sauvegarde001.dat et sauvegarde001.vtk** (pour GDAE3D, GRT3DRL, GRT3DRSL et GRT3DRSLC, les autres fichiers avec un numéro incréments)
    — # fichier de sauvegarde au point : temps.
    — xx, yy, zz pour chaque point.
    — cc : concentration en composant aqueux (pour chaque composant aqueux).
    — ss : concentration en composant fixé (pour chaque composant fixé).
    — pp : quantité en espèce précipitée (pour chaque précipitée).
    — uu : concentration en espèce secondaire aqueuse (pour chaque espèce aqueuse).
    — vv : concentration en espèce secondaire fixée (pour chaque espèce fixée).
    — CC : partie aqueuse de la concentration totale TT (pour chaque composant aqueux).
    — TT : concentration totale pour chaque composant aqueux.
    Les $*$.vtk correspondent au format de paraview (visualiseur 2D et 3D). Pour les sauvegardes format 'dat' les quantités sont stockées verticalement mais dans le format 'vtk' elles sont horizontales.
11. **mt3d.out** (pour GDAE3D, GRT3DRL, GRT3DRSL et GRT3DRSLC)
    — nombre de période de temps (nperiod)= nperiod.
    — nombre de pas de temps (nstp total)= nstp_tot.
    — nombre de couche (nlay) = nlay.

— nombre de ligne (nrow) = nrow.
— nombre de colonne (ncol) = ncol.
— nombre de nuds de la matrice de mt3d = nodes_mt3d.
— pas de temps : dt.
— temps final= time_final.
— nombre d'élément non nuls par ligne nz_row.
— valeur de K (i j).
— K (i j)= K[i][j] : les éléments de matrice de transport non nulle à chaque point nodes_mt3d.
— valeur de L(i). L(i)= L[i].
— valeur de icbund(i)
— icbund(i)= pour chaque point nodes_mt3d.
— valeur de RHS(i j).
— RHS(i j)=RHS[i][j] pour chaque point nodes_mt3d et chaque composant et période.
— valeur de T_init (i).
— T_init (i)= T_init[i] pour chaque point nodes_mt3d et chaque composant.
— valeur de W_init (i).
— W_init(i)= W_init[i] pour pour chaque point nodes_mt3d et chaque composant.
— valeur de poros (i).
— poros(i)= poros[i] pour chaque point (nodes).

Bibliographie

[1] `https://computation.llnl.gov/casc/sundials/download/download.html`.

[2] `http://www.andra.fr/pages/fr/menu1/les-dechets-radioactifs/qu-est-ce-qu-un-dechet-radioactif-r-8.html`.

[3] `http://hydro.geo.ua.edu/mt3d`.

[4] J. Carrayrou A. Dimier C.J. Van Duijn M. Kern P. Knabner A. Bourgeat, S. Bryant. Benchmark - reactive transport. *GDR MoMaS*, 2006.

[5] C.M. Bethke. Geochemical reaction modeling : Concepts and applications. *Oxford university Press US*, 1996.

[6] C.M. Bethke. Geochemical and biogeochemical reaction modeling. *Oxford university Press US*, second edition.

[7] N. Bouillard. Développemnt de méthodes numérique pour le transport réactif. *PhD thésis, université de Provence*, 2006.

[8] J. Erhel C. de Dieuleveult and M. Kern. A global strategy for solving reactive transport equations. *Journal of Computational Physics, 228 :6395-6410*, 2009.

[9] J. Carrayrou. Modélisation du transport de solutés réactifs en milieux poreux saturé. *PhD thèsis, Université louis Pasteur de Starsbourg*, 2001.

[10] C. W. Davies. Ion association. *Butterworths, Washington DC*, 1962.

[11] G. de Marsly. Cours d'hydrologéologie. *Université Paris VI*, 2004.

[12] Caroline De Dieuleveult. Un modèle numérique global et performant pour le couplage géochimie-transport. *Université de Rennes 1*, 2008.

[13] A. Fischer. A special newton-type optimization method. *Optimization*, pages 269–284, 1992.

[14] A.J. Valocchi G.E. Hammonde and P.C. Lichtner. Application of jacobian-free newton-krylov with physics-based preconditioning to biochemical transport. *Advance in Water Resources,28 :359-379*, 2005.

[15] A.R. Felmy. GMIN. A computerized chemical equilibrium model using a constrained minimization of the gibbs free energy. *Technical report, Pacific Northwest Laboratory, Richland, WA (USA)*, 1990.

[16] S. Sabit J. Erhel and C. de Dieuleveult. Solving partial differential algebraic equations and reactive transport models. *axe-Coburg Publications*, 2013.

[17] V.Lagneau J. van der Lee, L. L. De Windt and P. P. Goblet. Moduleoriented modeling of reactive transport with hytec. *Computers and geosciences, 29 : 265-275*, 2003.

[18] E.O. Frind K. U. Mayer and D. W. Blowes. Multicomponent reactive transport modeling in variably saturated porous media using a generalized formulation for kinetically controlled reactions. *Water Resources Research, 38(9)*, 2002.

[19] M. Kojima and S. Shindo. Extensions of newton and quasi-newton methods to systems of pc equations. *Journal of Optimization Research Society of Japan, 29 : 352- 374*, 1986.

[20] S. Kräutle. *General multi-species reactive transport problems in porous media : efficient numerical approaches and existence of global solutions.* PhD thesis, University of Erlangen-Nuremberg, Habilitation Thesis, 2008.

[21] S. Kräutle and P. Knabner. A new numerical reduction scheme for fully coupled multicomponent transport-reaction problems in porous media. *Water Resources Research*, 41 :W09414, 2005.

[22] S. Kräutle. The semismooth newton method for multicomponent reactive transport with minerals. *Advance in Water Resources*, 2010.

[23] S. Kräutle and P. knabner. A reduction scheme for coupled multicomponent transport-reaction problem in porous media : generalization to problems with heterogeneous equilibrium reactions. *Water Resources Research,43 : W03429*, 2007.

[24] D. A. Kulik. Gibbs energy minimization approach to modelling sorption equilibria at the mineral-water interface : thermodynamic relations for the multi-site-surface complexation. *American Journal of science, 302 : 227-279*, 2002.

[25] K. U. Mayer. A numerical model for multicomponent reactive transport in variable saturated porous media. 1999.

[26] K. U. Mayer, E. O. Frind, and D. W. Blowes. Multicomponent reactive transport modeling in variably saturated porous media using a generalized formulation for kinetically controlled reactions. *Water Resources Research*, 2002.

[27] M.Fahs. Modelisation du transport reactif multicomposants en milieu poreux saturé. *PhD thésis, Université de Strasbourg*, 2007.

[28] F. Morel and J. G. Hering. Principles and application of aquatic chemistry. *Wiley-IEEE*, 1993.

[29] C. Ayora J. Carrera S. Olivella M.W. Saaltink, F. Batlle. RETRASO, a code for modeling reactive transport in saturated and unsaturated porous media. *Geologica Acta*, 2 :235–251, 2004.

[30] E. Deville E. Adam J. Gaombalet A. Bengaouer L. Loth P. Montarnal, A. Dimier and C. Chavant. Coupling methodology within the software platform alliances. *In E. Oñate M. Papadrakakis and B. Schrefler, editors, Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering COUPLED PROBLEMS 2005, Barcelona*, 2005.

[31] J. S. Pang. Newton's method for b-differentiable equation. *Mathematics of operations Research, 15 (2) : 311-341*, 1990.

[32] D. L. Parkhust and C. A. Appelo. User's guide to phreeqc (version 2)-a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. *Technical Report 99-4259, U.S. DEPARTMENT OF THE INTERIOR, U.S. GEOLOGICAL SURVEY*, 1999.

[33] K. S. Pitzer and L. Brewer. Revised edition of thermodynamics, by g. n. lewis and m. randall. *2nd ed. MacGraw-Hill, New York*, 1961.

[34] L. N. Plummer. Geochemical modeling of water ?rock interaction : past, present, future. in y. k. kharaka and a. s. maest (eds.). *Water ?Rock Inter-action. Balkema, Rotterdam, pp. 23 ?33*, 1992.

[35] L. Qi. Convergence analysis of some algorithms for solving nonsmoth equation. *Mathematics of operations Research, 18(1) : 227-244*, 1993.

[36] J. Rubin. Transport of reacting solutes in porous media : Relation between mathematical nature of problem formulation and chemical nature of reactions. *Water Resources Research, Volume 19, Issue 5, pp. 1231-1252*, 1983.

[37] C. Ayora S. Molins, J. Carraera and M. W. saaltink. A formulation for decoupling components in reactive transport problems. *Water Resources Research,40 : W10301*, 2004.

[38] S. Sabit and N. Soualem. Suite logicielle grt3d (global reactive transport 3d). *Rapport de contrat andra, INRIA*, 2011.

[39] W. E. Schiesser. The numerical method of lines : Integration of partial differential equations. *Academic Press, San Diego*, 1991.

[40] C. I. Steefel and A. C. Lasaga. A coupled model for transport of multiple chemical species and kinetic precipitation/dissolution reactions with application to reactive flow in single phase hydrothermal systems. *American Journal of Science, 294 :529-592*, 1994.

[41] C. Kanzaw T. De Luca, F. Facchinei. A theoretical and numerical comparisonvof some semismooth algorithms for complementarity problems. *Computational Optimization and applications,(16) 173-205*, 2000.

[42] M. Ulbrich. Semismooth newton methods for variational inequalities and constrained optimization problems in function spaces. *Mos-SIAM series on optimization*, 2011.

[43] S. M. Jonhson W. S. White and G. B. Dantzig. Chemica equilibrium in complex mixtures. *Journal of Chemical Physics, 28 : 751-755*, 1958.

[44] T. J.. Wolery. a computer program for geochemical aqueous speciation-solubility calculations : theoretical manual, user ?s guide, and related documentation (version 7.0). *Lawrence Livermore National Laboratory Report UCRL-MA-110662 (3)*, 1992.

[45] F. J. Zelznik and S.Gordon. Calculation of complex chemical equilibria. *Industrial and Engineering Chemistry, 60 : 27-57*, 1968.

[46] C. Zheng and G.D. Bennett. Applied contaminant transport modeling. *Second Edition. John Wiley and Sons, Inc*, 2002.

Table des figures

1 Classification des déchets radioactifs français en fonction de leur mode de gestion 8

2.1 section de sable soumis à un débit de fluide Q 28

3.1 Logiciel GRT3D (MODFLOW, MT3D et GDAE3D communiquent par des fichiers ; les autres interfaces sont des appels internes classiques. Le symbole * indique que les logiciels sont modifiés) 40
3.2 Les points d'entrée et les sundials 45
3.3 Domaine 1D pour les cas tests aqueuse-adv et ex11-adv 48
3.4 Cas test aqueux-adv (1D) : concentrations des composants aqueux calculés avec SIA 48
3.5 Cas test Ex11-adv (1D) : Concentrations de chaque composant aqueux calculées avec GDAE1D 50
3.6 Cas test Ex11-adv (1D) : Concentrations de composant fixé calculées avec GDAE1D 51
3.7 Domaine 1D 52
3.8 Cas Benchmark-adv (1D) : Concentrations de chaque composant aqueux calculées avec GDAE1D et GDAE3D 54
3.9 Concentrations initiales de chaque composant 55
3.10 Concentrations de chaque composant aqueux calculées avec GDAE1D . 56
3.11 Concentrations de chaque composant fixé calculées avec GDAE1D 57

4.1 Domaine 2D du test ANDRA et système chimique 60
4.2 Conditionnement de la Jacobienne pour les maillages 21×14, 41×28 pour chaque logiciel à plusieurs instants 66
4.3 Conditionnement de la Jacobienne pour les maillages 81×56, 161×112 et 322×224 pour chaque logiciel à plusieurs instants 67
4.4 Les concentrations à l'état initial pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$ 68
4.5 Les concentrations à l'instant t=3h 36m pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$ 69
4.6 Les concentrations à l'instant t=36h pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$ 69

4.7 Les concentrations à l'instant t=15 jours pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$ 70
4.8 Les concentrations à l'instant t=24 jours pour les deux composants Na^+, OH^- et les deux espèces secondaires H^+ et $H_3SiO_4^-$ 70
4.9 Les concentrations à l'instant t=30 jours pour les deux composants et les deux espèces 71
4.10 Erreur E_{Na^+} pour GDAE (bleu) et SNIA (en rouge) pour les différents maillages par rapport au temps CPU 72
4.11 Erreur E_{Na^+} pour GDAE (bleu) et SNIA (en rouge) pour les différents maillages par rapport au mémoire RAM 73

5.1 Le domaine 2D du cas test Benchmark (Les dimensions sont indiquées en utilisant la longueur de référence L). Il y a deux entrées et une sortie. 75
5.2 Champ de vitesse dans le domaine 2D du cas test Benchmark MoMaS pour le maillage 40×84 76
5.3 La charge sur le domaine 2D du cas test Benchmark MoMaS pour le maillage 40×84 77
5.4 Les concentrations du composant aqueux c_1 à différents instants (avec des échelles différentes) 81
5.5 Les concentrations des composants aqueux et composant fixé à l'instant $t = 10T$ 82
5.6 Les concentrations des composants aqueux et composant fixé à l'instant $t = 2000T$ 82
5.7 Les concentrations des composants aqueux et composant fixé à l'instant $t = 4000T$ 83
5.8 Les concentrations des composants aqueux et composant fixé à l'instant $t = 5010T$ 83
5.9 Les concentrations des composants aqueux et composant fixé à l'instant $t = 5200T$ 84
5.10 Les concentrations des composants aqueux et composant fixé à l'instant $t = 5800T$ 84
5.11 Le temps de calcul cumulé à chaque itération externe avec et sans traceur pour les quatre maillages. 86
5.12 Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 20×42 87
5.13 Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 32×56 88
5.14 Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 40×84 88
5.15 Conditionnement de la Jacobienne du cas test Benchmark MoMaS pour le maillage 80×168 89
5.16 Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 20×42 (À gauche : avec traceur, à droite : sans traceur) 90

5.17 Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 32×56 (À gauche : avec traceur, à droite : sans traceur) 90
5.18 Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 40×84 (À gauche : avec traceur, à droite : sans traceur) 91
5.19 Nombre de pas de temps internes cumulés du cas test Benchmark MoMaS pour le maillage 80×168 (À gauche : avec traceur, à droite : sans traceur) 91
5.20 Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 20×42 (à gauche : avec traceur, à droite : sans traceur) 92
5.21 Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 32×56 (à gauche : avec traceur, à droite : sans traceur) 92
5.22 Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 40×84 (à gauche : avec traceur, à droite : sans traceur) 93
5.23 Nombre cumulé d'échecs du test d'erreur du cas test Benchmark MoMaS pour le maillage 80×168 (à gauche : avec traceur, à droite : sans traceur) 93
5.24 Nombre cumulé d'échecs de la Jacobienne du cas test Benchmark MoMaS pour GRT3D (À gauche : avec traceur, à droite : sans traceur) 94
5.25 Nombre cumulé d'échecs de la Jacobienne du cas test Benchmark MoMaS pour GRT3DRL (À gauche : avec traceur, à droite : sans traceur) . . . 94
5.26 Nombre cumulé d'échecs de la Jacobienne du cas test Benchmark MoMaS pour GRT3DRSL (À gauche : avec traceur, à droite : sans traceur) . . . 95
5.27 Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 20×42 (À gauche : avec traceur, à droite : sans traceur) 95
5.28 Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 32×56 (À gauche : avec traceur, à droite : sans traceur) 96
5.29 Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 40×84 (À gauche : avec traceur, à droite : sans traceur) 96
5.30 Logarithme du nombre cumulé de résolutions et de factorisations du cas test Benchmark MoMaS pour le maillage 80×168 (À gauche : avec traceur, à droite : sans traceur) 96
5.31 Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 20×42 (À gauche : avec traceur, à droite : sans traceur) 97
5.32 Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 32×56 (À gauche : avec traceur, à droite : sans traceur) 97
5.33 Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 40×84 (À gauche : avec traceur, à droite : sans traceur) 98
5.34 Logarithme du temps cumulé des résolutions et des factorisations du cas test Benchmark MoMaS pour le maillage 80×168 (À gauche : avec traceur, à droite : sans traceur) 98

5.35 Le pourcentage du temps de résolution final et du temps de factorisation final par rapport au temps total 99

Résumé

La modélisation du transport réactif du contaminants en milieu poreux est un problème complexe cumulant les difficultés de la modélisation du transport avec celles de la modélisation de la chimie et surtout du couplage entre les deux. Cette modélisation conduit à un système d'équations aux dérivées partielles et algébriques dont les inconnues sont les quantités d'espèces chimiques. Une approche possible, déjà utilisée par ailleurs, est de choisir la méthode globale DAE : l'utilisation d'une méthode de lignes, correspondant à la discrétisation en espace seulement, conduit à un système différentiel algébrique (DAE) qui doit être résolu par un solveur adapté. Dans notre cas, on utilise le solveur IDA de Sundials qui s'appuie sur une méthode implicite, à ordre et pas variables, et qui requiert à chaque pas de temps la résolution d'un grand système non linéaire associé à une matrice jacobienne. Cette méthode est implémentée dans un logiciel qui s'appelle GRT3D (Transport Réactif Global en 3D).

Le présent travail présente une amélioration de la méthode GDAE, du point de vue de la performance, de la stabilité et de la robustesse. Nous avons ainsi enrichi les possibilités de GRT3D, par la prise en compte complète des équations de précipitation-dissolution permettant l'apparition ou la disparition d'une espèce précipitée. En complément de l'étude de la méthode GDAE, nous présentons aussi une méthode séquentielle non itérative (SNIA), qui est une méthode basée sur le schéma d'Euler explicite : à chaque pas de temps, on résout explicitement l'équation de transport et on utilise ces calculs comme données pour le système chimique, résolu dans chaque maille de façon indépendante. Nous présentons aussi une comparaison entre cette méthode et l'approche GDAE. Des résultats numériques pour deux cas tests, celui proposé par l'ANDRA (cas-test 2D) d'une part, celui proposé par le groupe MoMas (Benchmark "easy case") d'autre part, sont enfin présentés, commentés et analysés.

Mots clés : DAE, global approche GDAE, méthode séquentielle non itérative (SNIA), méthode de Newton, Sundials, espèce aqueuse, espèce fixé, espèce précipité, problème de complémentarité, méthode de semismooth.

Abstract

Modeling reactive transport of contaminants in porous media is a complex problem combining the difficulties of modeling the trasport with those of modeling the chemistry and especially the coupling between the two . This model leads to a system of partial differential equations and algebraic equations whose unknowns are the quantities of chemical species. One approach , already used elsewhere , is choosing the global DAE method : using the method of lines, discretization in space only, leads to a differential algebraic system (DAE) to be solved by a suitable solver . In our case , the solver IDA Sundials relies on an implicit method, order is used but not variables, and requires at each time solving a large nonlinear system associated with a Jacobian matrix . This method is implemented in a software called GRT3D (Global Reactive Transport in 3D). This paper presents an improved GDAE method , from the standpoint of performance, the stability and robustness. We have enriched the possibilities of GRT3D , by taking full account of the equations of dissolution - precipitation for the appearance or disappearance of precipitated species. In addition to the study of the GDAE method, we also present a non-iterative sequential method (SNIA) which is a method based on the explicit Euler scheme : at each time step, we explicitly solve the transport equation and we use these calculations as data for the chemical system which is resolved in each cell independently.
We also present a comparison between this method and GDAE approach. Numerical results for two test cases , one proposed by ANDRA (2D test case) on one hand and one proposed by the group MOMAS (Benchmark "easy case") on the other hand, are finally presented , discussed and analyzed.

Keywords : DAE, global GDAE, non iterative sequential method (SNIA), Newton's method, Sundials, aqueous species, fixed species, precipitated species, complementarity problem, semismooth method.